New Techniques for Lower Bounds in Graph Streaming and Distributed Computing

by

Janani Sundaresan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2026

© Janani Sundaresan 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Kasper Green Larsen Professor, Department of Computer Science, Aarhus University
Supervisor(s):	Sepehr Assadi Associate Professor, Cheriton School of Computer Science, University of Waterloo
Internal Members:	Eric Blais Associate Professor, Cheriton School of Computer Science, University of Waterloo
	Lap Chi Lau Professor, Cheriton School of Computer Science, University of Waterloo
	Santhoshini Velusamy Assistant Professor, Cheriton School of Computer Science, University of Waterloo
Internal-External Member:	Vijay Bhattiprolu Assistant Professor, Department of Combinatorics and Optimization, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

All the results of this thesis are based on papers that I have coauthored, as well as works in preparation. I am a main contributor in all these publications. The details are as follows.

- Chapter 5 is based on:
  [AKNS24a] *$O(\log \log n)$ Passes is Optimal for Semi-Streaming Maximal Independet Set*
  Sepehr Assadi, Christian Konrad, Kheeran K. Naidu and Janani Sundaresan
  STOC 2024 (invited to SICOMP special issue)

- Chapter 6 is based on:

  - [AS23a] *Hidden Permutations to the Rescue: Multi-Pass Semi-Streaming Lower Bounds for Approximate Matchings*
    Sepehr Assadi and Janani Sundaresan
    FOCS 2023
  - [ABK+25] *Settling the Pass Complexity of Maximum Matching in Dynamic Streams*
    Sepehr Assadi, Soheil Behnezhad, Christian Konrad, Kheeran K. Naidu and Janani Sundaresan
    SODA 2025 (invited to TALG special issue)
  - [AKNS26] *Multi-pass Lower Bounds for Streaming Matchings*
    Sepehr Assadi, Christian Konrad, Kheeran K. Naidu and Janani Sundaresan
    Currently under preparation.

- Chapter 7 is based on:
  [AS25] *Distributed Triangle Detection is Hard in Few Rounds*
  Sepehr Assadi and Janani Sundaresan
  FOCS 2025 (invited to SICOMP special issue)

- Chapter 8 is based on:
  [Sun25] *Optimal Communication Complexity of Chained Index*
  Janani Sundaresan
  ITCS 2025

- Chapter 9 is based on:
  [AHS26] *Better Bounds for Semi-Streaming Single Source Shortest Paths*
  Sepehr Assadi, Gary Hoppenworth and Janani Sundaresan
  SODA 2026

- Chapter 10 is based on:
  [AS26] *Settling the Pass Complexity of Streaming Set Cover*
  Sepehr Assadi and Janani Sundaresan
  STOC 2026

Omitted Papers:

- [ASY26] *Coloring Graphs with Few Colors in the Streaming Model*
  Sepehr Assadi, Janani Sundaresan and Helia Yazdanyar
  SODA 2026

- [AS23b] *(Noisy) Gap Cycle Counting Strikes Back: Random Order Streaming Lower Bounds for Connected Components and Beyond*
  Sepehr Assadi and Janani Sundaresan
  STOC 2023

- [GKSS23] *Look Before, Before You Leap: Online Vector Load Balancing with Few Reassignments*
  Varun Gupta, Ravishankar Krishnaswamy, Sai Sandeep, and Janani Sundaresan
  ITCS 2023

Abstract

In this dissertation, we study lower bounds in the streaming and distributed computing models for graph problems. Both models arise naturally in the design of algorithms for the massive graphs that are ubiquitous in practice today.

In the streaming model, the edges of the graph arrive in an arbitrary order and must be processed without storing the entire graph. Of particular interest is the semi-streaming setting, in which the algorithm is allowed space that is near-linear in the number of vertices. The stream may be scanned multiple times, and the goal is to minimize the number of passes needed to solve the problem.

In distributed computing, each vertex acts as a computational agent and initially knows only its immediate neighborhood. Vertices communicate by sending messages along the edges of the graph. Communication proceeds in synchronous rounds: in each round, every vertex can send a (possibly different) message to each of its neighbors, receive all incoming messages, and then perform local computation. In the CONGEST model, each such message is restricted to a length logarithmic in the number of vertices. The goal is to minimize the number of rounds required to solve the problem.

We highlight three results from this dissertation:

1. The number of passes required to find a maximal independent set in semi-streaming is $\Omega(\log \log n)$. Our result is tight, matching a prior algorithm. This is the first multi-pass lower bound for this problem.

2. The number of passes required to find any $(1-\varepsilon)$-approximation of maximum matchings is $\Omega(\log(1/\varepsilon))$ in semi-streaming. This is the first lower bound on the number of passes with a dependence on ε for any constant ε.

3. We prove an $\Omega(\log \log n)$ lower bound on the number of rounds required to detect a triangle in CONGEST. This is the first multi-round lower bound for this problem.

Lower bounds in both models are proven via communication complexity: the input graph is split among multiple players who send messages to each other over a limited number of rounds. A common theme underlying the highlighted results is to build on and extend the round-elimination technique from communication complexity literature. Our main technical contribution is to adapt this method for a range of different settings to obtain the first nontrivial multi-pass and multi-round lower bounds for these problems.

Acknowledgements

First and foremost, I want to thank my advisor Sepehr. He was extremely patient with me and taught me all aspects of research: understanding prior literature, breaking hard questions into manageable parts, writing papers, reviewing other work, and giving talks. His insistence on building a strong community in the theory lab is something I hope to carry forward with me, as the communities at Waterloo and Rutgers were a steady source of support for all the students. The one other thing I wish he had taught me is how to be better at One Night Werewolf, as my skills leave much to be desired. Thank you, Sepehr.

Next, I want to thank my committee members Eric Blais, Lap Chi Lau, Vijay Bhattiprolu, Santhoshini Velusamy, and Kasper Green Larsen for taking the time to read this dissertation. Their comments were invaluable and the final version is much better than the first draft they read. Thank you also, Kasper, for visiting Waterloo for my defense.

I have learned a great deal from my co-authors while working on these results, and I am grateful for the many hours spent discussing research and writing papers together. Thanks to Soheil Behnezhad, Varun Gupta, Gary Hoppenworth, Christian Konrad, Ravishankar Krishnaswamy, Kheeran K. Naidu, Sai Sandeep, and Helia Yazdanyar.

Santhoshini Velusamy has been an incredible mentor and hosted me for a wonderful month at TTIC in summer of 2025. Thank you for being someone I have looked up to, both academically and personally.

Jayalal Sarma instilled in me the joy of theoretical computer science through his excellent courses and introduced me to research in my undergraduate days.

Many conversations with Eric Blais at Waterloo have shaped the way I approach problems and communicate ideas. He also taught a particularly inspiring course on randomized complexity theory in winter of 2025. Lap Chi Lau and Rafael Oliveira always made time for students at Waterloo, and their feedback on my talks was consistently thoughtful and encouraging.

Ronitt Rubinfeld and Clement Canonne were instrumental in creating a memorable experience for visitors to the Simons Institute Sublinear Algorithms program in the summer of 2024.

I was fortunate to be part of two active theory groups at Rutgers University and University of Waterloo. I will miss the impromptu whiteboard discussions, problem sessions and the productive chaos of both places. My academic siblings, Vihan Shah and Parth Mittal, were with me through almost the entirety of graduate school and things would have been much duller without them. I am sorry that I never went climbing with you both.

At Rutgers, I am grateful to Vikrant Ashvinkumar, Vishwas Bhargava, Chengyuan Deng, Prathamesh Dharangutte, Surya Teja Gavva, Rashmika Goswami, Prantar Ghosh, Zach Langley, Chaitanya Nalam, Adarsh Srinivasan and Harsha Tirumala.

A special shout-out to Aditi Dudeja for making me feel welcome at Rutgers when I just started by taking me out to lunch and to Hanna Komlos for organizing social events in the theory group at Rutgers. Thank you also to Ariel Schvartzman and Dave Pennock who worked with me and gave me the opportunity to learn from them. Karthik C S was always around for a chat. Nicole Wein and Zihan Tan were generous with their advice, especially

with the academic job process. The Rutgers student-run reading groups were regularly attended by Mike Saks and greatly benefited from his presence. The combinatorics course taught by Jeff Kahn in Winter 2021 gave me excitement during an otherwise isolating period of the pandemic.

At Waterloo, I thank Mahtab Alghasi, Enver Aman, Omkar Baraskar, Arkaprava Choudhury, Mohammad Ebtehaj, Renato Ferreira Pinto Jr., Prashant Gokhale, Sriram Gopalakrishnan, Parth Gupta, Raymond Liu, Argyris Mouzakis, Rian Neogi, Matthew Regehr, Remy El Sabeh, Ajay Singh, Dinis Vitorino, Ingrid Vukusic and Robert (the student) Wang.

I loved all the nights out with Avantika Agarwal and Helia Yazdanyar. Robert (the postdoc) Andrews taught me how to use the coffee machine among many other things, and I will try to emulate his patience and kindness towards me when I bothered him with questions. Thanks to Vahid R. Asadi for all the lively (non-)research discussions in my last year. Our office was rarely quiet thanks to Abhibhav Garg, Cameron Seth, and the many visitors who stopped by to chat.

I started going to the gym regularly while at Waterloo, and Mayuri was basically my gym trainer. Thanks for keeping me on track. The dance classes offered by Elnaz at the University of Waterloo served as a fun source of stress relief.

I want to thank my friends Akashnathan, Arjun, Barath, Midhuna, Navodita, Pari and Vishnu, who stuck with me and were just a phone call away. My roommates Hellan and Sana made a new country feel like home and gave me excellent home-cooked meals.

I want to thank Arasu Arun for always pushing me to aim higher and believing in me. He remained patient through every difficult stretch and moment of self-doubt, and celebrated every small success along the way. Lastly, even after I moved away, my family made sure that I spoke to them constantly and kept homesickness (mostly) at bay. The support and encouragement from my parents were crucial in making this dissertation happen. I could count on my brother to make me laugh, no matter what else was going on in my life. Thank you to all of you for being my biggest cheerleaders and sources of strength.

Dedication

To my parents and my little brother.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract vi

Acknowledgements vii

List of Figures xv

I Context and Contents 1

1 Introduction 2

1.1 Contributions of this Thesis . . . 5

1.1.1 Organization of this Thesis . . . 11

2 Landscape of Prior Lower Bounds 12

2.1 General Structure of Graph Lower Bounds . . . 12

2.2 Streaming Lower Bounds . . . 14

2.2.1 Role of Communication Complexity . . . 14

2.2.2 Reductions to Index, Pointer Chasing and Set Disjointness . . . 15

2.2.3 New Techniques for Stronger Lower Bounds . . . 18

2.3 Distributed Computing Lower Bounds . . . 21

2.3.1 LOCAL Lower Bounds . . . 22

2.3.2 Information Bottleneck . . . 23

2.3.3 Further Techniques . . . 25

3 A Brief Technical Overview 27

3.1 On Round Elimination . . . 27

3.1.1 Using Round Elimination for Index . . . 28

3.2 Streaming Maximal Independent Sets . . . 31

3.3 Streaming Matchings . . . 37

3.3.1 Insertion-only Streams . . . 37
3.3.2 Dynamic Streams . . . 40
3.4 Distributed Triangle Detection . . . 43
3.5 Lower Bounds Beyond Embedding . . . 46

4 Preliminaries **51**
4.1 Models of Computation . . . 51
4.2 Combinatorial Tools . . . 52
4.2.1 Ruzsa-Szemerédi graphs . . . 52
4.2.2 Concentration Inequalities . . . 53
4.3 Information-theoretic Tools . . . 53
4.3.1 Useful Properties of Entropy and Mutual Information . . . 54
4.3.2 Measures of Distance Between Distributions . . . 55

II Lower Bounds via Embedding Arguments **57**

5 Maximal Independent Sets in Streaming **58**
5.1 Introduction . . . 58
5.2 Technical Overview . . . 59
5.2.1 An "Ideal" Lower Bound Approach . . . 59
5.2.2 Our *Actual* Lower Bound Approach: Hierarchical Embeddings . . . 61
5.2.3 Combinatorial Part: Dense Graphs with Many Induced Subgraphs . . . 63
5.2.4 Information-Theoretic Part: Round Elimination via Direct-Sum . . . 65
5.3 Preliminaries . . . 67
5.3.1 (Multi-Party) Communication Complexity . . . 67
5.3.2 Information Cost and Message Compression . . . 68
5.4 Disjoint-Unique-Paths Graphs and Embedding Products . . . 69
5.4.1 Disjoint-Unique-Paths (DUP) Graphs . . . 70
5.4.2 Embedding Products . . . 71
5.4.3 A Construction of DUP Graphs . . . 73
5.5 A Round vs Communication Tradeoff for MIS . . . 76
5.5.1 A Family of Hard Distributions for MIS . . . 77
5.5.2 Basic Properties of the Distributions $\{\mathcal{G}_r(n)\}_{r\geq 1}$ . . . 81
5.5.3 Search Sequences and Predicates . . . 83
5.6 New Round Elimination . . . 85
5.6.1 The Setup for the Proof of Theorem 10 . . . 86
5.6.2 Step 1: A Low Communication Protocol for $\mathbb{F}_{r-1}$ in r Rounds . . . 87

5.6.3 Step 2: A Protocol for $\mathbb{F}_{r-1}$ in $(r-1)$ Rounds via Round Elimination 92
5.7 Using New Round Elimination . . . 99
5.7.1 Proof of Lemma 5.7.1 . . . 100
5.7.2 Concluding the Proof of Theorem 9 . . . 104

6 Matchings in Streams **106**
6.1 Insertion-only Streams . . . 109
6.2 Dynamic Streams . . . 122
6.2.1 Preliminaries . . . 123
6.2.2 A Hard Distribution . . . 124
6.2.3 Reduction to Bipartite Matching in Dynamic Streams . . . 148

7 Distributed Triangle Detection **161**
7.1 Introduction . . . 161
7.1.1 Related Work . . . 164
7.2 Preliminaries . . . 165
7.2.1 Notation . . . 165
7.2.2 Model of Communication . . . 165
7.3 Technical Overview . . . 167
7.3.1 Background: Round Elimination Arguments in Prior Work . . . 167
7.3.2 Our Approach . . . 169
7.4 A Hard Distribution and its Properties . . . 180
7.4.1 Base Case: Hard Distribution for 0-rounds . . . 181
7.4.2 Hard Distribution for r-rounds . . . 182
7.4.3 Another Way of Sampling from Hard Distribution . . . 186
7.4.4 Distribution of Input and Messages . . . 191
7.4.5 Proof of Theorem 12 Barring Round Elimination . . . 193
7.5 Round Elimination: Description . . . 194
7.5.1 Key Steps of the Protocol . . . 194
7.5.2 Full Protocol π_{r-1} and its Distribution . . . 196
7.5.3 Proof of Lemma 7.4.16 . . . 199
7.6 Round Elimination: Analysis . . . 200
7.6.1 Setting up the Analysis . . . 200
7.6.2 Proof of Lemma 7.5.6 . . . 203
7.6.3 Messages from One Vertex have Low Correlation . . . 204

III Lower Bounds beyond Embedding **211**

8 Chained Index **212**
8.1 Introduction . . . 212
8.2 Preliminaries . . . 216
8.3 The Lower Bound . . . 218

9 Streaming Shortest Paths **227**
9.1 Introduction . . . 227
9.2 Preliminaries . . . 230
9.3 The Lower Bound . . . 230

10 Streaming Set Cover **250**
10.1 Introduction . . . 250
10.2 Preliminaries . . . 254
10.3 The Lower Bound . . . 255

11 Future Directions **264**
11.1 Immediate Questions . . . 264
11.2 Broader Directions . . . 266
11.2.1 Other Related Models. . . . 266
11.2.2 Lower Bounds against Edge Sampling . . . 267
11.2.3 Structure of Hard Instances: Random versus Combinatorial . . . 267
11.2.4 Lower Bounds for Problems with Minimal Locality . . . 267
11.3 Conclusion . . . 268

References **268**

IV Appendices **305**

A Missing Proofs **306**
A.1 Round Elimination for Index: Proof of Claim 3.1.3 . . . 306
A.2 Matchings in Random Bipartite Graphs: Proof of Proposition 9.3.15 . . . 307
A.3 Extending Proposition 5.4.5 to all Graph Sizes . . . 308

B CONGEST Lower Bounds and Circuit Complexity Barriers **310**
B.1 Background on Expander Decomposition and Routing . . . 310
B.2 Results on Expander Decomposition in CONGEST . . . 311
B.3 Results on Routing Algorithms in CONGEST . . . 312

B.4 From Circuits to CONGEST Algorithms 313

List of Figures

2.1 Figure depicting the hard instance for shortest path. The path between s and t has length 3 if and only if an edge exists between $i^\star$ and $j^\star$. Otherwise, the path length is at least 5. 13

2.2 Index problem. Alice has a string $x \in \{0,1\}^k$. Bob has an index $i \in [k]$ that points to x. Alice can send a single message to Bob, who has to output value of x at position i. 13

2.3 Streaming algorithms give communication protocols. The players run the algorithm on their inputs and send the memory state to the other player as messages. 14

2.4 Some communication problems can be reduced to graph streaming problems. 15

2.5 An illustration of the functions in pointer chasing. Alice holds f_1 and Bob holds f_2. If $z_1 = a_1$, $z_2 = b_4$ and $z_3 = a_3$. 16

2.6 An illlustration of direct sum of pointer chasing. Each color represents one independent instance. Bob holds all the functions in one column, and Alice in the next and so on. 17

2.7 An illustration of the hard instance of s, t reachability for one pass algorithms. 20

2.8 An illustration of using set intersecion instances to hide the pointers in one pass reachability. 20

2.9 Using RS graphs to hide set intersection instances. 22

2.10 Round Elimination to prove LOCAL lower bounds. 23

2.11 An illustration of the hard instance of APSP in CONGEST. Each of the dashed edges can have weight as α or 1. The vertex v must learn all these weights using its one incident edge. 24

2.12 An illustration of using a small cut to prove lower bounds in CONGEST. There are only 2 edges between G_A and G_B, so a lower bound of $\Omega(t/\log n)$ will be proved. Here, t is a lower bound on the communication complexity of the underlying hard problem. 25

3.1 The round elimination framework of [MNSW95]. The hard instance is made of k smaller hard instances, where solving I^* gives the solution to the full instance. The identity of I^* is not revealed until the first round messages are sent. 28

3.2 Alice's input is G_1 and G_2 which are copies of the same random bipartite graph G. Bob's input connects every vertex other than copies of u and v to each other. In the image, this is illustrated as w being connected to all vertices in the grey cloud and vice-versa. If we pick a vertex w in the left to join the MIS, all its neighbors ar forced out. Only vertices u and v in the right remain to join the MIS (the figure only shows some of w's edges). 32

3.3 The special instances should not corrupt each other. Bob's edges can be used to isolate a special instance I^* 33

3.4 An example of DUP graph with $q = 3, p = 2$ and $k = 2$. In each UPC, there is no path of length 2 from a starting vertex to an end vertex unless the path belongs to the UPC itself. 35

3.5 Our round elimination framework. To find the answer to the instances, the players have to solve all p smaller instances in row $i^\star$. 35

3.6 An illustration of embedding smaller instances into DUP graphs. The figure only shows how one UPC is replaced, in the final instance, *all* UPCs in G_{dup} are replaced with smaller instances. 36

3.7 One round hard instance for (2/3) approximate matching in [GKK12]. Any $(2/3+\varepsilon)$-approximate matching must find at least an ε-fraction of the yellow edges. 38

3.8 A hard instance constructed by [KN24], Any $(8/9+\varepsilon)$-approximate matching finds at least an ε-fraction of the edges of the special matchings in the smaller instances. The smaller instances look exactly as in Figure 3.7a. 39

3.9 An illustration of the inputs A and B to Alice and Bob in the lower bound of [DK20]. Here, $n = 6$ and $i^\star = 4$ and both σ_r and σ_c are identity permutations. In general, the submatrix of B corresponds to a combinatorial rectangle in A, the induced matching is of size $n/2 - o(n)$ with high probability, and the vertex cover is of size $o(n)$. 41

3.10 An illustration of the two round hard distribution of Theorem 4. (X, Y) denote instances sampled from the 1-round hard distribution. Each submatrix has its own independent permutation ρ_r, ρ_c. Alice and Bob have to solve the instances highlighted in red. The squares shaded in blue are zeroed out in both matrices. 42

3.11 An illustration of the one round hard instance for Theorem 5. The dashed lines represent edges that may not exist. If these edges exist, there is a triangle in the input. 44

3.12 Viewing the one round hard instance as a correlated index problem. Vertex x does not initially know the pointers y, z but comes to know them only after the message from vertex y (which reveals that it may be a part of a triangle, hence fixing the pointer to z) and from vertex z. The message from y of $o(n)$ length contains very little information about whether z is present in the neighbor list. 45

3.13 Padding with dummy vertices to create a 2-round hard instance. 45

3.14 An illustration of the functions in pointer chasing. Alice holds f_1 and Bob holds f_2. If $z_1 = a_1$, $z_2 = b_4$ and $z_3 = a_3$. 47
3.15 Equality of Pointer Chasing. When the final vertex is x for both the instances, there is a (u, v)-path of length k. Otherwise, the two vertices are not connected. 48
3.16 The blue elements are from a $n^{-1/p}$ fraction of the universe U. The input sets are restricted to U' after it is revealed, but initially they cover all the elements of U. 49

5.1 Alice's input is G_1 and G_2 which are copies of the same random bipartite graph G. Bob's input connects every vertex other than copies of u and v to each other. If we pick a vertex w in the left to join the MIS, only vertices u and v in the right remain to join the MIS (the figure only shows some of w's edges). 59
5.2 An illustration of the base graphs discussed above. Here $q = 4$ and $p = 2$ and each box denotes a single sub-instance (with different colors for different $I_{i,*}$ for $i \in [4]$). There are no edges between vertices of $I_{i,*}$ except for edges of the same sub-instances, which themselves are vertex-disjoint. While the figure may suggest sub-instances as being bipartite graphs, that crucially cannot be true in the actual construction. 62
5.3 An illustration of the hierarchical embedding technique. 63
5.4 An example of UPCs in the graph G on the left. In each UPC, there is no path of length 2 from a starting vertex to an end vertex unless the path belongs to the UPC itself. 64
5.5 If the solid (red) edge between u and v is inserted by any other k-colorable graph $H_{i,j}$ for $i \neq 1$, we will also find a shortcut between "outer" versions of u and v, which cannot exist in the outer graph. 65
5.6 An illustration of layered graphs and strictly-layered graphs with a layered path. 70
5.7 An illustration of a UPC in a strictly-layered graph. Neither the dashed red edges (between the vertices of different paths in a UPC) nor the edges of the solid red path (using outside vertices) are allowed. 70
5.8 An illustration of the embedding product. For simplicity of exposition, this figure only shows a part of the final graph obtained via embedding inside a single UPC of the DUP graph; in the actual construction, every UPC is replaced by an embedded graph. 72

5.9 The gray boxes denote the subgraphs $H_{1,1}$ and $H_{1,2}$ that are created w.r.t. paths $P_{1,1}$ and $P_{1,2}$ in the same UPC of G_{DUP}. *If* there is an edge between vertices u and v that is added w.r.t. another path $P_{i',j'}$, it implies that some subpath of $P_{i',j'}$ connects two vertices in paths $P_{1,1}$ and $P_{1,2}$ together – this is because any edge of G is added "alongside" a path of G_{DUP}, although it is possible that it *shortcuts* a couple of layers in between as $H_{i',j'}$ may have edges between non-consecutive layers. This in turn creates another layered path between $\text{START}(P_{1,2})$ and $\text{FINAL}(P_{1,1})$, which is not allowed in a UPC, a contradiction. 73

5.10 An illustration of the construction of DUP graphs. The circle x at left does not correspond to a vertex and instead is used to define a UPC. The red path cannot exist because it requires $y_3 + y'_1 + y'_2 + y'_3 + y'_4 = 5 \cdot y_2$ which cannot happen because A is average free. 75

5.11 An illustration of the distribution $\mathcal{G}_r(n)$. The paths at the bottom correspond to the UPC $\mathcal{P}^\star$; each path is "replaced" with an instance H from $\mathcal{G}_{r-1}(n_{r-1})$ via the embedding product. Since the embedding product affects the entire graph, the paths in other UPCs are also replaced by instances from $\mathcal{G}_{r-1}(n_{r-1})$, e.g., for the top (red) block (remaining instances are not drawn). The last player Q_{r+1} receives a bipartite clique between the vertices corresponding to $W \setminus W^\star$ in $\widetilde{G}_L$ and $\widetilde{G}_R$, denoted by the top (green) blocks. Other players Q_a for $a \in [r]$ receive the inputs of the a^{th} player in *all* instances H sampled from $\mathcal{G}_{r-1}(n_{r-1})$ when placed inside graphs $\widetilde{G}_L$ and $\widetilde{G}_R$. 78

5.12 An illustration of case 1: picking vertex v in the MIS "removes" all vertices on $\widetilde{G}_L$ that are not part of the special subgraphs as the remaining ones are all neighbor to v (not all edges of v are drawn). Since the special subgraphs are now isolated, the MIS of G now needs to contain a separate MIS for each of the special subgraphs in $\widetilde{G}_R$ in bottom right (drawn in blue). 82

5.13 An illustration of a search sequence. Here, k_r picks a special sub-instance H_{t_r,k_r} from $\mathcal{G}_{r-1}$ (with t_r being the index of the special UPC of the outer instance), and then "inside" that instance, k_{r-1} picks another sub-instance $H_{t_{r-1},k_{r-1}}$ from $\mathcal{G}_{r-2}$ (with t_{r-1} being the index of the special UPC of the instance H_{t_r,k_r}), and this continues on. 84

5.14 A schematic organization of the proof of Theorem 10: step 1 encompasses moving from part 1 to part 3 here, and step 2 is for moving from part 3 to part 4. 87

6.1 An example of the embedding product on RS graphs. The RS graph from Figure 3.7 is used and there are totally 12 many smaller graphs. We show three such graphs on the right. The shaded circles indicate vertices which are groups of three. Each edge is replaced by a specific bipartite graph from the given family. 110

6.2 The hard distribution $\mathcal{G}_0(n)$ and the gadgets. 112

6.3 An illustration of the distribution $\mathcal{G}_p$ for $p \geq 1$. When $p = 1$, the smaller instances are as in Figure 6.2. Only the smaller instances in the special matching are shown. All edges of the outer RS graph are replaced also. The shaded circles represent groups of vertices. 113

6.4 An illustration of $\mathsf{AHM}_r(n_r, \alpha)$ where $b_r = 7$, $k_r^\star = 2$, and $\alpha = 2/7$, i.e., $b_r \cdot \alpha = 2$. In this figure, the $b_r \times b_r$ matrix is presented after applying the respective permutations of rows and columns using σ_{R} and σ_{C}, which are held by Bob (alternatively, think of σ_{R} and σ_{C} as identity permutations here). In each position of the matrix, the corresponding input is either a complete instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ denoted by $(A, B) = (A^{(r-1)}, B^{(r-1)})$ or a partial instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ denoted by $B = B^{(r-1)}$ due to the asymmetry of the construction. All of the B's are held by Alice, all of the A's are held by Bob, and Bob further holds the B's corresponding to B^{OFF}, which are bolded and underlined. The shaded instances are the special sub-instances $(A^{\text{SPEC}}, B^{\text{SPEC}})$, i.e., the ones in the diagonal of the bolded box. The remaining B's belong to B^{REST}, i.e., the ones not in the bolded box. Given the search sequence $(k_r^\star, k_{r-1}^\star, \ldots, k_1^\star)$, the goal is to solve the the instance identified by $k_r^\star = 2$ (shaded in green, and denoted by $(A_2^{\text{SPEC}}, B_2^{\text{SPEC}})$), using the search sequence $(k_{r-1}^\star, \ldots, k_1^\star)$. 128

6.5 An illustration of the partial definitions T_{ROW} and T_{COL} for an instance of $\mathsf{AHM}_r(n_r, \alpha)$ where $b_r = 7$, $\alpha = 2/7$, and $k = 2$. In this figure, the $b_r \times b_r$ matrix is after applying the respective permutations of rows and columns using σ_{R} and σ_{C} held by Bob. In each position of the matrix, the corresponding input is the $B's$ from each instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ that is held by Alice. T_{ROW} and T_{COL} are partial definitions of the permutations ρ_{R} and ρ_{C}, i.e., the inverses of σ_{R} and σ_{C}, respectively. T_{ROW} corresponds to fixing the rows filled red and T_{COL} corresponds to fixing the columns filled blue, which implies fixing positions filled purple. The unfilled (white) positions remain unrestricted by $T_{\text{ROW}}, T_{\text{COL}}$, i.e., the green outlined special position B_k^{SPEC} could still (uniformly) be in any of the unfilled positions when conditioned on $T_{\text{ROW}}, T_{\text{COL}}$ (instead of $\sigma_{\text{R}}, \sigma_{\text{C}}$). 143

6.6 An illustration of the bit graph on vertices corresponding to a bit $x \in \{0, 1\}$. 149

6.7 Illustrations of Alice's edges $\textsc{edges}(r, A^{(r)}, L, R)$ (Figure 6.7a) and Bob's edges $\textsc{edges}(r, B^{(r)}, L, R)$ (Figure 6.7b). In both illustrations, the nodes on the left and right corresponds to vertex groups $L[i]$ and $R[j]$ for each $i, j \in [b_r]$, respectively. An edge between vertex groups $L[i]$ and $R[j]$ represents the set of edges corresponding to the instance $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$. 151

6.8 An illustration of a graph $G = \textsc{graph}(A^{(r)}, B^{(r)})$ constructed from an instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$. The nodes on the left and right corresponds to vertex groups $L[i]$ and $R[j]$ for each $i, j \in [b_r]$ and are ordered after applying the permutations σ_{R} and σ_{C}, respectively. An edge between vertex groups $L[i]$ and $R[j]$ represents the set of edges corresponding to the instance $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$, where a lack of edge implies that all edges inserted have also been deleted, i.e., the off-diagonal sub-instances $(A^{\text{OFF}}, B^{\text{OFF}})$. Each purple edge represents the edges that correspond to each special sub-instance in $(A^{\text{SPEC}}, B^{\text{SPEC}})$. The remaining thick gray edges represent the edges corresponding to sub-instances in B^{REST}. The box represents the subgraph of G induced by the vertices $L^{\text{SPEC}} \cup R^{\text{SPEC}}$. See Figure 6.4 for a comparison to AHM_r. 152

7.1 An illustration of our hard input distribution. The solid (blue) edges show the $(r-1)$-round hard instance inside an r-round hard instance and the dashed (gray) edges show the extra unique channels sampled for each vertex of the inner $(r-1)$-round instance. 170

7.2 An illustration of sampling partial $(r-1)$-round sub-instances. Here, from the perspective of x, the edge (x, y) plus any of K_i's for $i \in [k_r]$ form a proper input of x in an instance sampled from $\mathcal{G}_{r-1}$. One of these instances corresponds to the special inner instance G^*, while for all other instances, the instance is not complete, as in there are no edges between Y- and Z-part of the instance. 173

7.3 An illustration of sampling messages via a mixture of public and pair randomness. The type of the channel between (x, y) any (x, y') for any $y' \in L$ is the same. 175

7.4 An illustration of an instance from the hard distribution for 0-rounds with $n_0 = 3$. The middle (yellow) vertices are a^*, b^*, c^*. The double line indicates that the edge is present with probability $1/2$. 181

7.5 An illustration of some parts of the 1-round instance. Inner vertices from 0-round instance are blue and yellow (in accordance with Figure 7.4), and outer vertices are green. We used $n_0 = d_1 = 3$ for illustration, but this is certainly not true in the actual instance. Dashed lines indicate a channel of type 1, straight lines indicate a channel of type 0, and other pairs of vertices with no lines are of type 2. The input of blue and yellow vertices of the inner graph have exactly 3 channels of type 0 and type 1 in total. 184

7.6 An illustration of what the input of one inner vertex x looks like in $\widetilde{\mathcal{G}}_r$. Pink outer vertices correspond to collections $\mathcal{J}$, green outer vertices correspond to collections $\mathcal{K}$ (only for one other layer Y are shown, this is repeated once more), blue outer vertices correspond to sets L (again, only for Y are shown, this is repeated for Z), yellow vertices correspond to vertices in $Y^* \cup Z^*$, and the unshaded vertices are the other outer vertices with various types to ensure correct number of channels of each type $t \in [r] \cup \{0\}$. 188

8.1 An illustration of the $\textsc{chain}_{n,k}$ problem with k correlated sub-instances of $\textsc{index}_n$ from Definition 8.1.1. The arrows illustrate that the message is from P_i to P_{i+1} for $i \in [k]$. 212
8.2 An illustration of the $\textsc{chain}_{n,k}$ problem from Definition 8.3.1. The solid arrows illustrate that player $\mathcal{P}_i$ writes a message M_i to the board. The dashed arrows indicate that $\mathcal{P}_i$ can read the contents of the board. It also shows the order in which the messages are sent by the players and indices are released. . . . 218

9.1 Product of two layered graphs from $\mathcal{L}_{3,3}$ lying in $\mathcal{L}_{3,9}$. Some edges are omitted. 232
9.2 Join of two layered graphs from $\mathcal{L}_{3,3}$ lying in $\mathcal{L}_{6,3}$. 234

10.1 An illustration of pointer chasing for $p = 3, a = 3$ and $b = 4$. Each function is represented by a 3×4 grid containing the corresponding values. The special pointers of each player are highlighted in green. The red dashed arrows represent how the pointers move. The answer to the pointer chasing instance is 0. 256
10.2 Creation of U-common family for each player: this is repeated for all $P^1, P^2, P^3, \ldots, P^p$. The illustration is when $a = 3$ and $b = 4$. There are three sets U_1, U_2 and U_3 each of size $n/(10\alpha \cdot (p+1))$ chosen uniformly at random. These are split into four sets, each of size $n/(10\alpha \cdot b \cdot (p+1)) = n/(40\alpha \cdot (p+1))$. Although the sets U_1, U_2 and U_3 are shown as disjoint, they may have common elements as they are sampled independently. However, for each $i \in [3]$, the sets $U_{i,j}$ for $j \in [4]$ are disjoint from each other. 257
10.3 An illustration of the S-private family for players P^1 and P^2 with $a = 3, b = 4$. Each S^i_j for $i \in [2], j \in [3]$ is a union of all the sets to its right. 259

Part I

Context and Contents

1 Introduction

Leonhard Euler laid the foundations of graph theory in his famous resolution of the problem "The Seven Bridges of Königsberg" in 1736. There were seven bridges and four landmasses in the city of Königsberg. The problem was to design a walk through the city that traverses all the bridges *exactly* once. He abstracted this puzzle into a graph with four nodes representing landmasses and seven edges representing the bridges [BLW76].

Graphs have since then found varied applications in genome sequencing, web crawling, task scheduling, and road networks. The sizes of the graphs in these practical use cases can range from a few thousand to a few billion vertices. The human genome itself consists of nearly three billion base pairs, and graph-based representations used in genome assembly and variation analysis routinely involve billions of nodes and edges [CLJ+14]. The graph network corresponding to Facebook has more than three billion active users presently [Fac26].

Classical algorithms that analyze graphs make certain assumptions like space usage proportional to the input size and arbitrary (RAM) access to the input as desired. These assumptions are hard to justify for large graphs we want to process today because they would be very costly to execute. For instance, counting the number of triangles of a graph with seventeen million nodes and roughly a billion edges takes more than twenty minutes even with multiple GPUs [MDQ+18].

Many models of computation have arisen in theoretical computer science to formally address these access and space challenges to reflect real-world settings. Some of these models include the streaming model, distributed communication model, query model and massively parallel computation model (MPC). The focus of this thesis is the following two models:

Multi-pass Streaming Model: There may be situations where the input graph is stored on a disk, and is too large to fit into memory (RAM). It must be accessed via disk accesses, and the edges of the graph are stored in a sequential order on the disk. Multi-pass graph streaming [FKM+04] addresses this scenario: the graph is accessed in an arbitrary sequential order, and must be processed in limited space without storing it in its entirety. It may be accessed multiple times, corresponding to each sequential read from the disk [Vit01]. A variant of this model, called dynamic streaming, also captures situations where updates to the graph are stored externally and are accessed sequentially.

Distributed Computing Model: Imagine a situation where the graph is a network topology with routers being nodes and connections being edges. It is reasonable to expect the routers to only know their immediate connections, and not the entire network topology. This situation is captured by the distributed computing model: nodes of the graph operate with only local information, and they want to process the underlying graph by communicating with each other across many rounds [AY05, Pel00]. The communication may not be unlimited and could be restricted to the bandwidth of the underlying links.

While the time required to run algorithms is considered the primary resource in classical algorithms, in these models that process large data, the bottleneck may be different. In the streaming model, the number of passes over the input and the internal memory required are considered the main resources we want to optimize over. In distributed computing, the bandwidth and the number of rounds of communication are of interest. The two models may look different at the outset, but algorithms in the streaming and distributed models often employ similar techniques. In fact, progress made in one setting has been used to move algorithms forward in the other [KMVV13, FMU22, AKM22, FHM23, HS23].

The past few years have seen tremendous algorithmic progress for many crucial problems. Some examples of such results follow.

(*i*) Algorithms for finding matchings and vertex covers: in streaming [FKM+08, Kon18b, AG18, AJJ+22, FMU22, PS18, ABKL23, HW24]; in distributed computing [BEPS12, LPP15, BCGS17, AKO18, FFK21, IKY24].

(*ii*) Algorithms concerning distances and connectivity: in streaming [FKM+08, HKN16, ET22, BFKL21, AJJ+22]; in distributed computing [Nan14, Elk17, FN18b, BN19, GT24].

(*iii*) Algorithms for vertex coloring and edge coloring: in streaming [CDK19, ACK19a, AKM22, ACGS23, CCZ25]; in distributed computing [BKM20, GK21, HKMT21, BBKO22a, BBKO22b, BMdV26].

These algorithms, when compared to the naive benchmark of storing the entire input in the streaming model, or sending the entire input in the distributed computing model, perform extremely favorably. How far can we expect these algorithms to go in terms of optimizing the space and number of passes in graph streaming, or the number of rounds and amount of bandwidth in distributed computing? Certainly, as low as possible would be ideal, but it is unclear what "low" means in this context. This brings us to proving formal lower bounds about the capabilities of algorithms.

Importance of Formal Lower Bounds. Alongside the design of efficient algorithms, proving formal impossibility results on the feasibility of tasks with a limited amount of resources is crucial. These lower bounds have a two-fold purpose:

- What can we expect to achieve? Formal lower bound results facilitate informed algorithm design. They can help us set realistic goals for algorithms.
- Which inputs are hard? Constructing hard instances of inputs helps us understand why certain amounts of resources are needed. Perhaps, for smaller classes of inputs, better algorithms could be designed if we understand hard input instances well.

Proving lower bounds has been a main focus in theoretical computer science. For classical algorithms these lower bounds may take the form of hardness of approximation results [FGL+91, Sud09, Kho05, DS05], circuit lower bounds [KL80, Kan82, Has86, Smo87, Raz87, Wil14] or fine-grained complexity results [GO95, IPZ01, BI15, WW18]. For dynamic algorithms and the MPC model, lower bounds are often based on certain widely believed

conjectures like the OMv conjecture [AW14, HKNS15, AD16, JX22] or the 1vs2 cycle conjecture [YV18, RVW18, ABJ+25].

For the streaming and distributed models, what shape do these lower bounds take? *Unconditional*[1] results about the impossibility of achieving some tasks with a limited number of passes over the input or space (in the streaming model) [FKM+05, GO16, CKP+21a] or a limited number of rounds and amount of bandwidth (in the distributed model) [Lin92, KMW16, Suo20, ACKL20] can be proven. The one common tool underlying these lower bound results about space and bandwidth is communication complexity.

Communication complexity: There are players, say Alice and Bob, among whom the edges of the input graph $G := (V, E)$ are partitioned. Alice has a set $E_A \subset E$ and Bob has a set $E_B \subset E$, which are disjoint and partition the edge set E. They want to process the graph G by sending messages to each other.

The resource of interest in communication complexity is the number of messages Alice and Bob send to each other and the length of the messages. There are multiple other ways to model communication problems, based on the number of players, the number of rounds, and the order in which the messages are sent. The model is primarily used as a tool for proving lower bounds in other contexts like streaming [FKM+08, GKK12], distributed computing [SHK+12, CHKP17], data structures [MNSW98, KW20], property testing [BBM11], and circuit complexity [KRW95].

There has been exciting progress over the past few years in proving lower bounds for graph problems in both these models [DKO14, GO16, CHKP17, CK18, FGKO18, KK19, AR20, Kap21, CKP+21a] using communication complexity techniques. However, *optimal* results are not known for even fundamental problems like streaming matchings, streaming maximal independent sets, and distributed subgraph detection. There is a 'soft' barrier in terms of techniques to extend prior lower bounds to solve these open questions. The communication complexity framework used to prove lower bounds is algorithmically stronger, both in streaming and distributed computing. It may not lead to optimal lower bounds in a lot of cases.[2] This thesis aims to overcome these barriers by way of communication complexity models closer to graph streaming and distributed computing.

Aim: To develop a toolkit to prove optimal lower bounds for graph problems with more stringent models of communication.

The rest of this chapter will provide a bird 's-eye view of our technical contributions. The landscape of lower bounds before this thesis is presented in Chapter 2, and a more detailed overview of our techniques is presented in Chapter 3.

[1] Here, unconditional refers to information-theoretic impossibility results which do not rely on computational assumptions like P does not equal NP, or the Strong Exponential Time Hypothesis.

[2] See Sections 2.2 and 2.3 for more details on this issue and papers that overcome it for specific problems.

1.1 Contributions of this Thesis

We now present our contributions in lower bounds for graph problems via communication complexity tools. The techniques behind our results will be covered in Chapter 3. This section is just a compilation of the results, how they improve quantitatively over prior work, and a brief explanation on how they make progress on the aforementioned aim.

Our results can be broadly classified into two parts.

- Lower bounds via embedding arguments: a common theme underlying these results is the embedding technique. Hard instances for progressively larger $r > 1$ rounds are constructed by combining smaller $(r-1)$-round hard instances in a bottom-up fashion.
- Lower bounds beyond embedding: these results employ the more classic technique of reductions to communication complexity problems.[3] Nonetheless, they fit the theme of constructing round-specific hard instances, and borrow ideas from prior results.

Chapter 5: Maximal Independent Sets in Streams. We study the maximal independent set (MIS) problem in the semi-streaming setting. The study of MIS is motivated by practical applications like community detection [DKRM25] and document classification [SGB23], and numerous theoretical uses as a primitive in other algorithms for correlation clustering [ACG+15,BCMT22], minimum vertex cover [Vel24] and maximum matchings [ABK+25].

The problem of finding an MIS has been studied in many models like LOCAL [Lub85, KMW16,BBH+19,KS25], MPC, and Congested-Clique [GGK+18,Kon18a]. These algorithms use the vertex sampling approach dating back to Luby's parallel algorithm [Lub85], along with model-dependent insights and modifications. Semi-streaming algorithms (that use $\widetilde{O}(n)$ space) for MIS that run in $O(\log\log n)$ passes were given by [ACG+15] which runs the random greedy algorithm [YYI09][4] for MIS between batches of vertices.

In terms of lower bounds, in the standard two-party communication where Alice and Bob can send many messages to each other, finding an MIS is easy. The random greedy algorithm can be implemented with $O(n\log n)$ total communication. However, restricting the number of rounds makes the problem hard: single-pass algorithms in $o(n^2)$ space have been ruled out by considering one-way communication [ACK19b, CDK19].

Restricting the number of rounds to some $r > 1$ would lead to lower bounds for multi-pass algorithms but such results were not known as the standard techniques are not applicable directly.[5] We present a new technique called hierarchical embeddings and close this large gap by proving that the current algorithms are *optimal*:

Theorem 1. *Any semi-streaming algorithm for finding an MIS with constant probability of success requires* $\Omega(\log\log n)$ *passes.*

This settles the complexity of this fundamental problem in the semi-streaming model, and constitutes one of the first optimal multi-pass lower bounds in this model. Moreover,

[3] In fact, the techniques are simpler compared to Chapters 5 to 7, and give optimal or close to optimal bounds.

[4] This is similar to Luby's except for one detail: while Luby's uses fresh randomness in multiple rounds of $O(\log n)$, the random greedy algorithm uses one fixed random permutation of vertices for symmetry breaking.

[5] Refer to Chapter 3 for an explanation.

our result provides a pass-space tradeoff of the following form: any algorithm that uses p passes requires $n^{1+1/2^{\Omega(p)}}$ space, which is optimal for the entire range of p up to constant factors in the exponent [ACG+15].

The $O(\log\log n)$-pass algorithm of [ACG+15] for MIS was designed as part of a semi-streaming implementation of the so-called *Pivot* algorithm [ACN05] which computes the *random order MIS* to approximate *Correlation Clustering* (improving upon [CDK14]; see also [BFS12, FN18a]). In general, computing MIS, and, in particular, random order MIS, is a highly useful subroutine in various algorithmic problems; e.g. [YYI09, BFS12, BDH+19, Beh21, HZ23, CKL+24]. At the same time, this meant that for many of these problems, the best bounds remained stuck at the same $O(\log\log n)$ passes of the MIS.

There have been several attempts to bypass this "barrier" by relying on weaker subroutines than MIS. Most notably, for correlation clustering, [CLM+21, AW22] designed "non Pivot" algorithms that use an entirely different subroutine based on sparse-dense decompositions (at a cost of much larger approximations); and, [BCMT22, BCMT23, CKL+24] designed semi-streaming algorithms for more relaxed versions of Pivot that can be implemented in much fewer passes. Another example of similar nature is the $O(1)$-pass semi-streaming algorithm of [CKPU23] for a relaxation of MIS called the 2-*ruling set*[6], which can replace MIS sometimes, e.g., in metric facility location [BHP12] (see [KPRR19, AD21] for other streaming algorithms for ruling sets).

Our lower bound in Theorem 1 now definitively confirms the necessity of relying on these relaxations (and their accompanying complications), as there is provably no way of implementing *any* MIS-based algorithm in the semi-streaming model in fewer than $O(\log\log n)$ passes.

Theorem 1 appears in a paper co-authored with Sepehr Assadi, Christian Konrad and Kheeran K. Naidu in the proceedings of STOC 2024 [AKNS24a], and the SICOMP special issue of STOC 2024 [AKNS24b].

Chapter 6: Matchings in Streams. Matchings are perhaps the most widely studied problem in the graph streaming literature [McG05, GKK12, AGM12, AG18, Kap21, AS22, DK20, ALT21]. Matchings also have practical applications like task assignment [RXC+21] and kidney exchange [MKPV21]. We prove multiple results on matchings in the streaming model.

Understanding the approximation ratio achievable by multi-pass semi-streaming algorithms for matchings has been a sought-after goal since the introduction of the semi-streaming model by [FKM+05]. A long line of work since then [FKM+05, McG05, KMM12, Kap13, EKMS12, KT17, AG18, Kon18b, Tir18, ALT21, FMU22, AJJ+22] has culminated in semi-streaming algorithms with poly$(1/\varepsilon)$ passes for general graphs [FMU22] and $O(1/\varepsilon^2)$ passes for bipartite graphs [ALT21].[7]

The lower bound front, however, has seen much less progress with only a handful of

[6] For every $\beta \geq 1$, a β-ruling set of a graph G is an independent set such that all other vertices are within distance β of some vertex in the ruling set. Thus, an MIS is a 1-ruling set.

[7] There are also algorithms with pass-complexity with better dependence on ε at the cost of mild dependence on n, namely, $O(\log n/\varepsilon)$ passes in [AG18, AJJ+22, Ass23] or for finding perfect matchings in $n^{3/4+o(1)}$ passes [AJJ+22] (see also [LSZ20]).

results known for single-pass algorithms [GKK12, Kap13, AKL17, Kap21, AJX26] that rule out any approximation ratio larger than 0.55, and two-pass algorithms [KN21, Ass22, KN24] that rule out any approximation ratio larger than (8/9). No lower bounds beyond two-pass algorithms are known for constant factor approximation of matchings in the semi-streaming model.

We present a new lower bound for multi-pass semi-streaming algorithms for the estimation of maximum matching size. The lower bound is parameterized by the density of RS graphs that we mentioned in Section 2.2.3, namely, graphs whose edges can be partitioned into induced matchings of size $\Theta(n)$ (see Section 4.2.1). Let $\beta_{\text{RS}} \in (0,1)$ denote the largest parameter such that there exist n-vertex RS graphs with $\Omega(n^{\beta_{\text{RS}}})$ edge-disjoint induced matchings of size $\Theta(n)$. The current known values of β_{RS} are:

$$\Omega\left(\frac{1}{\log\log n}\right) \underset{[\text{FLN}^+02]}{\leq} \beta_{\text{RS}} \underset{[\text{FHS15}]}{\leq} 1 - \Theta\left(\frac{\log^* n}{\log n}\right),$$

Theorem 2. *Any (possibly randomized) semi-streaming algorithm for $(1-\varepsilon)$-approximation of even the size of maximum matchings requires*

$$\Omega\left(\frac{\log(1/\varepsilon)}{\log(1/\beta_{\text{RS}})}\right)$$

passes over the stream. The lower bound holds for the entire range of $\varepsilon \in [n^{-\Theta(\beta_{\text{RS}})}, \Theta(\beta_{\text{RS}})]$.

When $\beta_{\text{RS}} = \Omega(1)$, our result gives an $\Omega(\log(1/\varepsilon))$-lower bound on the number of passes. Theorem 2 appeared in FOCS 2023 and was co-authored with Sepehr Assadi [AS23a].

More recently, we proved the same result unconditionally for outputting a $(1-\varepsilon)$-approximate matching. This uses the same technique of hierarchical embeddings that was developed to prove Theorem 1. Underlying our result is a tradeoff between the number of rounds and the approximation factor for matchings when the message length is bounded. Prior to our work, no lower bounds even for three-round protocols in communication complexity were known.

Theorem 3. *Any (possibly randomized) semi-streaming algorithm that outputs a $(1-\varepsilon)$-approximation of maximum matchings requires $\Omega(\log(1/\varepsilon))$-passes over the stream.*

Although Theorem 2 and Theorem 3 are far from the upper bound of $O(1/\varepsilon^2)$, we view them as a first step towards proving multi-pass streaming lower bounds with a dependence on ε in the number of passes.

Theorem 3 is unpublished work that is in preparation to be submitted for peer review, and is co-authored with Sepehr Assadi, Christian Konrad and Kheeran K. Naidu.

Matchings in Dynamic Streams. A closely related model of interest is dynamic streams, which models situations where the graph is not static. The input is given as a sequence of edge insertions and deletions. We want to process the final graph after all the updates using limited space [AGM12]. This is suitable for scenarios where the updates to a graph are stored sequentially without being processed immediately due to memory constraints. We are still limited to semi-streaming space of $\widetilde{O}(n)$.

Strong separations are known between the insertion-only streaming model and dynamic streams. In one pass, it is known that finding any constant approximation in dynamic streams requires $\Omega(n^2)$ space [AKLY16, DK20], again by way of a one-round communication complexity lower bound. In insertion-only streams this can be trivially done by storing a maximal matching. In dynamic streams, the best algorithms that find a constant approximation of matchings use $\approx \log n$ passes [AGM12, AG15]. It was conceivable that the upper bound could be improved all the way to two passes.

Our result is the first multi-round lower bound for this problem:

Theorem 4. *Any randomized semi-streaming algorithm for $O(1)$-approximation of maximum matching in dynamic streams with constant probability of success requires $\Omega(\log \log n)$ passes. The lower bound holds even on (unweighted) bipartite graphs.*

We also prove a matching upper bound of $O(\log \log n)$-passes to find any constant approximation in the same work, showing Theorem 4 is tight, although we do not focus on the details of the upper bound in this thesis.

Theorem 4 is presented in [ABK+25], in the proceedings of SODA 2025. It is another application of the hierarchical embeddings technique developed in Theorem 1, where prior techniques failed to give any meaningful multiround lower bounds. This paper was invited to the TALG special issue of SODA 2025 papers.

Chapter 7: Distributed Triangle Detection. Now, we move to the distributed computing model and specifically look at the triangle detection problem. We work with the CONGEST model, where the bandwidth limit is $O(\log n)$. Detecting a triangle in this problem means that if the input G contains no triangles, *all* players should output *No*; and, if G contains a triangle, at least *some* player should output *Yes*.

The first algorithm for distributed triangle detection is due to [IG17] and achieves $\widetilde{O}(n^{2/3})$ rounds. This was subsequently improved in [CPZ19] to $\widetilde{O}(n^{1/2})$ rounds using expander decompositions, and further refined in [CS19, CPSZ21] to $\widetilde{O}(n^{1/3})$ rounds. The only lower bounds for this problem are due to [ACKL20] and [FGKO18] who, respectively, showed that deterministic or randomized algorithms cannot solve this problem in a *single* round. There are two reasons as to why we do not have stronger lower bounds:

(*i*) There is a provable barrier for establishing strong lower bounds: [EFF+22] shows that a lower bound of n^{δ} rounds for any constant $\delta > 0$ implies circuit complexity lower bounds that are entirely out of reach of existing techniques.

(*ii*) The two-party communication complexity technique inherently fails to give any meaningful lower bounds as a triangle is comprised of three vertices. In any partition of the three vertices, at least one of the two players comes to know the existence of a triangle trivially.

The circuit complexity barrier does not exist for proving an $n^{o(1)}$-round lower bound, and no lower bound even for two rounds was known. Again, our result gives the first multiround lower bound for this problem:

Theorem 5. *Any algorithm for distributed triangle detection in CONGEST on n-vertex graphs requires $\Omega(\log \log n)$ rounds to succeed with high constant probability.*

Our model of communication is very close to the CONGEST framework: each vertex is treated as a separate player and overcomes the two-party issue. The embedding techniques of Theorems 1, 3 and 4 heavily influence our result, although Theorem 5 is not a direct application of hierarchical embeddings.

In addition, we perform a slight optimization of the parameters in [EFF+22], and show that even an $n^{\Omega(1/\log \log n)}$ lower bound for distributed triangle detection breaks circuit complexity barriers. Moreover, the proof of [EFF+22] relies on algorithms for expander decomposition and routing in the CONGEST model, which may conceivably become much faster [CMGS25]. If such fast algorithms exist, even an $\Omega(\log^c(n))$-round lower bound for some large constant $c > 1$ for triangle detection could lead to breaking these circuit complexity barriers. Hence, we believe that realistically, the best lower bound one can hope to prove for triangle detection is $\Omega(\log n)$, leaving the gap between Theorem 5 and the target only an exponential one instead of a doubly exponential one.

Theorem 5 appeared in the proceedings of FOCS 2025 and is co-authored with Sepehr Assadi [AS25]. It was invited to the SICOMP special issue of FOCS 2025.

Chapter 8: Chained Index. The Chained Index communication problem (CHAIN) was introduced by [CDK19] as a way to prove lower bounds for finding independent sets in vertex arrival streams. It can be viewed as the INDEX with multiple tries afforded to get at the same answer.

For $n, k \geq 1$, in $\text{CHAIN}_{n,k}$, there are k instances of INDEX_n, all with the same answer, distributed among $(k+1)$ players. Players 1 and 2 take on the role of Alice and Bob respectively in the first instance, players 2 and 3 take on the role of Alice and Bob respectively for the second instance, and so on, all the way till players k and $k+1$ for the last instance.

Communication is one-way from each player to the next in ascending order. The last player has to output the answer. The communication cost is the total number of bits in all the messages sent by the players.

A trivial protocol where one player sends the entire input takes $O(n)$ bits. In [CDK19], they prove a lower bound of $\Omega(n/k^2)$ for any $k \geq 1$ through a reduction from the conservative multi-party pointer jumping problem, introduced by [DJS96]. They also conjecture that the trivial protocol is tight for any k. [FNFSZ20] made some progress on proving this conjecture, showing that among all the messages sent by the players, there is at least one message with $\Omega(n/k^2)$ bits for every $k \geq 1$.

We prove a tight lower bound for a very large range of $k = o(n/\log n)$, almost settling the conjecture.

Theorem 6. *For any $n, k \geq 1$, any protocol for $\text{CHAIN}_{n,k}$ with probability of success at least $2/3$ requires $\Omega(n - k \log n)$ total bits of communication.*

Theorem 6 is obtained via a multi-round perspective: we keep track of exactly how much information is revealed about the answer by each message sent to the blackboard. Theorem 6 is from a solo-authored paper in the proceedings of ITCS 2025 [Sun25].

Chapter 9: Shortest Paths in Streams. The problem of finding shortest paths in the streaming model was started by [FKM+04], which presents a single-pass semi-streaming algorithm for $O\left(\frac{\log n}{\log\log n}\right)$-approximation of single-source shortest paths, which was shown to be optimal soon after in [FKM+05]. Multi-pass algorithms are known to improve over this approximation ratio, and even give constant approximations. After a long line of work [EZ04, EP04, GO16, HKN16, EN19, BFKL21, AJJ+22], the state-of-the-art for constant approximation in semi-streaming is as follows:

- polylog(n)-pass upper bounds for arbitrarily small constant approximation ratios, [BFKL21, AJJ+22] (at least a $\log^7 n$ factor) and
- $\omega(1)$-pass lower bounds for sufficiently small constant approximation ratios [GO16].

We improve the lower bound significantly:

Theorem 7. *For any constant $\alpha \geq 1$, any semi-streaming algorithm on unweighted n-vertex graphs that, with constant probability of success, can output an α-approximation to the distance between a fixed pair of vertices requires $\Omega(\log n/\log\log n)$-passes.*

We also complement Theorem 7 with an upper bound of $O(\log^2 n)$-passes for any constant approximation, bringing the gap down to an $O(\log n)$ factor from $\operatorname{poly}\log n$ factors. Theorem 7 can be seen as an extension of the lower bound of [GO13] for exact distances to constant approximation factors.

Theorem 7 is from a paper co-authored with Sepehr Assadi and Gary Hoppenworth from SODA 2026 [AHS26].

Chapter 10: Hypergraph Edge Cover in Streams. Our last result is on the hypergraph edge cover problem, equivalent to set cover. In the streaming set cover problem, m sets from a universe $\mathcal{U}$ of size n are arriving one by one in a stream, and the algorithm is allowed to process the stream using one or a few passes and a space of $o(mn)$, which is sublinear in the input size. The goal is to determine the minimal (or approximately minimal) number of sets that cover the universe at the end of the last pass.

This problem has been studied extensively over the years, leading to several $O(\log n)$ approximation algorithms in $\widetilde{O}(mn^{1/p})$ space and p passes [HIMV16,BEM17,IMR+17,Ass17], which has remained the state-of-the-art. The work of [HIMV16] conjectured that: "this tradeoff might be tight, at least for 'low enough' approximation factors".

We establish an *optimal* three-way **space-pass-approximation** tradeoff for this problem, proving the conjecture of [HIMV16].

Theorem 8. *Any p-pass α-approximation algorithm, whenever $\alpha \ll n^{1/(p+1)}$, for streaming set cover requires*

$$\widetilde{\Omega}\left(\frac{m}{\alpha}\cdot\left(\frac{n}{\alpha}\right)^{1/p}\right)\text{space}.$$

In light of prior work, this result is optimal up to constant factors in p and up to logarithmic factors in n for any p and $\alpha \geq p$. Our bound is also optimal with respect to the range

of α [Ass17], and fully settles the complexity of this problem in the streaming model. This is perhaps our simplest result and uses a direct reduction from a well-known communication problem called pointer chasing.

Theorem 8 is from a paper co-authored with Sepehr Assadi, and will appear in the proceedings of STOC 2026 [AS26].

1.1.1 Organization of this Thesis

This thesis is divided into three parts. The first part contains the introduction, a detailed overview of the state of affairs in graph streaming and distributed computing lower bounds, an overview of our techniques and some technical preliminaries.

The second part contains the results that embed smaller instances to construct larger instances. More specifically, hard instances for larger $r > 1$ are constructed recursively using smaller $(r-1)$-round instances. The proof of the lower bounds is via the celebrated round elimination technique dating back to [MNSW95].

The third part has the results that use classic techniques of reductions to communication complexity problems to prove lower bounds. The two problems we use are Index [Abl93, KNR95] and Pointer Chasing [PS84]. This part also contains a discussion of future directions.

Reading this Dissertation. A common technical overview of our results is presented in Chapter 3. The basic concepts of information theory and combinatorial graph constructions that we use in our results can be found in Chapter 4. We suggest the reader use Chapter 4 as a reference while reading Chapter 3 for any new terms or constructions they encounter. The conclusion and open problems are given in Chapter 11.

The rest of the chapters in the second and third parts contain all the papers and the technical proofs. These chapters are self-contained and can be read in any order.

2 Landscape of Prior Lower Bounds

In this chapter, we cover the prior research and techniques in proving lower bounds for graph problems.

2.1 General Structure of Graph Lower Bounds

Lower bound proofs typically begin with the construction of hard instances, and a proof as to why all algorithms fail to solve these instances with restricted resources. We can broadly split the structure of lower bounds into two major parts.

(*i*) Combinatorial graph constructions: the first step is identifying what kind of graphs are hard to begin with. This part of the lower bounds predominantly involves using combinatorial tools to construct appropriate graphs.

(*ii*) Information-theoretic analysis of constructions: Once hard instances are identified, they may possess certain properties which could be used to prove that the instances are impossible to solve with a limited amount of resources. This part of the proof typically uses information-theoretic arguments to prove these impossibility results.

We give a specific simple example that highlights these two parts. The input is a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, where the vertices are known and the edges are not known (this is the standard graph streaming model introduced by [FKM+04]). The edges are sent in a stream in an adversarial order, and two vertices $s, t \in V$ are given in the beginning of the stream of edges, and we want to output the shortest path distance between s, t after all the edges are seen. The graph must be processed in a limited space $s = o(m)$: without storing all the edges. We can prove that this is impossible [FKM+04], i.e., one *needs* $\Omega(n^2)$ space to find the exact shortest path distance.

Combinatorial Construction. We begin by constructing a graph instance where the algorithm has no choice but to remember $\Omega(n^2)$ edges. Let the vertex set V be:

$$V = A \cup B \cup \{s, t\}$$
$$A = \{a_1, a_2, \ldots, a_{n-2}\}$$
$$B = \{b_1, b_2, \ldots, b_{n-2}\}.$$

There will be no edges inside vertex set A or inside vertex set B. However, any pair a_i, b_j for $i, j \in [n-2]$ can have an edge between them. These edges, which are a subset of $A \times B$, appear at the beginning of the stream. After all such edges are sent, edges to s, t may be added so that the shortest path distance between s, t can determine whether edge $(a_{i^\star}, b_{j^\star})$ exists for *any* single choice of $i^\star, j^\star \in [n-2]$. We can simply add edges $(s, a_{i^\star})$ and $(b_{i^\star}, t)$ to the end of the stream.

Refer to Figure 2.1 for an illustration. The shortest path distance will be three if the edge exists between $(a_{i^\star}, b_{j^\star})$ and at least five otherwise.

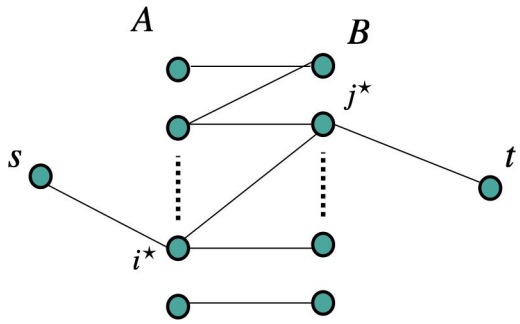


Figure 2.1: Figure depicting the hard instance for shortest path. The path between s and t has length 3 if and only if an edge exists between $i^\star$ and $j^\star$. Otherwise, the path length is at least 5.

Information-theoretic Analysis. In the graph from Figure 2.1, intuitively, it is easy to see why the algorithm may need $\Omega(n^2)$ space: each pair (a_i, b_j) for $i, j \in [n-2]$ may become the important edge, and we have to retain a majority of them to be able to answer the exact shortest path distance effectively. How would we formally prove this statement? As mentioned earlier, communication complexity is the main tool used in theoretical computer science to prove unconditional lower bounds of this form.

For our purpose, the INDEX problem can be used. It is one of the foundational problems in communication complexity. For $k \geq 1$, in the INDEX_k problem, there are two players Alice and Bob. Alice has a string $x \in \{0,1\}^k$ and Bob has an index $i \in [k]$, and Bob has to output the value of x at position i. If the communication is one-way from Alice to Bob, it is easy to show that Alice needs to send $\Omega(k)$ bits to get any success probability strictly better than $(1/2)$ by a constant [Abl96, KNR95]. Refer to Figure 2.2 for an illustration.

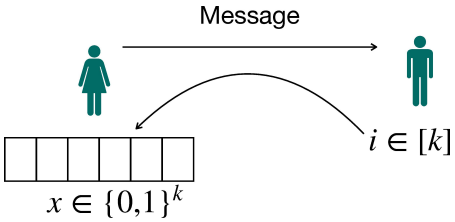


Figure 2.2: Index problem. Alice has a string $x \in \{0,1\}^k$. Bob has an index $i \in [k]$ that points to x. Alice can send a single message to Bob, who has to output value of x at position i.

The lower bound on the Index problem can also be used to prove a lower bound on the space required by the streaming algorithm. We can interpret whether or not each pair (a_i, b_j) is an edge as one of the bits of Alice, and Bob's index will choose one of these pairs via edges to s, t. [1]

This theme of a combinatorial component and an information-theoretic component will be prevalent throughout this thesis in all the technical results. Now that we have a basic

[1] In fact, there is a reduction from a streaming algorithm for exact shortest path to a protocol that solves the Index problem. We will talk about these reductions in Section 2.2.

idea of what these lower bounds look like, we give a more comprehensive overview of the known results. Lower bounds in the streaming model and distributed computing are not too related beyond the fact that they both use tools from communication complexity, so we cover them separately.

2.2 Streaming Lower Bounds

The prior results in graph streaming are almost exhaustively proved using communication complexity, and range from direct reductions to novel information-theoretic tools.[2]

2.2.1 Role of Communication Complexity

Communication complexity lower bounds give rise to streaming lower bounds, and this was observed in the seminal paper [AMS99]. More specifically, streaming algorithms give rise to communication protocols. In the streaming model, input may be adversarially ordered, and the algorithm has limited space to process the input. In communication complexity, the input is *partitioned* across many players. They want to process the input by sending small length messages to each other.

The players can simulate the streaming algorithm through the messages simply by running the algorithm on their respective inputs, and sending the memory state of the algorithm to each other back and forth. If the algorithm is run for multiple passes, many messages could be necessary. Proving a lower bound on the length of messages immediately implies lower bounds on the space complexity of the streaming algorithm. See Figure 2.3 for an illustration.

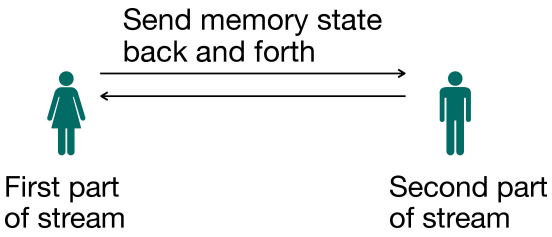


Figure 2.3: Streaming algorithms give communication protocols. The players run the algorithm on their inputs and send the memory state to the other player as messages.

Certain canonical hard problems in communication complexity like Index [Abl96,KNR95], Pointer Chasing [PS84,NW91,Yeh16] and Set Disjointness [KS92,Raz92,BJKS02] have been used to prove streaming lower bounds by way of reductions. The graph problem of interest is used to encode the communication problem.

Concretely, let Alice and Bob be two players who want to solve problem $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. Alice is given some input $x \in \mathcal{X}$ and Bob is given some input $y \in \mathcal{Y}$. Alice and Bob collectively create a graph $G = (V, E_A \cup E_B)$ that encodes $(x, y) \in \mathcal{X} \times \mathcal{Y}$, such that the streaming algorithm that process graph G also solves f. Alice can find edges E_A only using input x and Bob finds E_B only using y, without any communication between each other.

[2]More details can be found in the surveys [McG14, Ass23].

One pass of the algorithm can be simulated with one message from Alice to Bob, and another message from Bob to Alice. Multiple passes require multiple rounds of communication between Alice and Bob. See Figure 2.4 for a depiction of the simulation.

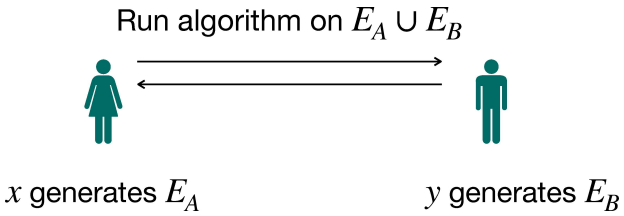


Figure 2.4: Some communication problems can be reduced to graph streaming problems.

2.2.2 Reductions to Index, Pointer Chasing and Set Disjointness

The canonical communication complexity problems that have been used to prove graph streaming lower bounds are Index, Pointer Chasing and Set Disjointness.

Index and Pointer Chasing. The first set of results that prove lower bounds for graph problems in the streaming model rely on the information-theoretic hardness of the Index problem [Abl96, KNR95] and the Pointer Chasing problem [PS84, Yeh20].

The result covered in Section 2.1 is an example of a graph streaming lower bound that relies on the hardness of the Index problem. Other examples include a lower bound of $\Omega(n^2)$ for finding an exact maximum matching [FKM+05], directed s-t reachability in one pass [FKM+05], approximating the number of triangles by a constant factor [BOV13] and finding a maximal independent set [ACK19a, CDK19]. These results show that no non-trivial streaming algorithm exists for these problems, as the space required to store the entire graph is $O(n^2)$.

What about problems where non-trivial $o(n^2)$ algorithms exist? Many problems like testing connectivity, bipartiteness and finding the cost of an exact MST in graphs with poly(n) edge weights admit $O(n \log n)$ space algorithms in a single pass.[3] Again using Index, an $\Omega(n)$ lower bound was established for these problems by [FKM+08], which was later improved to the optimal $\Omega(n \log n)$ lower bound by [SW15] using a slight variant of Index called the Perm problem.[4]

All of the aforementioned results *only* hold for single-pass algorithms due to a very simple reason: the Index problem is only hard with one-way communication. There is a trivial protocol with $O(\log n)$ bits if Bob can send messages to Alice, as Bob can just send the index [MNSW95]. The Pointer Chasing problem, introduced by [PS84], can be viewed as a multiround extension of Index. There are two sets A and B, each of size ℓ. Alice has

[3] The formal semi-streaming model with $O(n \cdot \text{poly} \log n)$-space was introduced by [FKM+05] as the sweet spot where non-trivial algorithms exist for many problems. Moreover, $o(n)$-space algorithms are provably impossible for many basic questions like computing a spanning tree and bipartiteness testing.

[4] Some problems, like parameter estimation, property testing or general problems on restricted classes of graphs, admit $o(n)$-space algorithms also [CCHM15, CJMM17, HP19, CCEW23]. There is a line of work studying lower bounds in sublinear-in-n space regime [VY11, KKS15, AKSY20, AN21, CKP+23], relying on the information-theoretic hardness of gap cycle counting. As the technical details of gap cycle counting is orthogonal to the rest of this thesis, we refer the reader to these papers.

a function f_1 from A to B and Bob has a function f_2 from B to A. Let z_0 be an arbitrary element in A known to both Alice and Bob. [5] We define:

$$z_1 = f_2(z_0), z_2 = f_1(z_1), z_3 = f_2(z_2), z_4 = f_1(z_3), \ldots$$

These values $z_0, z_1, \ldots, z_k$ for any $k \in \mathbb{N}$ are found by following the functions starting from z_0, and alternating between Bob and Alice. Both players want to know the value of z_k. Alice begins speaking, and they can exchange messages with each other over many rounds. Refer to Figure 2.5 for an illustration.

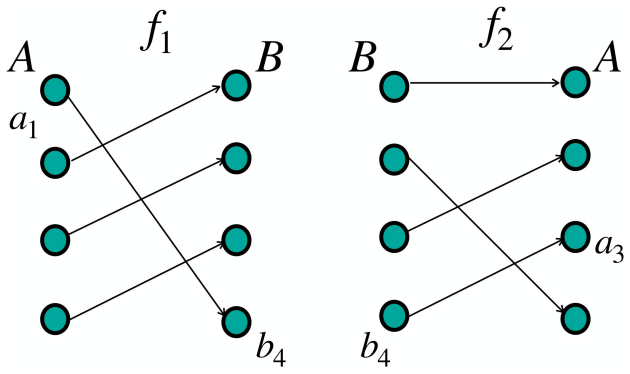


Figure 2.5: An illustration of the functions in pointer chasing. Alice holds f_1 and Bob holds f_2. If $z_1 = a_1$, $z_2 = b_4$ and $z_3 = a_3$.

If $(k+1)$ messages can be sent in total, Alice and Bob can find the value of z_k in $O(k \log \ell)$ length messages, by simply following the pointers from f_1 and f_2. However, if only k messages can be sent, it has been proved that the message length must be $\Omega(\ell/k)$ [NW91, Yeh20]. This is close in spirit to the streaming model: the hardness of the problem is rooted in the minimal number of passes over the input.

Pointer Chasing has been used to prove multi-pass streaming lower bounds. [CW16] used a direct reduction to prove a tight lower bound of $n^{1/(p+1)}$ against p-pass algorithms for hypergraph edge cover in $\widetilde{O}(n)$[6] space for any $p \geq 1$. A direct sum version of pointer chasing, where the players are required to solve t-many independent versions of pointer chasing *simultaneously*, is known to require $\Omega(t \cdot (\ell/k))$ length messages [JRS03]. This version, with $t = n^{\Omega(1/p)}$ and $\ell = \Omega(n/\text{poly}(k))$ was used by [FKM+08] to prove that any p-pass streaming algorithm for finding undirected shortest paths or directed reachability paths requires $n^{1+\Omega(1/p)}$.

In place of a direct sum where *all* t instances of pointer chasing have to be solved, [GO16] introduced the set chasing problem, where it is sufficient to solve the OR of t many pointer chasing instances. They proved that the set chasing problem also requires $\tilde{\Omega}(t \cdot (\ell/k))$ message length, and used a reduction to prove similar tradeoffs of p-passes requiring $n^{1+\Omega(1/p)}$-space for decision versions of undirected shortest path, and directed reachability. Set chasing was used

[5] Many versions of pointer chasing exist, and we use the simplest two player version to illustrate the problem here.

[6] We use $\widetilde{O}(f), \tilde{\Omega}(f)$ to hide logarithmic factors in f.

by [HIMV16] to prove a lower bound of $mn^{\Omega(1/p)}$ against p-pass algorithms computing the exact cost of hypergraph edge cover. Similar arguments as [GO16] were used by [CKP+21b] to prove a lower bound for sampling random walks in directed graphs.

We mentioned that the reductions from the Index problem are optimal for single-pass algorithms for many standard problems [FKM+05, FKM+08, SW15]. In contrast, the reductions from pointer chasing are not optimal even for two passes for problems like directed reachability and shortest paths [FKM+08, GO16].[7] The hard instances only have $n^{1+\Theta(1/p)}$ edges: $t = n^{\Theta(1/p)}$ many "random" looking graphs of constant degree are superimposed to create the input. There are two technical reasons behind this construction:

(i) The smaller sparse graphs of constant degree are derived from individual pointer chasing instances, and the hard inputs to pointer chasing are uniform functions [NW91, Yeh20] .

(ii) The direct sum arguments of [JRS03, GO16] rely on the independence of the t many individual instances: they cannot be correlated with each other.

This leaves us with hard instances that are almost random graphs with degree $n^{\Theta(1/p)}$. These instances only form easy cases of the respective problems in [FKM+08, GO16]. Refer to Figure 2.6 for an illustration.

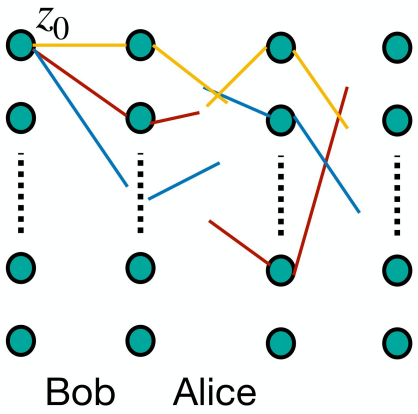


Figure 2.6: An illlustration of direct sum of pointer chasing. Each color represents one independent instance. Bob holds all the functions in one column, and Alice in the next and so on.

Thus, we do not know any way of extending the techniques of [FKM+08, GO16] to get stronger lower bounds. The current state-of-the-art upper bounds for exact undirected shortest paths and directed reachability are $O(\sqrt{n})$ [CFHT20, BFKL21]. We now move to another technique for proving multi-pass streaming lower bounds.

Set Disjointness. Index and Pointer Chasing are hard only for limited rounds of communication, and become trivially easy even if one extra round is given to the players. In

[7] The exceptions are [CW16] which is optimal for all passes, and [CKP+21b] which is optimal for two passes.

the other extreme is the set disjointness problem, which is proven to be hard, even with unbounded rounds of communication. Here, Alice and Bob have a set $A, B \subset [\ell]$. They simply want to find if $A \cap B = \emptyset$ or not, and we know that, for any $r \geq 1$, this problem requires $\Omega(\ell/r)$ message length if at most r messages can be sent [KS92, Raz92, BJKS02].

Reductions from set disjointness have been successful in proving lower bounds for maximum clique or independent set approximation [HSSW12], girth computation [FKM+08], triangle detection [BOV13, BC17], and finding minimum vertex cover [ACKP21]. These lower bounds are tradeoffs of the form p-passes versus $\Omega(n^c/p)$ space for some constant $1 \leq c \leq 2$. Thus, poly(n)-pass lower bounds for algorithms using poly(n) space can be shown.

Communication complexity with unbounded rounds is algorithmically much stronger than the streaming model. This captures the setting when there is no limit on the number of passes, and we are only interested in the total space used across all the passes, whereas in the standard streaming model, we want to minimize the number of passes and the space used by the algorithms. Hence, we can use this technique to prove lower bounds for problems which have high unbounded round communication complexity only. Directed reachability and maximal independent sets have trivial $O(n)$ total communication protocols. Approximation of maximum matchings also has a protocol that has total message length as $O(n \log^2 n)$ [BvdBE+22].

Next, we look at technical advances that overcome the challenges associated with direct reductions from index, pointer chasing and set disjointness.

2.2.3 New Techniques for Stronger Lower Bounds

There have been many breakthrough results that go beyond direct reductions, and these have been focused on semi-streaming algorithms which use $\widetilde{O}(n)$ space. We can broadly classify these results in two directions.

Strong Streaming Lower Bounds but Low Communication. Given the caveat that strong poly(n)-pass lower bounds for semi-streaming algorithms only exist when the communication complexity is large, the work of [ACK19a] asks the following question:

> *Is there a problem that has $\widetilde{O}(n)$ communication protocols, but needs $n^{\Omega(1)}$-passes in semi-streaming?*

[ACK19a] answers this question in the affirmative. They show that the lexicographically first maximal independent (LFMIS) set problem, in p-passes, requires $\Omega(n^2/p^5)$ space. A maximal independent set (MIS) is any independent set that is not a proper subset of some other independent set. LFMIS is a particular MIS that is picked by choosing the vertex with the lowest ID at each step, removing its neighbors and recursing on the remaining graph. This simple algorithm can be run in $O(n \log n)$ communication in unlimited number of rounds.

The main technique of [ACK19a] is a new communication problem called hidden pointer chasing (HPC). It can be viewed as a composition of set intersection and pointer chasing:

instead of the functions given to Alice and Bob in the standard pointer chasing instance, the output of the function is hidden via a set intersection instance between two players.

In the set intersection problem, there are two players Alice and Bob and a universe $\mathcal{W}$. Alice and Bob each have a set A and B respectively, such that $A, B \subseteq \mathcal{W}$ and $A \cap B = \{w\}$ for some $w \in \mathcal{W}$. Alice and Bob have to find the value of w. It is known that this problem requires $\Omega(|\mathcal{W}|)$ communication irrespective of the number of rounds [KS92, Raz92, BJKS02].

In the HPC problem, there are four players P_A, P_B, P_C and P_D, and two disjoint universes $\mathcal{X} = \{x_1, \ldots, x_\ell\}$ and $\mathcal{Y} = \{y_1, \ldots, y_\ell\}$. P_A, P_B together take on the role of Alice in pointer chasing, except that each pointer is hidden with a set intersection instance. For each $x \in \mathcal{X}$ players P_A, P_B are given a set intersection instance A_x, B_x, with a unique intersecting element $t_x \in A_x \cap B_x$, and the function $f_{AB} : \mathcal{X} \to \mathcal{Y}$ is defined as $f_{AB}(x) = t_x$. Similarly, P_C, P_D take on the role of Bob with $|\mathcal{Y}|$ many set intersection instances over $\mathcal{X}$, each of which points to an element in $\mathcal{X}$. The function f_{CD} is used to denote these pointers.

The sequence $z_0, z_1, \ldots, z_r$ is defined the same as in pointer chasing:

$$z_0 = g, \qquad z_1 = f_{CD}(z_0), \qquad z_2 = f_{AB}(z_1), \qquad z_3 = f_{CD}(z_2), \ldots$$

where g is an arbitrary element in $\mathcal{Y}$. The players want to compute z_r.

If $(r+1)$ messages can be exchanged from $\{P_A, P_B\}$ to $\{P_C, P_D\}$, solving this problem is easy, and can be done in $O(r\ell)$ communication. The players can solve the corresponding set intersection instances in $O(\ell)$ total communication. However, [ACK19a] shows that if only r messages can be exchanged across $\{P_A, P_B\}$ and $\{P_C, P_D\}$, the communication required is at least $\Omega(\ell^2/r^2)$. This "bounded round" aspect allows [ACK19a] to prove a lower bound of $\Omega(n^2/p^5)$ space for p-pass algorithms for LFMIS.

Follow-up work by [AGL+24] improves the lower bound of HPC to $\Omega(\ell^2)$ when different set intersection instances are used for each value of z_i for $i \in [r]$, with a total of $r\ell$ instances. They give a novel combinatorial construction which yields a lower bound of $\tilde{\Omega}(n^2/p^3)$ for degeneracy and k-core computation. A k-core of graph G is a maximal vertex subset $S \subset V$ such that each vertex in S has at least k other neighbors in S, and the degeneracy of G is the largest integer $k \geq 0$ such that G contains a non-empty k-core.

[AGL+24] also gives an $\widetilde{O}(n)$ communication protocol that computes the degeneracy and the k-core, answering the earlier question definitively. The combinatorial component of these results is the reduction from HPC and the information-theoretic component is the communication lower bound for the HPC problem. The next direction focuses on few passes and aims to get optimal lower bounds for small number of passes.

Smaller Independent Subgraph Instances. [FKM+08, GO16] prove an $\Omega(\log n / \log\log n)$ lower bound on the number of passes required to solve s, t reachability. They use a direct sum version of pointer chasing and to construct a graph which contains all the independent instances of pointer chasing, they simply overlap the edges from each instance (see Figure 2.6). What if we want to capture independent instances of other complex constructions? As we will see, a family of graphs called Ruzsa-Szemerédi-graphs [RS78] can be used for this purpose.

The use of RS-graphs in proving streaming lower bounds was pioneered by [GKK12]

for single pass algorithms who give an upper bound of 2/3 on the approximation ratio of maximum matchings in $\widetilde{O}(n)$ space, and by [AR20] for algorithms using more than one pass.

More concretely, [AR20] proves that any algorithm that checks if $t \in V$ is reachable from $s \in V$ in two passes requires $\Omega(n^{2-o(1)})$ space, this is almost quadratic, and close to optimal! In one pass, a lower bound of $\Omega(n^2)$ is easy to see (Figure 2.7) [FKM+05].

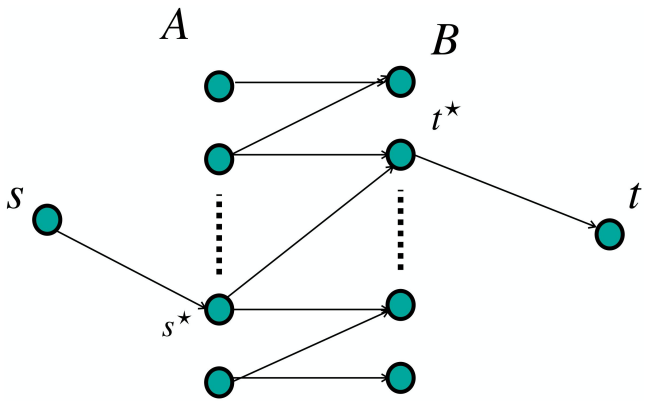


Figure 2.7: An illustration of the hard instance of s, t reachability for one pass algorithms.

[AR20], similar to HPC, uses the set intersection problem to hide the pointers from s, t even after one pass over the edges. For parameters p, q, they construct q many set intersection instances over a universe of size p each, to hide the pointer from s to $s^\star$ and from t to $t^\star$. See Figure 2.8. Among the q-many set intersection instance, *which* one reveals the pointer $s^\star$ and $t^\star$ is revealed only at the end of the first pass, and therefore, an algorithm would have to solve all of them using $\Omega(pq)$ space. This can be proven using direct sum arguments.[8]

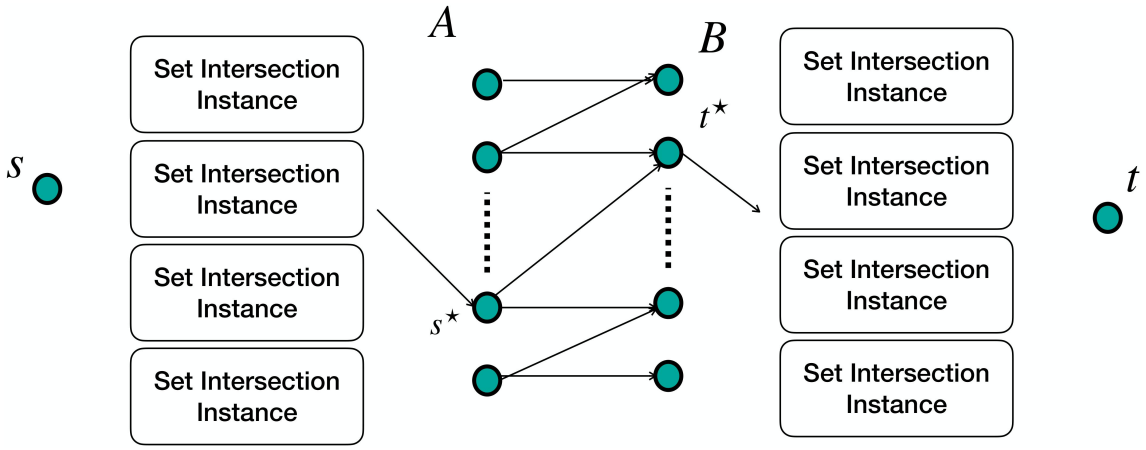


Figure 2.8: An illustration of using set intersecion instances to hide the pointers in one pass reachability.

This plan would allow one to prove the following lower bound for two-pass algorithms

[8] [AR20] proves something stronger: for each set intersection instance, the distribution of the answer remains ϵ-close to uniform in total variation distance with $o(\epsilon^2 \cdot q)$ length messages.

that find whether t is reachable from s:

$$\Omega(1) \cdot \min\{\underbrace{p \cdot q}_{\text{set intersection instances}}, \underbrace{q^2}_{\text{index instance}}\}$$

Taking $p = \Theta(q)$ gives an $\Omega(q^2) = \Omega(n^2)$ lower bound. However, there is one important caveat here. How do they capture the set intersection instances as a graph input? Importantly, the independence of the set intersection instances have to be preserved, while using only $O(q)$ many vertices in total to embed the instances.

The survey by [McG05] raises this issue prevalent in proving graph lower bounds:

> *less is known about graph stream lower bounds because it is often harder to decompose graph problems into multiple simpler "independent" problems and use existing communication complexity techniques*

[AR20] shows that Ruzsa-Szemerédi (RS) graphs [RS78] can be used to construct graphs that embed q many set intersection instances over universe of size p. They are parametrized by p, q and are a family of graphs whose edges can be partitioned into q many induced matchings of size p each. In an induced matching, there are no other edges between the end points of the matching edges.

The edges of each induced matching can embed one set intersection, and the *inducedness* of the matchings makes sure that different instances do not interfere with each other. More explicitly, they can be used to hide set intersection instances over universe $[p]$, which is the size of each matching. The end points of the matching are arbitrarily mapped to $[p]$. From s, there are edges corresponding to one set of the set intersection instance, and edges to t corresponding to the other set. Refer to Figure 2.9 for more details.

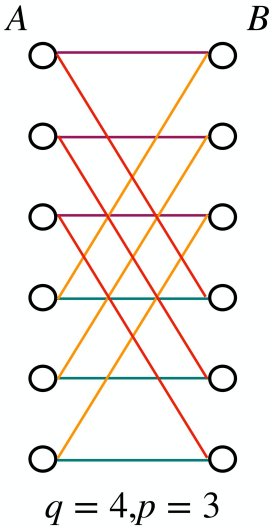
The framework of [AR20] was extended to many passes using by [CKP+21a] who show that any streaming algorithm that uses even $o(\sqrt{\log n})$ passes and computes whether t is reachable from s requires $\Omega(n^{2-o(1)})$ space. The framework was also extended to provide a lower bound on the approximation ratio of matchings in two passes and semi-streaming space ($\widetilde{O}(n)$ space) based on the density of Ruzsa-Szemerédi-graphs [Ass22].

The work by [KN24] uses RS-graphs to prove an unconditional upper bound of 8/9 on the approximation ratio of matchings in $\widetilde{O}(n)$ space, by directly extending the framework of [GKK12], and embedding "smaller" RS-graphs through induced matchings. Roughly speaking, each edge is replaced by an RS-graph which is a hard instance for one-pass algorithms.

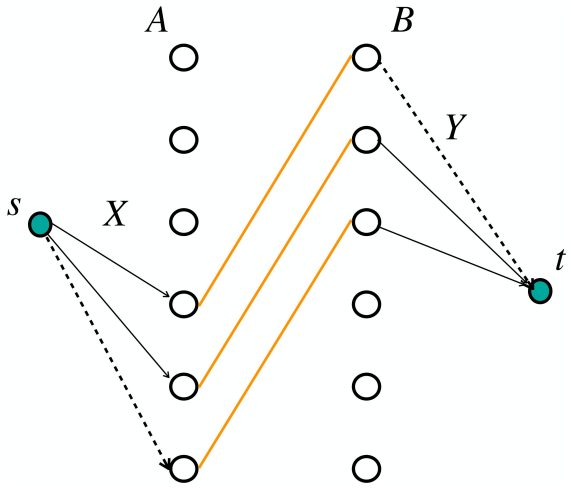
This concludes our overview of the prior work on graph streaming lower bounds. Let us move to the lower bounds for graph problems in distributed computing.

2.3 Distributed Computing Lower Bounds

We start by defining the distributed computing model. We have an n-vertex graph $G = (V, E)$ representing a distributed network with one player for each vertex of the graph who sees the edges incident on the vertex. Players communicate in synchronous rounds over



(a) An example of an RS graph with 4 matchings of 3 edges each.



(b) Hiding one set intersection instance $X, Y \subset [p]$ with one matching. Vertices t is reachable from s if and only if the endpoints of at least one edge in the matching is present in both the sets.

Figure 2.9: Using RS graphs to hide set intersection instances.

the edges of the graph. In the LOCAL model, the players can send arbitrary messages of unlimited length over the edges in each round. In the CONGEST model, the message length is restricted to $O(\log n)$ over each edge.

The players want to process the underlying graph, and after all the rounds are over, they output the required answer in a distributed manner.[9]

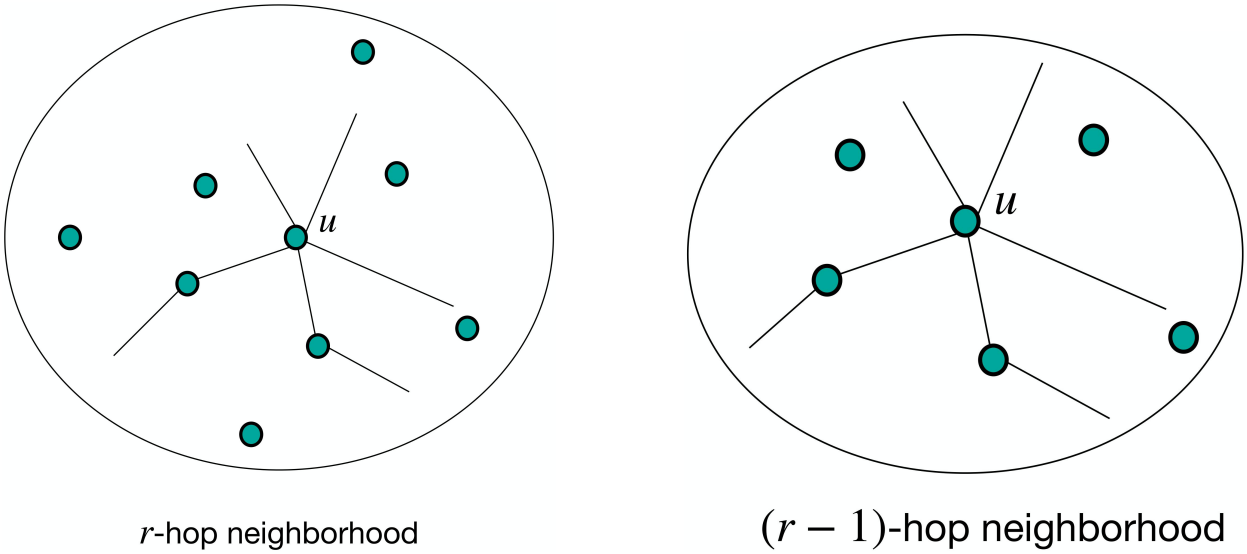
2.3.1 LOCAL Lower Bounds

LOCAL lower bounds are typically not proven using communication complexity, and there are a lot of other techniques that can yield optimal results. There is a rich history of work in proving lower bounds in the this model starting from [Lin92] which proves a tight lower bound of $\Omega(\log^* n)$ on the number of rounds required to 3-color an n-cycle [CV86]. This is the first use of the celebrated round elimination technique. Given an algorithm $\mathcal{A}$ that uses r-many rounds for a problem, the technique transforms $\mathcal{A}$ into another algorithm $\mathcal{A}'$ that uses only $(r-1)$-rounds to perform a different (slightly simpler) task. Repeating this procedure r-times leads to an algorithm that performs a non-trivial task using 0-rounds, and gives a contradiction.

With unlimited bandwidth and arbitrary messages, observe that any node in the LOCAL model, in r rounds, knows its entire r-hop neighborhood and no other information. Hence, an r-round algorithm is assumed to have access to the entire r-hop neighborhood while

[9]This is highly problem dependent. For instance, in vertex coloring, each vertex must output its own color. In subgraph detection, if there is no subgraph, then all players must say NO, and otherwise at least one player must say YES. Typically, the format of output is such that the players do not have to communicate the answer through the edges, and it is sufficient for one of the players to find the answer.

proving lower bounds (see Figure 2.10).[10] This technique has led to many interesting lower bounds including: maximal matching and maximal independent sets [KS25, BBH+21], ruling sets [BBO20], vertex coloring [BBKO22a, BBK+25] and sinkless orientation [BKK+23].



(a) The r-hop neighborhood surrounding vertex u. This is what any r-round protocol has access to.

(b) The r-round protocol is transformed into one that only looks at $(r-1)$-hop neighborhood to solve a different problem.

Figure 2.10: Round Elimination to prove LOCAL lower bounds.

Another notable way of proving LOCAL lower bounds is the indistinguishability technique. Here, a family of graphs is constructed so that the r-hop neighborhoods of many vertices look identical, but a correct algorithm forces them to output different answers, which would be impossible. Perhaps the most prominent LOCAL lower bound is the work by [KMW16] that gives lower bounds for minimum vertex cover and maximum matching among other problems. More results that employ the indistinguishability technique are [Nao91, BFH+16].

Now, we move to the CONGEST model where in addition to locality constraints, there is also a bandwidth limitation of $O(\log n)$. The LOCAL model is significantly more powerful than the CONGEST model, and hence lower bounds here automatically apply to the CONGEST model as well. However, we do not know of round elimination or the indistinguishability technique yielding stronger lower bounds in CONGEST.

2.3.2 Information Bottleneck

The main technique to prove lower bounds in CONGEST is via an information bottleneck. If a vertex with degree d needs to output a solution that is $\Omega(n^c)$ in length for some constant c, a lower bound of $\Omega(n^c/(d \log n))$ on the number of rounds follows directly in the CONGEST model, because each vertex only receives a total of $O(d \log n)$ bits as messages in each round. An example of this technique is a lower bound of $\Omega(n/\log n)$ on the number of rounds required

[10] The article [Suo20] is an excellent introduction to round elminiation in the LOCAL model.

to approximate weighted all pairs shortest path (APSP) up to any poly(n) approximation by [Nan14].

In APSP, at the end of all the rounds, we require any vertex $v \in V$ to know the shortest path distance of v from every other node $v' \in V$. [Nan14] shows that if v has only one edge to the rest of the graph, it has to potentially learn n bits about the entire graph through one edge, leading to a lower bound of $\Omega(n/\log n)$ rounds. See Figure 2.11 for an illustration.

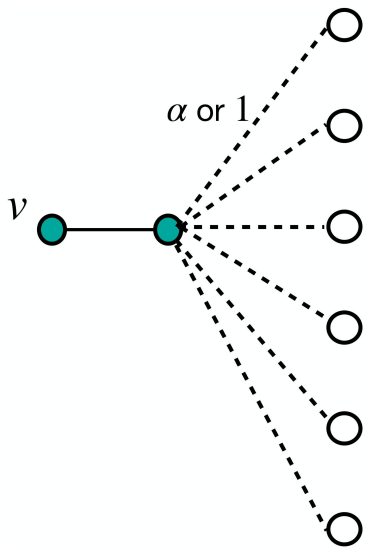


Figure 2.11: An illustration of the hard instance of APSP in CONGEST. Each of the dashed edges can have weight as α or 1. The vertex v must learn all these weights using its one incident edge.

Further examples of this technique are the work of [IG17] for triangle listing, [FGKO18] for more general clique listing. Observe that the output size for this problems is extremely large, and the information bottleneck is over one vertex having to receive a lot of information.

It is not necessary to only focus on the information one vertex receives, because we can isolate a small cut and prove that many messages have to be sent across the cut. Choosing cuts of small size leads to strong lower bounds, even for problems where the output size is small. Formally, these lower bounds are proved via reductions from communication complexity, starting from [PR00]. The hard problem used is almost always set disjointness [SHK+12, FHW12].

The input graph $G = (V, E)$ is carefully split between two *induced* subgraphs G_A and G_B with only *few* edges between G_A and G_B. They then show that the *hard* two-party communication problem can be embedded in the edges of G_A and G_B, while keeping the few edges between G_A and G_B fixed and input-independent. Assuming one needs t bits of communication for solving the two-party problem and there are only k edges between G_A and G_B, this construction immediately implies an $\Omega(t/(k \log n))$ round lower bound for the original problem; see Figure 2.12.

This technique has had a lot of success in proving lower bounds in CONGEST including but not limited to approximation of diameter [ACK16, FHW12], minimum spanning tree

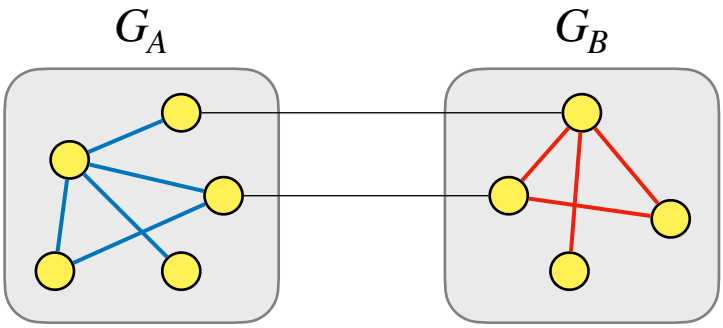


Figure 2.12: An illustration of using a small cut to prove lower bounds in CONGEST. There are only 2 edges between G_A and G_B, so a lower bound of $\Omega(t/\log n)$ will be proved. Here, t is a lower bound on the communication complexity of the underlying hard problem.

[KP95, SHK+12], minimum vertex cover [CHKP17, ACKP21], subgraph detection [DKO14, CK18, Cen22] and minimum dominating set [BCD+19].

Reductions from set disjointness have led to *optimal* lower bounds for many problems such as global min-cut [DEMN21, SHK+12], approximate minimum spanning tree [GKP98, PR00, SHK+12], 4-clique detection [CK18, CCGL21], maximum independent sets [CHKP17] and all pairs shortest path [LP15, Nan14, BN19], but the technique suffers from similar drawbacks as in the graph streaming model. Namely, algorithmically, the two-party communication model is much stronger than the CONGEST model where players only have local access, and hence some problems are provably easy in two-party communication.

2.3.3 Further Techniques

For many problems, efficient two-party communication protocols have been found, which preclude using a reduction from two-party communication to prove strong CONGEST lower bounds. For instance, state-of-the-art algorithm for detecting k-cliques uses $\tilde{O}(n^{1-2/k})$ [CCGL21] rounds, and the lower bound is $\tilde{\Omega}(\sqrt{n})$ rounds for constant k and this lower bound is tight in two-party communication [CK18]. Another example is that of maximum independent sets where there is a protocol that gets a $(1/2)$ approximation ratio in two-party communication, whereas this problem could be hard in the CONGEST model [BCD+19, EGK20].[11]

Multiparty Communication Complexity. A natural way to overcome the issue of two parties is to use reductions from multiparty communication complexity. This is closer to the CONGEST model, where the entire graph is divided among n players. Multiparty set disjointness is a natural candidate here. In this problem, there are k players, each with a set $S_1, S_2, \ldots, S_k \subseteq \mathcal{U}$. In one case, there is some element $x \in \mathcal{U}$ which is present in all the sets, and in the other case, all the sets are pairwise disjoint. It is known that this problem requires $\Omega(|\mathcal{U}|/(k \log k))$ communication [CKS03].

The work of [EGK20] uses a reduction from set disjointness over $t = O(1/\epsilon)$ players to prove that maximum independent set cannot be approximated to a $(1/2 + \epsilon)$ factor in $O(n/\log^3 n)$ rounds, and to a $(3/4 + \epsilon)$-factor in $O(n^2/\log^3 n)$ rounds, and these results

[11] Refer to Section 5 of [BCD+19] for a list of problems and the limitations of proving lower bounds via two-party communication.

cannot be proved using two-party communication complexity. Their reduction uses the tools developed in [BCD+19], which proves that an approximation ratio better than $(7/8 + \epsilon)$ requires $\tilde{\Omega}(n^2)$ rounds. [BCD+19] uses error correcting codes and probabilistic tools to create gaps in the size of the maximum independent set in the input instances, to extend the exact computation lower bound of $\tilde{\Omega}(n^2)$ to approximate maximum independent set also.

Fooling Views. The problem that perhaps exemplifies the issue of two-party communication is that of triangle detection. However one divides the three vertices of a triangle, at least one of Alice or Bob will have two vertices. [ACKL20] states that triangle detection is:

> *one of the best illustrations of the lack and necessity of new techniques for proving lower bounds in distributed computing*

They introduce a new technique called fooling views which combines the indistinguishability tool from LOCAL lower bounds and the information bottleneck in CONGEST lower bounds. For a simple hard instance of tripartite graphs composed of 6-cycles and triangles, their bound shows that if the bandwidth is 1, then the number of rounds required is $\Omega(\log^* n)$.[12] In the standard indistinguishability arguments, any r-round algorithm is assumed to have access to the entire r-hop neighborhood, and applying this technique directly for triangle detection does not give any meaningful multiround lower bounds. [ACKL20] overcomes this issue by exploiting the bandwidth limitations, and proving that a large subset of nodes receive the *same messages* irrespective of whether they are a part of a 6-cycle or a triangle. Such a vertex is called a fooled vertex, and the number of fooled vertices decreases exponentially in each successive round leading to a lower bound of $\Omega(\log^* n)$.

The assumption of many vertices receiving the same messages is infeasible in the standard CONGEST model because there may be $2^{O(\log n)} = n^{O(1)}$ different messages possible over each edge. Follow-up work by [FGKO18] uses the same technique, without the assumption that all messages have to be the same. They look at the entire transcript of the protocol[13] and show that a small number of vertices will still not know whether they are a part of a 6-cycle or two different triangles with a local view of the transcript. Their lower bound also holds only in the restricted model where the bandwidth is 1-bit and as far as we know this technique has not been adapted to the CONGEST model. This concludes our overview of the prior techniques used to prove lower bounds in distributed computing for large graphs.

[12] Their result does not lead to any lower bounds in the standard CONGEST model, and the main contribution of the paper is the technique of fooling views.

[13] Transcript contains all the messages across all the rounds sent by the protocol, with the identifiers of the edges through which the messages are sent.

3 A Brief Technical Overview

A central theme of our results is the construction of round-specific hard instances, in contrast to classical reductions that are inherently insensitive to the number of rounds. We see that these techniques can give optimal or near-optimal results in many contexts, and non-trivial tradeoffs between rounds and communication for a variety of problems.

We describe our techniques with a focus on three parts: (*i*) the prior one round lower bounds; (*ii*) the combinatorial components in our constructions; and (*iii*) the information-theoretic analysis of our constructions. In this chapter, we only provide a high level overview, and more specific tools unique to the results can be found in the respective chapters. We start with a primer on the round elimination technique of [MNSW95].

3.1 On Round Elimination

The one technical tool in common between Chapters 5 to 7 is the round elimination technique of [MNSW95] that was used to prove round-sensitive communication lower bounds. We give a brief summary of this technique, and how it can be adapted to prove our results. This technique was used by [ANRW15] to prove a lower bound for the communication complexity of bipartite maximal matching, where only one partition of the vertices can communicate. Another use of this technique is the work of [AKZ22], giving round-communication tradeoffs for maximal independent sets. Note that this technique is different from the round elimination of LOCAL lower bounds that was discussed in Section 2.3.1.

The 'Round Elimination Lemma' of [MNSW95] is one way to prove communication complexity lower bounds when the number of rounds is limited. Informally, the lemma has the following structure: an r-round hard instance of the problem is created by picking multiple $(r-1)$-round hard instances of the same problem $I_1, \ldots, I_k$, and giving them to the players. The players also receive additional inputs that "point" to one of these instances $I^* \in \{I_1, \ldots, I_k\}$ as being the **special** one in a way that, to obtain the final answer, they need to solve I^*.

The pointer is hidden to the players at first (e.g., is only given to the player not speaking first) and so they cannot reveal enough information about I^* in their first round; thus, they now need to solve I^* which is a hard $(r-1)$-round instance in $(r-1)$ rounds, with "almost no help" from the first round, implying the lower bound. At a high level, the hardness of the instances constructed using the round elimination lemma has an Index flavor. The bits of the Index problem are replaced with hard instances for $(r-1)$-rounds, and this makes the problem harder for one more round. See Figure 3.1 for an illustration.

Some important observations about the round elimination lemma are in order. The k instances given to the players are *independent* of each other. If the answers or the instances are correlated in some way, even in the first round, it is conceivable that the players make a lot of headway in solving them together.

Let us say that the input size of the smaller instance is n_{r-1}, and solving it in $(r-1)$-

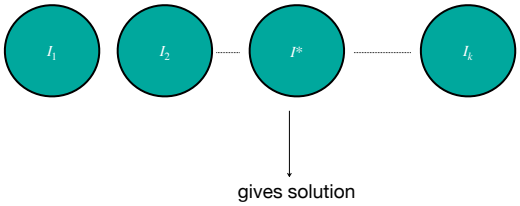


Figure 3.1: The round elimination framework of [MNSW95]. The hard instance is made of k smaller hard instances, where solving I^* gives the solution to the full instance. The identity of I^* is not revealed until the first round messages are sent.

rounds requires $s_{r-1}(n_{r-1})$ communication. The best lower bound we can hope to prove, even for r rounds is the same $s_{r-1}(n_{r-1})$, as solving I^* can be done in $(r-1)$-rounds in $s_{r-1}(n_{r-1})$ communication. The input size of the r-round instance, however, grows by a factor k. Thus, we can only prove weaker bounds with each successive round (which is to be expected; we are adding computational power with each round!), and mitigating this decay boils down to having small values of k.

What happens if k is extremely small, however? The players can use the first round to gain some information about *all* the instances, with only $\Omega(k)$ bits of communication. This leads to a balancing act:

1. Large values of k lead to larger instances for each subsequent round, and weaker bounds relative to the size of the instances.

2. Small values of k may mean that the first round actually reveals non-trivial information about I^* that aids the players in solving the problem.[1]

	r-rounds	$(r-1)$-rounds
Input size	$k \cdot n_{r-1}$	n_{r-1}
Lower bound	$\min\{k, s_{r-1}(n_{r-1})\}$	$s_{r-1}(n_{r-1})$

Table 3.1: Table with the instance and lower bound sizes in Round Elimination Lemma.

The best lower bound we can hope to prove, hence becomes a minimum of $\Omega(k)$ and $\Omega(s_{r-1}(n_{r-1}))$. The lemma shows that this lower bound indeed holds in many settings when the underlying k instances are independent. We illustrate round elimination with a simple example by proving an $\Omega(n)$ lower bound for the Index problem.

3.1.1 Using Round Elimination for Index

We informally defined the Index problem in Section 2.2.2. Now, we give a formal definition and use round elimination to prove a communication lower bound for Index.[2] The key idea is to simulate the first-round message M using public randomness which eliminates one round.

[1] More formally, we want the players to only convey $o(1)$ information about I^*, which is only true if $o(k)$ bits are sent in the first round.

[2] There are multiple ways to prove a linear lower bound for Index, and our proof is folklore.

The joint distribution of the inputs and the message is approximately preserved even with this sampling.

Definition 3.1.1. *For any $n \geq 1$, in the* INDEX$_n$ *problem, there are two players Alice and Bob. Alice receives a string $x \in \{0,1\}^n$ and Bob receives $i \in [n]$. Alice sends a single message M to Bob, and after receiving the message from Alice, Bob has to output the value of x_i.*

In addition, Alice and Bob also have a source of public randomness that they can use.

The INDEX is a one round communication problem. Assume towards a contradiction that there exists a protocol π_1 where Alice sends messages of length at most $s \ll n$ with probability of success at least $2/3$.

Our proof for INDEX is by way of Yao's minimax principle [Yao77], which reduces our task to proving a lower bound for deterministic algorithms over a hard input distribution. The hard distribution we use is simple: $x \in \{0,1\}^n$ and $i \in [n]$ are chosen uniformly at random and independently. This distribution of the inputs induces a distribution over the messages of π_1 also. Let $\mathcal{D}$ be the joint distribution of (x, M, i) in protocol π_1.

We use the following 0-round problem, that is hard to solve trivially.

Definition 3.1.2. *The* ECHO *is defined as follows. There are two players Alice and Bob. They have a source of finite but arbitrary length random bits public to both Alice and Bob. They also have a source of private randomness for both of them.*

Alice has a bit $b \in \{0,1\}$ chosen uniformly at random, and Bob has to output the bit b without any communication from Alice.

It is easy to see that whatever bit Bob outputs, he is not going to succeed with probability more than $1/2$. However, if there is a protocol π_1 for INDEX$_n$ that uses $o(n)$ communication, we will see that we can eliminate one round from π_1 to get a protocol for ECHO that gets a probability of success larger than $1/2$. To get this protocol, we view the INDEX$_n$ as n different instances of ECHO, and Bob wants to solve one of them that $i \in [n]$ points to.

The first step in constructing a protocol for ECHO is an *embedding* argument: we want to use the input to ECHO to generate an input to INDEX$_n$ so that solving this instance gives an answer to ECHO also.

Embedding ECHO into INDEX$_n$:

(*i*) Alice and Bob use public randomness to sample a uniformly random index $i^\star \in [n]$. Bob fixes the value of i to be $i^\star$.

(*ii*) Alice sets $x_{i^\star} = b$, and uses private randomness to sample the rest of input x, conditioned on the value of $x_{i^\star}$ and $i^\star$.

It is easy to see that in the above process, Alice and Bob generate an instance of INDEX$_n$ such that finding $x_i = b$ solves ECHO. However, any protocol for INDEX$_n$ uses one message from Alice to Bob that ECHO does not allow. We see that whatever protocol INDEX$_n$ uses, we can sample the message from Alice to Bob using public randomness.

Protocol π_0 for ECHO using π_1:

(*i*) Alice and Bob use public randomness to sample a uniformly random index $i^\star \in [n]$. Bob fixes the value of i to be $i^\star$.

(*ii*) They also use public randomness to sample a message M from distribution $\mathcal{D}$.

(*iii*) Alice sets $x_{i^\star} = b$, and uses private randomness to sample the rest of input x, conditioned on the value of $M, x_{i^\star}$ and $i^\star$.

(*iv*) Bob uses the message M, index $i = i^\star$ and his own source of private randomness to output the answer of π_1.

We have sampled the message M from the distribution $\mathcal{D}$, without Alice actually sending any message to Bob, thus *eliminating* one round. The players in protocol π_0 do not perform any communication at all.

In an ideal world, the message we sample would fully agree with the input of Alice, the string x. In our scenario, Alice sets the value of $x_{i^\star}$ independently of the sampled message M. Thus, the distribution of (x, M, i) in our protocol *is not the same as* the distribution in π_1. Let us look at a few examples of what π_1 can do.

π_1 **sends all the bits:** If protocol π_1 sends all the bits of x to Bob, Alice cannot sample (x, M, i) consistently always. The message M fixes value of x_i, but the input bit b of Alice agrees with x_i only with probability $(1/2)$. Thus, with probability $(1/2)$, any (x, M, i) sampled in π_0 will not be valid.

π_1 **sends a few random bits:** Let π_1 send k bits of x chosen randomly along with their indices. As $i^\star$ is chosen uniformly at random, whenever $i^\star$ falls into one of these k bits, the (x, M, i) we sample is valid with probability $(1/2)$. If $i^\star$ does not fall into one of these k bits, we can sample (x, M, i) consistent with π_1.

π_1 **sends a sketch:** Let $A \subset \{0,1\}^{s \times n}$ be a randomly chosen matrix, and Alice sends $M = A \cdot x$ in π_1. When simulating this message in π_0, matrix A and sketch M are fixed. If the bit at position i is fixed by A, M[3], then bit b is faithful with probability $(1/2)$. Otherwise x_i is uniformly random. As the rank of A is at most s, only at most s co-ordinates of x can be fixed by A, M.

In all these examples, if the message length is small, the simulation is closer to the one we desire, i.e., closer to being valid in protocol π_1. We can formalize this intuition using information-theoretic arguments. We prove that the joint distribution of (x, M, i) in protocol π_0 (say $\mathcal{D}'$) is close to the distribution in protocol π_1, using which we can prove that the answer from Bob will be close to correct as well.[4] Formally, we prove this claim:

[3]In other words, $e_i \in \{0,1\}^n$, which is the vector with 1 at position i and 0 everywhere else, belongs to the row span of A

[4]Through this proof sketch, we use the term total variation distance, and some properties about this

Claim 3.1.3. *The total variation distance between the joint distribution of (x, M, i) in protocol π_1 under the uniform hard distribution of (x, i) and in protocol π_0 is small.*

$$\|\mathcal{D}(\mathsf{x}, \mathsf{M}, \mathsf{i}) - \mathcal{D}'(\mathsf{x}, \mathsf{M}, \mathsf{i})\|_{\text{tvd}} \leq \frac{1}{2} \cdot \sqrt{\frac{s}{n}}.$$

The full proof details are not relevant to the rest of this chapter, but it crucially uses that π_1 only sends s bits of communication. We postpone the proof to Section A.1. Equipped with Claim 3.1.3, proving that π_0 has high probability of success is easy.

We use Fact 4.3.5 with the event $\mathcal{E}$ as the output of Bob being correct.

$$\begin{aligned}
\Pr[\pi_0 \text{ succeeds }] &\geq \Pr[\pi_1 \text{ succeeds }] - \|\mathcal{D}(\mathsf{x}, \mathsf{M}, \mathsf{i}) - \mathcal{D}'(\mathsf{x}, \mathsf{M}, \mathsf{i})\|_{\text{tvd}} \\
&\geq \frac{2}{3} - \frac{1}{2} \cdot \sqrt{\frac{s}{n}} \\
&> \frac{1}{2}. && (\text{for } s \ll n)
\end{aligned}$$

This shows that π_0 has $> 1/2$ probability of success, which is a contradiction, because π_0 had no communication at all. Bob has no information about the bit b at all, and any value from Bob only has a probability $1/2$ of being correct.

Now, that we have a proof sketch, let us focus on the most important (and non-trivial) step of protocol π_0. Alice and Bob collectively sampled the distribution of M, in such a way that not conditioning on x_i did not change the distribution too much, and this is formally stated in Claim 3.1.3. This is the key step in many applications of the round elimination lemma: showing that the first round messages did not reveal much about the hidden instance, and providing a way to sample these messages. In our example with the Index problem, this step was possible because the distribution of M did not reveal much information about x_i when Alice did not know the value of i.

This example illustrates the central principle of round elimination:

> *If the first-round message reveals little information about the hidden instance, then we may be able to simulate it and remove it altogether.*

Next, we go into the details of how classical round elimination is unsuitable, and how we adapt it to different settings to yield optimal and near optimal lower bounds. We do not focus on the specific sampling process in the results, and instead we present the high level overview of the constructions and the parameters that lead to the final lower bound.

3.2 Streaming Maximal Independent Sets

Recall that Theorem 1 was a lower bound of $\Omega(\log \log n)$ on the number of passes required to find a maximal independent set in semi-streaming. We establish Theorem 1 by proving

distance that may be unfamiliar. Roughly speaking, it quantifies how far two distributions are from each other. The formal definitions are presented in Chapter 4.

a **round vs communication tradeoff** for the (multi-party) communication complexity of MIS.

We start from the one round hard instance of [ACK19b,CDK19], which prove that finding an MIS in one pass requires $\Omega(n^2)$ space. We discuss this instance directly with the view of two-party communication complexity.

- Alice receives *two identical copies* of a bipartite graph G where each pair becomes an edge with probability $1/2$. There are totally $\Theta(n^2)$ edges.
- Bob receives the following graph on the same set of vertices: pick a pair $\{u, v\}$ of vertices on the two sides of the bipartition of G and connect all vertices that are *not* a copy of these vertices to each other.

We want to compute an MIS of the union of the graphs of Alice and Bob. Computing the MIS of such a graph requires the players to figure out if (u, v) is an edge in G or not. If some vertex $w \neq u, v$ in the first copy is picked in the MIS, then all other vertices barring u, v are excluded as they are neighbors of w. Thus, any valid MIS contains both u, v if (u, v) is not an edge, and only one of u, v if (u, v) is an edge. Similarly, if no such vertex w is picked, the MIS reveals whether (u, v) is an edge in the first copy. See Figure 3.2 for an illustration.

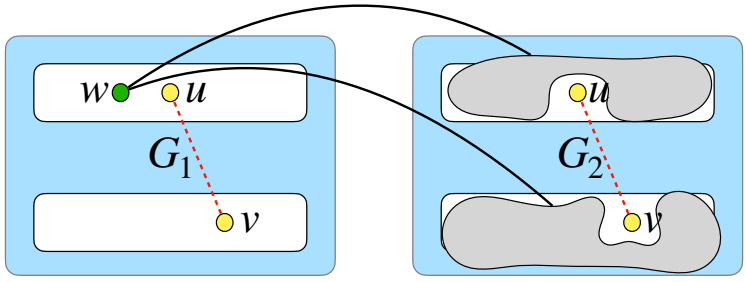


Figure 3.2: Alice's input is G_1 and G_2 which are copies of the same random bipartite graph G. Bob's input connects every vertex other than copies of u and v to each other. In the image, this is illustrated as w being connected to all vertices in the grey cloud and vice-versa. If we pick a vertex w in the left to join the MIS, all its neighbors ar forced out. Only vertices u and v in the right remain to join the MIS (the figure only shows some of w's edges).

To extend the lower bound to two rounds and more, we employ the round elimination technique with one tweak: instead of the players needing to solve one smaller instance I^*, we force them to solve *many* small hard $(r-1)$-round instances to solve one hard r-round instance. This change is found to be necessary to get the optimal lower bound for MIS, and we explain why briefly.

Limitations of Round Elimination

Suppose we want to extend the round elimination approach beyond a single round, by having just one special smaller instance. Let us try to prove a $n^{1+\varepsilon}$ lower bound for two-round protocols with $\varepsilon > 0$.

In the one-round lower bound, the MIS was forced to find a particular edge. We can use the same approach to force the MIS to focus on a special set of vertices. Again, we have two identical copies of the same **base graph** for Alice and give Bob an input that connects these copies. Instead of a random graph however, we need Alice's base graph to be a collection of *independent* copies of the 1-round hard instances. Here, 'independent' copies not only mean in a probabilistic sense, but also in a graph-theoretic sense: the edges of one instance should not appear between vertices of another as otherwise the instances become "corrupted", namely, they are no longer necessarily hard 1-round instances. (See Figure 3.3.)

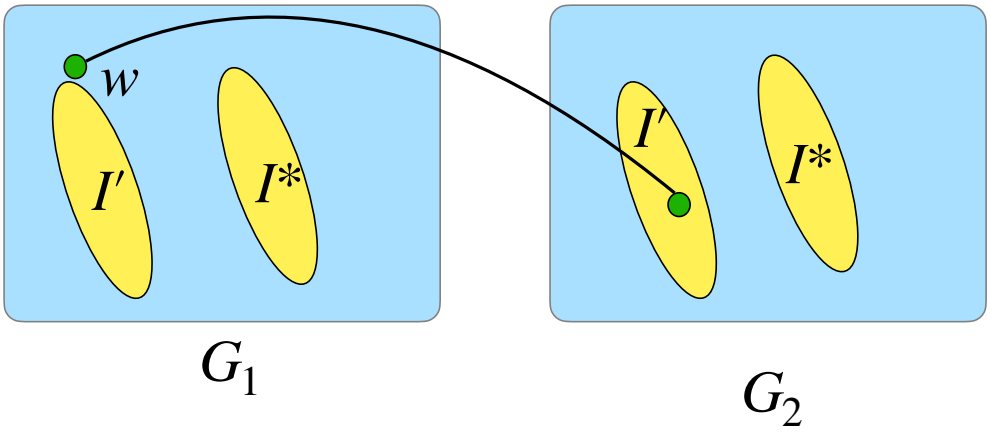


Figure 3.3: The special instances should not corrupt each other. Bob's edges can be used to isolate a special instance I^*

We also need these instances to be quite *large*, so that a 1-round lower bound for them incurs a significant communication cost in terms of the *entire* graph size. A bit more specifically, to make this strategy work for a two-round $n^{1+\varepsilon}$ communication lower bound, for any constant $\varepsilon > 0$, we need Alice's base graph to have the following properties:

1. It must contain $(n^{1+\varepsilon})$ 1-round instances so Alice cannot reveal too much about a random instance;
2. Each 1-round instance needs to have $(n^{(1+\varepsilon)/2})$ vertices so that the quadratic 1-round lower bound implies that the second round of the protocol needs $(n^{(1+\varepsilon)/2})^2 = n^{1+\varepsilon}$ communication;
3. The instances cannot have edges between each others' vertices so as not to "corrupt" one another.

Recall the parameters of round elimination from Section 3.1. These properties are essentially saying:

$$\min\{k, s_1\} = n^{1+\varepsilon},$$

while maintaining independence among the smaller instances.

Given that one-round instances have quadratic number of edges, satisfying all these constraints means having a base graph on n vertices with $n^{2+2\varepsilon}$ edges or even pairs of

vertices![5]

This highlights an inherent difficulty (basically, impossibility) of implementing this natural type of round elimination argument for proving communication lower bounds on graphs. This is why in our approach, we force the (r)-round instance to solve *many* smaller $(r-1)$-round instances. We see why this leads super linear lower bounds when discussing the parameters later in this section.

Combinatorial Component

In order to translate the round elimination lemma into graph instances, we need to embed the instances in graphs in a way that they do not interact with each other, as the instance of [AR20] does for two passes. We briefly discussed this property of [AR20] in Section 2.2.3. These independent subgraphs in [AR20], and later in [Ass22, CKP+21b, KN24] are bipartite. Using standard RS graphs, it is not clear how to embed more complex smaller hard instances.

In the combinatorial component of our work on MIS in streaming, we give a way to embed non-bipartite graphs in a way that they do not interfere with each other by constructing a family of large graphs which have many disjoint unique path collections. Roughly, each edge of an induced matching in an RS graph is replaced with a long path instead (they can be viewed as "stretching" the edges of an induced matching into paths that do not interfere with each other). The construction borrows ideas from [Beh46, RS78], which were used to construct bipartite RS graphs. See Figure 3.4 for an illustration.

More specifically, for parameters $p, q, k \in \mathbb{N}$, in our Disjoint Unique Path (DUP) graph, we have:

- An induced unique path collection (UPC) of size p is a collection of p-many paths $P_1, \ldots, P_p$ of length k each, such that $P_i = (u_i, \ldots, v_i)$ is the unique shortest path between u_i and v_i. Moreover, the subgraph induced by the vertices in all p-paths do not contain any edges that are not in any of these p-paths.
- A (p, q, k)-DUP graph is a graph whose edges can be decomposed into q many induced path collections of size p and length k.

Similar to using RS graphs in [AR20, CKP+21b], each UPC can now embed independent smaller hard instances, but the embedded instances are not necessarily bipartite.

Information-theoretic Component

The information-theoretic component of Theorem 1 is mainly proving the new round elimination lemma. More specifically, if the players have to solve q-many instances in the r-round instance, they will correspondingly need more communication. [6]

In our round elimination lemma, to construct the hard r-round instance, we use $q \cdot p$ many $(r-1)$-round instances, $I_{i,j}$ for $i \in [q]$, $j \in [p]$. The players are required to solve *all* $I_{i^\star,j}$ for $j \in [p]$ for a random $i^\star \in [q]$ that is revealed at the end of the first round.

[5] Making 1-round instances sparser or similar modifications does not work either, as that forces us to pick even larger 1-round instances in Line 2, leading to the same exact contradiction.

[6] This lemma is isolated from any combinatorial constructions about how the instances are combined or that the instances are graphs, and is purely a communication complexity lower bound.

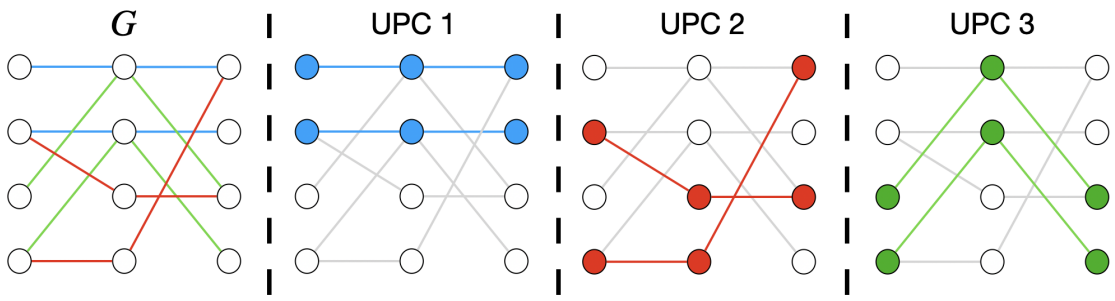


Figure 3.4: An example of DUP graph with $q = 3, p = 2$ and $k = 2$. In each UPC, there is no path of length 2 from a starting vertex to an end vertex unless the path belongs to the UPC itself.

We prove the following lower bound for r-rounds:

	r-rounds	$(r-1)$-rounds
Input size	$p \cdot q \cdot n_{r-1}$	n_{r-1}
Lower bound	$\min\{p \cdot q, p \cdot s_{r-1}(n_{r-1})\}$	$s_{r-1}(n_{r-1})$

Table 3.2: Table with the sizes of the instance and lower bound on the message lengths.

Now, even though the input size grows by a $p \cdot q$ factor, we can prove a stronger lower bound proportional to p. Our proof uses information complexity direct sum [CSWY01] and message compression arguments [HJMR07, BG14]. The proof details are postponed to Chapter 5.

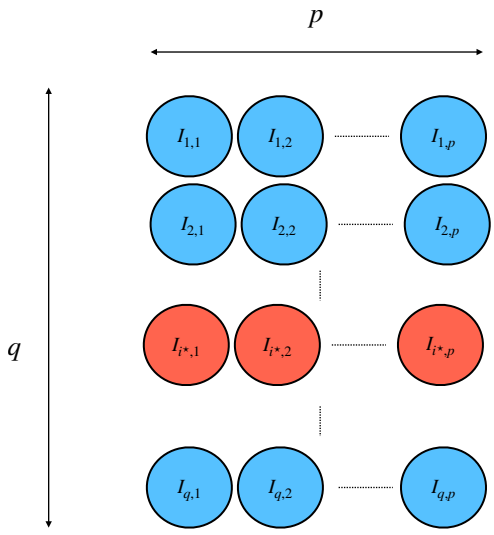
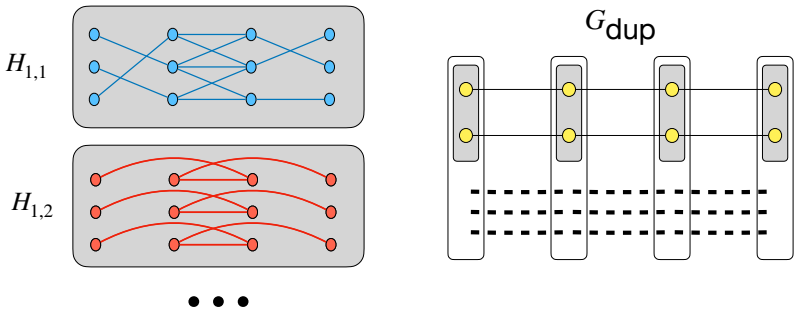


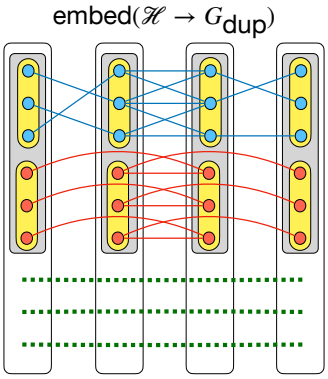
Figure 3.5: Our round elimination framework. To find the answer to the instances, the players have to solve all p smaller instances in row $i^\star$.

Putting Things Together

We use our combinatorial construction to embed $p \cdot q$ many smaller $(r-1)$-round hard instances into a single graph, in a way that in a UPC, the p-many instances do not interfere with each other. Naively embedding them would cause the number of vertices to grow by a $p \cdot q$ factor, but we avoid this growth using our extremal graph construction. See Figure 3.6.



(a) Two smaller hard instances H_1 and H_2 on the left, and a DUP-graph on the right with $p = 2$, $k = 3$ and unspecified q.



(b) The resulting layered graph as the embedding of H_1, H_2 into G_{dup} – the drawing only shows the effect of the embedding on one UPC of G_{dup} and the remainder of the graph is unspecified.

Figure 3.6: An illustration of embedding smaller instances into DUP graphs. The figure only shows how one UPC is replaced, in the final instance, *all* UPCs in G_{dup} are replaced with smaller instances.

Each path in all q-UPCs is replaced with a smaller instance. Finally, one UPC among the q-many ones is picked so that any algorithm that finds an MIS will have to solve *all* p smaller instances in the UPC. This corresponds to picking one row out of q rows and solving all the instances in row $i^\star$ in our information-theoretic component.

To conclude our discussion on Theorem 1, we briefly discuss the parameters, and why our bound is $\Omega(\log \log n)$. Let n_2 be the number of vertices in the hard instance for the 2-round lower bound, and n_1 be the number of vertices in the smaller 1-round hard instance.

Our parameters grow as:

$$s_1(n_1) = \Omega((n_1)^2) \qquad p, q \approx n_2/n_1 \qquad n_2 \approx (n_1)^3 \qquad s_2(n_2) \approx (n_2/n_1) \cdot (n_1)^2 \approx n_2^{4/3}.$$

The boost we get in the $s_2(n_2)$ term with $p = (n_2/n_1)$ is the reason why we are able to get meaningful two-round lower bounds. We are able to extend these parameters for any round $r \geq 1$. This gives a lower bound of $\Omega(n^{1+1/2^{\Theta(r)}})$ on the message length for any r-round algorithm for n-vertex graphs. For $r = o(\log \log n)$, this lower bound exceeds $\omega(n \text{poly} \log n)$ proving that $\Omega(\log \log n)$ passes are required by semi-streaming algorithms to find an MIS.

3.3 Streaming Matchings

For streaming matchings, Theorem 2 and Theorem 3 prove a lower bound of $\Omega(\log(1/\varepsilon))$ for finding $(1-\varepsilon)$-approximations conditionally on the density of RS graphs and unconditionally respectively. We also have Theorem 4 which proves a stronger $\Omega(\log \log n)$-pass lower bound to find any constant approximation in dynamic streams.

Theorem 2 is subsumed by Theorem 3 quantitatively under the current best lower β_{RS}.[7] Moreover, the techniques behind Theorem 2 are mostly disjoint from the rest of this thesis. Hence, we focus on the techniques behind Theorem 3 and Theorem 4.

3.3.1 Insertion-only Streams

Theorem 3 uses extremely similar techniques as Theorem 1, but with a focus on the approximation ratio of matchings instead. Our starting point is the work of [GKK12] which proves that single pass semi-streaming algorithms cannot obtain an approximation ratio better than 2/3 for maximum matchings, and its subsequent extensions in [KN24], who showed that two-pass algorithms cannot obtain better than a 8/9-approximate matching.

The construction of [GKK12] uses the RS graph $G = (L \cup R, E)$ in Proposition 4.2.2. There are q-many induced matchings of size p each, and the edges of these matchings exist independently of each other with probability 1/2. To store whether all the edges exist or not, any algorithm will require $\Omega(p \cdot q)$ space.

After the edges of the RS graph are sent in the stream, one induced matching (say $M^\star$ between $L^\star$ and $R^\star$) is chosen at random. All vertices in the RS graph except $L^\star$ and $R^\star$ are connected to new vertices $(A \cup B)$ via an auxiliary matching. The graph still has a perfect matching by taking the auxiliary matching and $M^\star$, of size $3n/2$. However, if edges of $M^\star$ are excluded, the set $(L \setminus L^\star) \cup (R \setminus R^\star)$ forms a vertex cover of size n. Therefore, if an approximation ratio larger than (2/3) is to be achieved, edges of $M^\star$ *must* be included. The algorithm does not know the identity of $M^\star$, and ends up storing an ε-fraction of the edges of $M^\star$ for any $(2/3 + \varepsilon)$-approximation. This leads to a lower bound of:

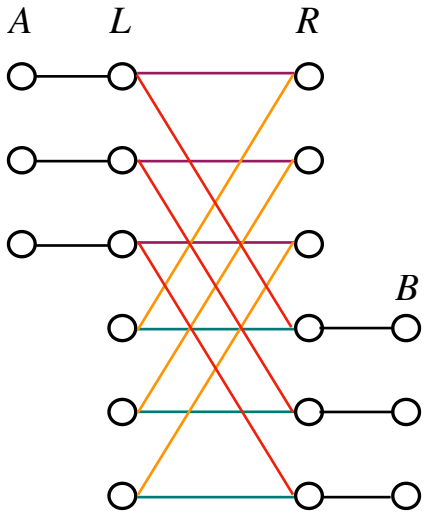
$$\Omega(pq) = n^{1+\Omega(1/\log \log n)}$$

for any constant ε. This shows that semi-streaming algorithms cannot obtain an approximation ratio better than (2/3).[8] See Figure 3.7 for an illustration.

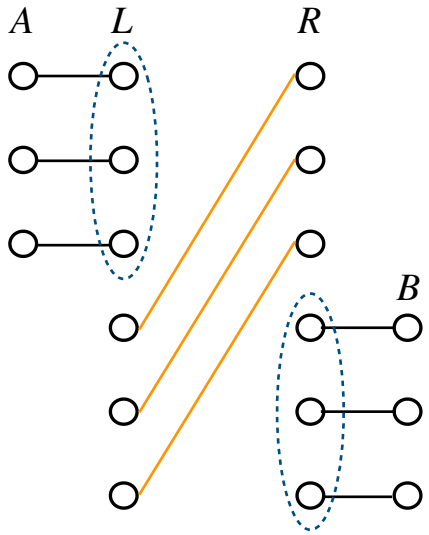
The work of [KN24] directly extends [GKK12] to two passes, by replacing each edge of the RS graph with a smaller hard instance. More specifically, the one round lower bound

[7] Although Theorem 2 holds for approximate estimation of the matching size also, Theorem 3 holds only when the algorithm is required to output the matching edges.

[8] In subsequent work [Kap13,Kap21], this result has been improved, and we know that any approximation ratio better than 0.59 is not possible in a single pass in semi-streaming.



(a) Given an RS graph on vertices $L \cup R$ with 4 matchings of size 3 each, vertices $A \cup B$ are added, which connect to all vertices except the end points of the special matching.



(b) The orange edges of the RS graph are present with probability (1/2) independently. The vertices in the blue circle are a vertex cover of the other edges in the graph.

Figure 3.7: One round hard instance for (2/3) approximate matching in [GKK12]. Any $(2/3+\varepsilon)$-approximate matching must find at least an ε-fraction of the yellow edges.

of [GKK12] relies on the algorithm not knowing the identity of the special matching before the auxiliary edges are revealed. What if each edge of the special matching is now replaced by an independent hard instance for one round? The inducedness of the special matching ensures that these smaller hard instances will not interfere with each other. This could potentially make it hard for two-pass algorithms: in the first pass, the special matching of the outer graph will be unknown, and in the second pass, all the special matchings of the smaller independent instances will be unknown. See Figure 3.8 for an illustration.

[KN24] implements this approach to prove that any algorithm with an approximation ratio better than $(1-(1/3)^2) = (8/9)$ requires more than semi-streaming space (see Figure 3.8). In their instance:

1. The auxiliary edges of the outer instance cover $(2/3)^{\text{th}}$ of the vertices, and the auxiliary edges of the smaller instances in the special matching cover another (2/3) fraction of the vertices. In total, these auxiliary edges cover:

$$\frac{2}{3} + \frac{1}{3} \cdot \frac{2}{3} = \frac{8}{9}$$

   fraction of the vertices. Thus, the lower bound holds when an approximation ratio larger than (8/9) is found.

2. Let $2n_1$ be the number of vertices in the inner RS graph hard instances. The outer graph is chosen with $b_2 \approx n_1^{\log\log n_1}$ vertices. The total number of vertices will be $b_2 \cdot n_1$. Any streaming algorithm has two choices:

(a) Use $\Omega(b_2^{1+1/\log\log b_2})$ space to gain information about the $\approx b_2$ many smaller special instances in the first pass;

(b) Or spend $\Omega(b_2 \cdot n_1^{1+1/\log\log n_1})$ space to solve a majority of the smaller instances in the second pass without help from the first pass.

This leads to a lower bound of

$$b_2 \cdot \min\{b_2^{1/\log\log b_2}, n_1^{1+1/\log\log n_1}\} = n_2^{1+\Omega(1/\log\log n_2)^2},$$ [9]

which is not possible in semi-streaming.

The techniques of [KN24] however, do not extend beyond two passes, since in their information theoretic component they rely on the two-way communication complexity tradeoffs of the Index problem [JRS03].[10] We extend this framework further, and show that any p-pass algorithm cannot get an approximation ratio better than $(1-1/3^p)$ in semi-streaming space, which in turn proves that any $(1-\varepsilon)$-approximation algorithm requires $\Omega(\log(1/\varepsilon))$-passes.

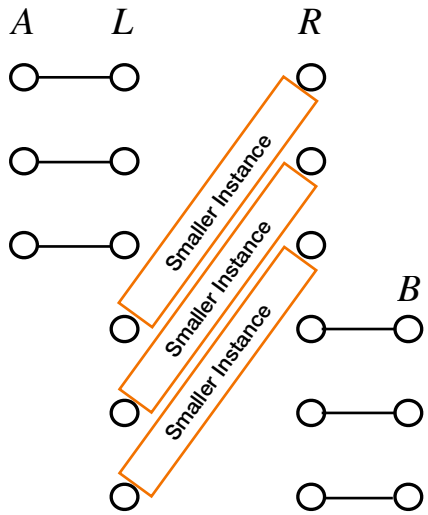


Figure 3.8: A hard instance constructed by [KN24], Any $(8/9+\varepsilon)$-approximate matching finds at least an ε-fraction of the edges of the special matchings in the smaller instances. The smaller instances look exactly as in Figure 3.7a.

The hard instances of [GKK12] and [KN24] are bipartite, so to embed them in the form of independent smaller subgraphs, we can directly use RS-graphs, unlike Theorem 1. Hence, for the combinatorial component, we do not need to construct any new family of graphs to do the embedding, and simply use RS graphs from Proposition 4.2.2.

The information-theoretic component of Theorem 1 is applicable here, and allows us to extend this lower bound. More concretely, we proved a new round elimination lemma,

[9] Refer to Section 6.1, proof of Lemma 6.1.10 for more detailed calculations.

[10] If Bob sends $O(\log n)$ bits to Alice, we know Alice needs to send one bit to solve the problem. [JRS03] proves the more general tradeoff that whenever Bob sends b bits, Alice needs to send $\Omega(n/2^b)$ bits for 2/3 probability of success.

where solving groups of smaller $(r-1)$-round hard instances proportionately takes more communication. Similarly, each induced matching can be viewed as a group of smaller instances, and any algorithm that finds a large matching must solve *many* of these smaller instances.

Let us move to the techniques behind Theorem 4.

3.3.2 Dynamic Streams

Theorem 4 also uses the tools from Theorem 1, but there are challenges in adapting them for dynamic streams and matchings. Moreover, we want to prove a much stronger lower bound than Theorem 3, in that *any* constant approximation takes $\Omega(\log\log n)$-passes.

Combinatorial Component

The construction of Theorem 3 does not give any meaningful results for dynamic streams, as finding constant approximations in the hard instance is easy. We rely instead on hard instances specific to dynamic streams, and adapt the tools of Theorem 1.

We begin with the hard instance of [DK20], which shows that any constant approximation of maximum matching in one pass requires $\Omega(n^2)$ space. They forgo complex constructions using RS graphs by crucially exploiting the edge *deletions* in the stream and construct much simpler elegant graphs.

Consider the following communication problem.

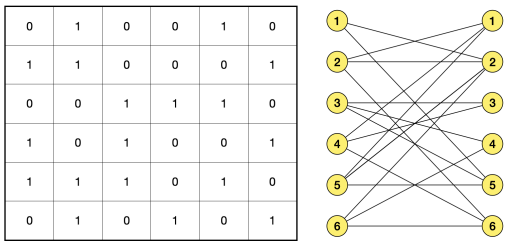
- Alice receives an $n \times n$ matrix A encoding the bipartite adjacency matrix of a *random* bipartite graph G between two sets of vertices L and R, each of size n. Each entry of A is independently chosen to be 0 or 1 with probability half.

  We think of the input to Alice as the insertions of edges E_{ins} at the beginning of the stream.

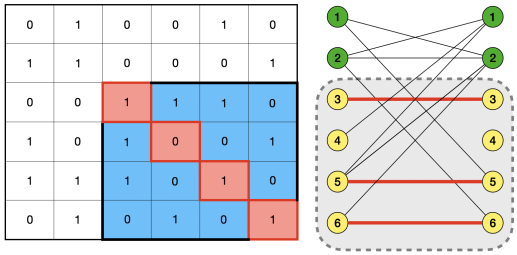
- The input to Bob is random permutations σ_r and σ_c of $[n]$ that reorder rows and columns of A, respectively. Bob is also given the following submatrix B of A: reorder the rows and columns of A according to σ_r and σ_c; then, for some previously-fixed $i^\star = n - o(n)$, consider the $(i^\star \times i^\star)$ submatrix of A at the bottom right corner of A (after the reordering). Provide this sub-matrix except for its *diagonal* entries as B, and let diagonal entries of B be 0's instead.

  We think of the input to Bob as the deletion of edges E_{del} in the second half of the stream (based on the bipartite adjacency matrix). Since B is a submatrix of A, all edges deleted by Bob are already inserted by Alice. The input graph G at the end of the stream is then the bipartite graph on bipartition L and R and edges $E_{\text{ins}} \setminus E_{\text{del}}$.

It is not hard to see that the graph G created this way has an induced matching M of size $n/2 - o(n)$ with high probability (inside the submatrix B of A with edges $A - B$, as each edge of the diagonal is present with probability $(1/2)$). Moreover, the graph $G \setminus M$ has a vertex cover of size $o(n)$ (by picking the $o(n)$ vertices not part of B). This implies that any $O(1)$-approximate matching algorithm on the graph $E_{\text{ins}} \setminus E_{\text{del}}$ needs to recover many edges of M. See Figure 3.9 for an illustration.



(a) Alice's input A is the bipartite adjacency matrix of the graph E_{ins}.



(b) Bob's input B is the sub-matrix with blue entries, with diagonal (red) entries replaced with 0. The graph here is $E_{\text{ins}} \setminus E_{\text{del}}$ with red edges being the induced matching M and green vertices being vertex cover of $G \setminus M$.

Figure 3.9: An illustration of the inputs A and B to Alice and Bob in the lower bound of [DK20]. Here, $n = 6$ and $i^\star = 4$ and both σ_r and σ_c are identity permutations. In general, the submatrix of B corresponds to a combinatorial rectangle in A, the induced matching is of size $n/2 - o(n)$ with high probability, and the vertex cover is of size $o(n)$.

In [DK20], it was shown that given only A and without the knowledge of σ_r and σ_c, majority of edges in E_{ins} have an equal chance of appearing as the edges of the induced matching M. Thus, from Alice's perspective, most edges in E_{ins} can become important in finding a large matching in $E_{\text{ins}} \setminus E_{\text{del}}$. Hence, Alice needs to communicate $\Omega(n^2)$ bits before Bob can output an $O(1)$-approximate matching in the graph. There is one caveat here: Bob already knows a big part of the matrix A, and [DK20] proves that even in this scenario, Alice needs to send $\Omega(n^2)$ length messages through a reduction to the augmented Index problem [CCKM13].

We are able to implement the round elimination framework of Theorem 1 again with this view: each position in the matrix is replaced by a smaller subinstance. More specifically, to construct an r-round hard instance, we replace each matrix by a smaller $(r-1)$-round hard instance, and any successful algorithm must solve a large fraction of these instances.

The hard instances consists of a stream that inserts many $(r-1)$-pass hard instances together, not necessarily with any induced subgraph collections[11], followed by deletions of edges in many of these instances so that effectively only one large induced collection of $(p-1)$-pass instances remain, corresponding to the edges that an algorithm needs to find in the one round lower bound of [DK20] (see Figure 3.10). We show that in the context of $O(1)$-approximate matchings, this is enough to force any algorithm for the r-round instance to also solve many $(r-1)$-round hard instances.

There is one crucial aspect of this combinatorial component that we do not cover in this overview, namely, how do the players ensure that the deletions happen across multiple passes in a consistent way? In one round, this was simple, because Bob had access to matrix B correlated with A. We use the techniques of [AKZ24] which shows that the Augmented Index problem can be extended to Augmented Tree Pointer Chasing to handle the shared

[11] This aspect is different from Theorem 1, as the many smaller instances need not come from any one row of the $p \cdot q$ many smaller instances.

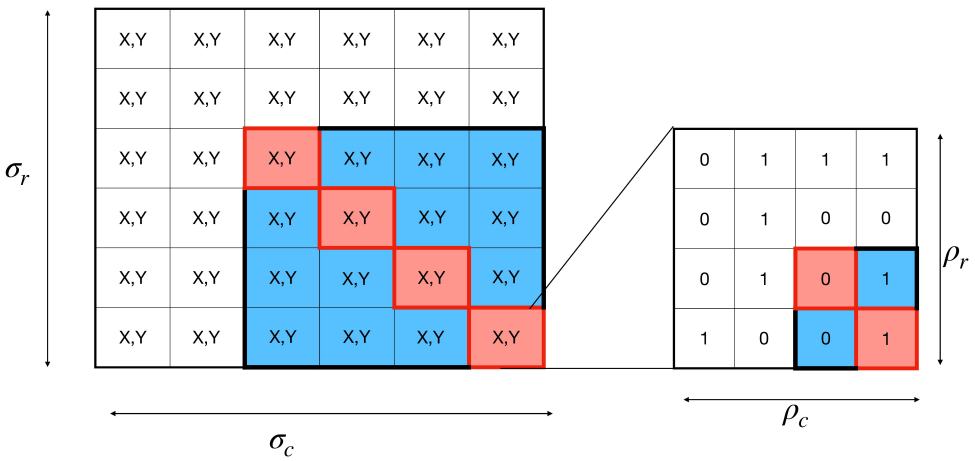


Figure 3.10: An illustration of the two round hard distribution of Theorem 4. (X, Y) denote instances sampled from the 1-round hard distribution. Each submatrix has its own independent permutation ρ_r, ρ_c. Alice and Bob have to solve the instances highlighted in red. The squares shaded in blue are zeroed out in both matrices.

input between the players. These details can be found in the proof in Chapter 6.

Information-theoretic Component

Addressing the previous challenge using edge deletions creates a significant hurdle in applying the information-theoretic arguments of Theorem 1. The new round elimination argument we established earlier crucially relies on the *independence* of the inputs of players in the corresponding communication game used to establish the streaming lower bound. In contrast, such an independence cannot hold for us given that we need to ensure the input of one player is only deleting edges already inserted by another player (otherwise, the stream may delete edges that have not been inserted). These deletions reveal the large induced smaller subgraphs, and hence must appear at the later part of the stream.

Addressing the correlation issue requires a careful sharing of the input of players with each other to guarantee that any edge that gets deleted was already inserted, while making sure there is also not too much correlation between their inputs (correlation generally makes the task of proving these lower bounds harder or even impossible). Proving the general round elimination argument with these restrictions also requires a different *direct-sum* result based on *internal information complexity* [BBCR10] and a corresponding *message compression* argument for internal information [JPY16] (in contrast to the external information complexity direct sum [CSWY01] used in Theorem 1 and its own message compression technique from [HJMR07]).

Putting Things Together

Let t_2 be the dimension of the outer matrix when constructing a 2-round hard instance, let n_{11} be the dimension of the smaller matrices and let $n_2 = t_2 \cdot n_1$ be the dimension of the final instance. Let us again consider the two natural choices of the algorithms:

- Either spend $\Omega(t_2^2)$ communication/space to learn something non-trivial about the t_r

hidden instances in the first pass;

- Or, spend communication/space proportional to what is needed to solve (most of) the t_2 instances in the second round independently

By using $s_2(n_2)$ to denote the communication/space needed for solving the problem on n_2-size instances, the above strategies for the algorithm suggests that:

$$s_2(n_2) \gtrsim \min\left(t_2^2\ ,\ t_2 \cdot s_1(n_1)\right) = \min\left(\left(\frac{n_2}{n_1}\right)^2\ ,\ \left(\frac{n_2}{n_1}\right) \cdot s_1(n_1)\right). \tag{3.1}$$

Using $s_1(n_1) = \Omega(n_1^2)$ as the base case by the lower bound of [DK20], we obtain

$$s_2(n_2) \gtrsim \min\left(\left(\frac{n_2}{n_1}\right)^2\ ,\ n_2 \cdot n_1\right) = n_2^{4/3},$$

by letting $n_2 = (n_1)^3$. Repeating this inductively using Eq (3.1) allows us to prove that

$$s_r(n) \gtrsim n^{1+1/2^{\Omega(r)}}.$$

This implies that $\Omega(\log\log n)$ rounds are needed by $\widetilde{O}(n)$-communication protocols for $O(1)$-approximation of matchings.

3.4 Distributed Triangle Detection

We proceed to describe the techniques of Theorem 5, which proves an $\Omega(\log\log n)$ lower bound for detecting triangles in the CONGEST model. This result does not use the framework of Theorem 1, and instead applies round elimination argument of [MNSW95] directly to the CONGEST model. However, the insights from [ANRW15], Theorem 1 and Theorem 4 in constructing round-specific hard instances were crucial in proving the lower bound.

The information bottleneck approach in proving distributed computing lower bounds that we discussed in Section 2.3.2 do not apply directly to triangle detection. Any reduction from two-party communication complexity fails because if the vertices of a triangle are divided between two players, at least one player gets two vertices, and they know all the edges of the triangle if one exists. A natural question is whether reductions from multi-party communication complexity can work. These have proven to give stronger lower bounds where two-party reductions are insufficient (see Section 2.3.3). [DKO14] uses three-party number-on-forehead[12] set disjointness [She13] to prove a lower bound for triangle detection in the broadcast model: if all players send the *same* messages across all the edges in every round, they give a lower bound of $\Omega(n/e^{\Theta(\sqrt{\log n})})$[13].

In the standard CONGEST model, however, it is easy to see that attempting a reduction from any two-party communication problem does not lead to any $\omega(1)$-round lower bounds.

[12] In this model, the players have their inputs on their foreheads, and they can see all the inputs of *everyone else*. Unlike in number-in-hand communication complexity that we discussed, the input-sharing aspect of number-on-forehead makes it harder to prove lower bounds [LS08].

[13] The lower bound of [DKO14] holds even when messages can be sent to every other player in each round, irrespective of whether an edge exists or not.

However the vertices of a triangle are partitioned, at least one of the two players must receive all the edges of at least two of the vertices of the triangle. This player will know the existence of the triangle trivially.[14] In our lower bound, we are faithful to the CONGEST model: each vertex is treated as a separate player, and the challenges are in applying [MNSW95] to the point-to-point communication aspects.

Combinatorial Component

Unlike in the construction of our graph streaming lower bounds, the combinatorial component of Theorem 5 is fairly simple and intuitive. [FGKO18] proves a one round lower bound of $\Omega(n)$ on the bandwidth of triangle detection, and we take inspiration from their hard instance (see Figure 3.11).

There are three important vertices, say $x, y, z \in V$. These vertices could form a triangle, but x does not know the identities of y, z and vice-versa. Hence, given a long list of neighbors, x is unable to distinguish whether it has edges to y and z, or only to other degree one vertices.

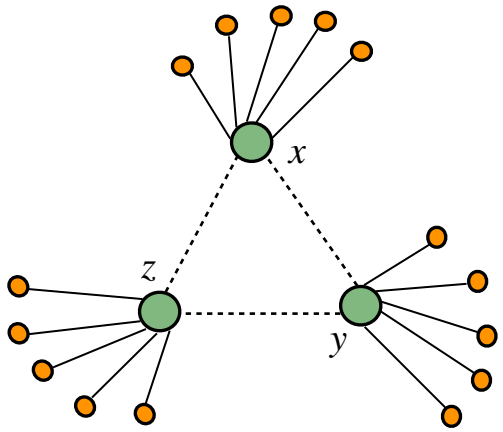


Figure 3.11: An illustration of the one round hard instance for Theorem 5. The dashed lines represent edges that may not exist. If these edges exist, there is a triangle in the input.

After one round of messages, the vertex x knows the identities of y and z, but it is still unknown whether an edge exists between y, z. This instance can be seen as a correlated version of two different Index problems:

1. Among the possible list of neighbors of y, vertex x points to z, and the message of y must convey information about whether (y, z) is an edge.

2. Similarly, among the list of neighbors of z, vertex x points to y, and the message from z must convey whether (y, z) is an edge.

The answers to both these problems are the same, based on whether (y, z) is an edge.

[14] Even if there are three parties, Alice, Bob and Charlie we cannot get $\omega(1)$-round lower bounds. Let the edges between Alice and Bob be sparser than the ones between Alice and Charlie or Bob and Charlie. Then, these edges between Alice and Bob can be communicated to Charlie with 1 round of communication, revealing the existence of a triangle to Charlie. This can be extended to any $O(1)$-players.

[FGKO18] shows that the messages from y or z must have $\Omega(n)$ length. See Figure 3.12 for an illustration.

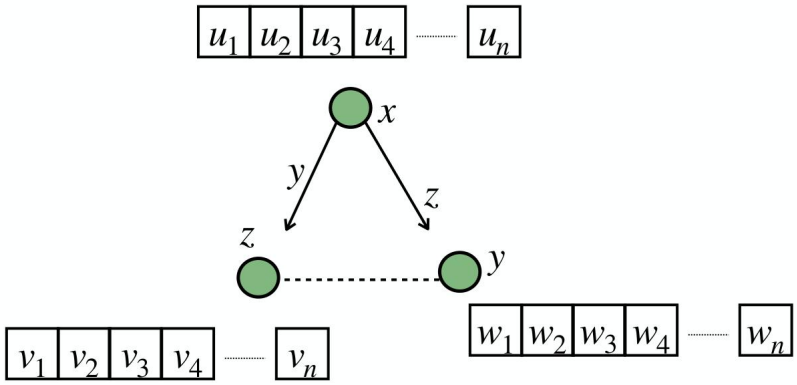


Figure 3.12: Viewing the one round hard instance as a correlated index problem. Vertex x does not initially know the pointers y, z but comes to know them only after the message from vertex y (which reveals that it may be a part of a triangle, hence fixing the pointer to z) and from vertex z. The message from y of $o(n)$ length contains very little information about whether z is present in the neighbor list.

The hardness of the instance relies on the *local view* of the vertices. We extend the instance of [FGKO18] to multiple rounds by simply padding each vertex from the 1-round instance with other degree one vertices. Now, the vertices in the 1-round instance do not know which of their neighbors are a part of the 1-round instance also, and they cannot communicate about the smaller instance to each other.

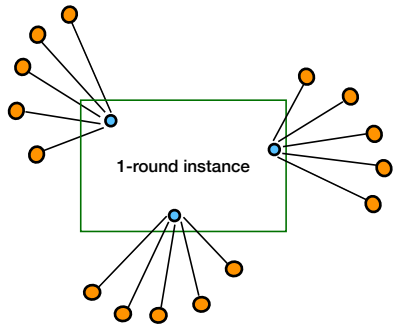


Figure 3.13: Padding with dummy vertices to create a 2-round hard instance.

This padding can be continued to create hard instances for any $r \geq 2$ rounds.

Information-theoretic Component

Typically in applications of round elimination, the property of the messages in the first round is that they do not reveal much information about the smaller instance I^*.[15] In our construction from Figure 3.13, this is simply not true, as the inner vertices can send messages to each other, and this may be heavily correlated with the input.

[15]This is true in the example we discussed in Section 2.2.2, and Theorems 1, 3 and 4.

The main challenge is dealing with the following two kinds of correlation:

(*i*) Message-Input correlation: Any message over an underlying edge reveals that an edge exists.

(*ii*) Message-Message correlation: All the messages one vertex sends can be correlated with each other.

We give a multistep way to sample the first round messages, so that the distribution in our simulation remains close to the original joint distribution of messages and the input in the r-round protocol, similar to Claim 3.1.3. This sampling approach is beyond the scope of this chapter, but a more detailed overview is presented in Section 7.3.

Putting Things Together

Let n_r be the size of the r-round instance. To implement the information-theoretic component, we require:

$$n_r = (n_{(r-1)})^c,$$

for some large constant c (see Section 7.3 for more details on why this is necessary). Hence, the sizes of our hard instances grow by a polynomial factor in each round. In the first round, the vertices do not convey information about the smaller $(r-1)$-round instance to each other, so they still require $s_{r-1}(n_{r-1})$ bandwidth. We have $s_1(n_1) = \Omega(n_1)$ from [FGKO18]. This gives a lower bound of:

$$s_r(n_r) \geq s_1(n_1) = \Omega(n_1) = (n_r)^{1/c^r}.$$

In $o(\log \log n_r)$ rounds, the bandwidth is $\omega(\log(n_r))$, which is beyond the limits of the CONGEST model.

3.5 Lower Bounds Beyond Embedding

All the results we have seen obey the following property:

> *The players do not learn anything about the hidden instance or group of instances in the first round.*

This may not be true for many problems; the smaller hidden instance could remain hard to solve with fewer passes or rounds even if $\Omega(1)$ information is revealed in the first round. This chapter covers our results where non-trivial information about the instance that the players are trying to solve may be revealed by the first round messages.

Let us consider the standard pointer chasing problem from Section 2.2.2 as an example. We briefly recall the definition. There are two sets A and B, each of size ℓ. Alice has a function f_1 from A to B and Bob has a function f_2 from B to A. We define a sequence of pointers starting from an arbitrary $z_0 \in A$, and following functions f_1 and f_2 alternately. We define:

$$z_1 = f_2(z_0), z_2 = f_1(z_1), z_3 = f_2(z_2), z_4 = f_1(z_3), \ldots$$

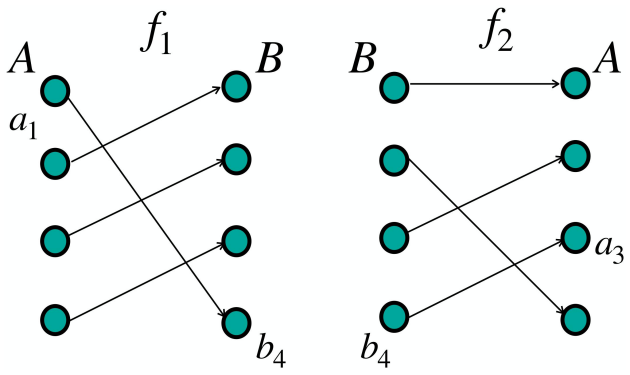


Figure 3.14: An illustration of the functions in pointer chasing. Alice holds f_1 and Bob holds f_2. If $z_1 = a_1$, $z_2 = b_4$ and $z_3 = a_3$.

Both players want to know the value of z_k. Alice begins speaking, and they can exchange messages with each other over many rounds. Refer to Figure 3.14 for an illustration.

If Alice sends an s-bit message about her function f_1 to Bob, this reveals a lot about the function to Bob, so round elimination arguments do not work. Pointer Chasing instances are typically analyzed by keeping track of exactly how much information about f_1 is revealed in each round. Informally, any s-bit message from Alice can only fix the pointers from $O(s)$ elements of A, and the rest of the function remains unknown to Bob [NW93, MYZ25].

In this chapter, the first result Theorem 6 proves an $\Omega(n - k \log n)$ lower bound for the Chained Index problem. It is slightly different from the other results of this chapter and directly analyzes Index instances when, on average, $\Omega(1)$ bits of information about the answer may already be known to the players.

The other two, Theorem 7 and Theorem 8 are an $\Omega(\log n / \log \log n)$-lower bound for any constant approximation of shortest path distances in semi-streaming and a tight lower bound on the approximation of set cover in streaming respectively. They rely on direct reductions to pointer chasing.

Chained Index. Let us recall the definition of the Chain communication problem.

For $n, k \geq 1$, in $\textsc{chain}_{n,k}$, there are k instances of $\textsc{index}_n$, (x_1, i_1), $\ldots$, (x_k, i_k) all with the same answer, distributed among $(k+1)$ players. Player 1 gets x_1, player j gets i_{j-1} and x_j for $1 < j \leq [k]$ and player $(k+1)$ gets i_k. Communication is one way starting from player 1 sending a message to player 2 and so on. The last player has to output the value of the answer to the k instances of $\textsc{index}_n$.

The problem can be viewed as giving the players multiple tries at Index: they have k different instances, and they try to find the same answer in each of these instances. We prove that they *cannot* do better than the trivial bound of $O(n)$ bits for a wide range of k.

With the multi-round view of the prior chapters, we look at each message from the players separately, and see how they change the distribution of the answer consecutively. After the message from the first player, the answer may be *partially* revealed, so we analyze Index instances where Alice and Bob already have some prior information in the form of the Biased Index problem (see Definition 8.3.5).

In standard analysis of Index, the initial input distribution is almost always assumed to be uniform over the string and the pointer from Bob. One exception is the Sparse Index problem [AKL16, Sag11], where the inputs to Alice and Bob are correlated. We use similar tools to analyze the Biased Index problem.

Let us say we have an Index instance with a corresponding distribution such that the entropy of the answer is δ for some $\delta < 1$: this corresponds to the initial information that Alice and Bob have, as a truly random bit would have entropy 1[16]. If the entropy further reduces by any ε-factor to $\delta - \varepsilon$, we show that there must be a corresponding message from Alice of length $\Omega(n\varepsilon - \log n)$. Any protocol must change the entropy by a constant factor, which implies an $\Omega(n - k \log n)$ lower bound for k players.

Shortest Paths. Theorem 7 follows the work of [GO16] (itself inspired by [FKM+05]) by reducing the problem to *many* instances of the classical pointer chasing problem on n-vertex graphs (see Section 2.2.2 for more details on these results). The key difference is in the analysis of the underlying pointer chasing instances.

More specifically, [GO16] uses the equality of pointer chasing problem (EQPC). Let u, v be the two vertices for which we want to find if the shortest path length is at most k. There are two instances of pointer chasing, say P_1 and P_2, with the starting vertex as u, v respectively. They model these two instances together directly as a graph so that the distance between u, v is at most k if the output of the two pointer chasing instances is equal. They prove that to solve EQPC in $((k/2) - 2)$-rounds, $\Omega(n/\text{poly}(k))$ communication is required. See Figure 3.15 for an illustration.

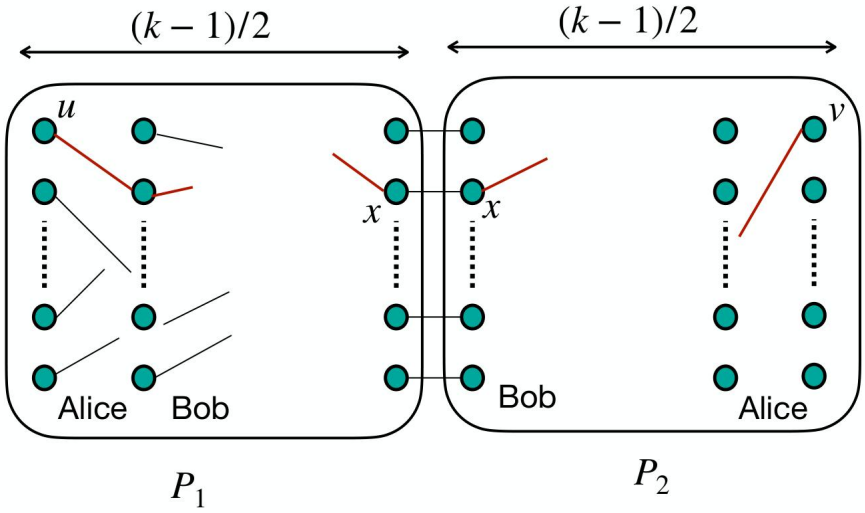


Figure 3.15: Equality of Pointer Chasing. When the final vertex is x for both the instances, there is a (u, v)-path of length k. Otherwise, the two vertices are not connected.

The lower bound of [GO16] uses a direct sum version with $t := n^{\Theta(1/k)}$ many instances of EQPC, and constructs a graph input where these many instances of EQPC do not interfere

[16] In round elimination arguments (or player elimination in this context), the entropy left in the answer was $1 - o(1)$. Theorem 6 analyzes Index even where the answer may have very little entropy left, even any constant less than 1.

with each other too much. The final lower bound is $n^{1+\Theta(1/k)}$ space for $O(k)$-passes, giving a lower bound of $\tilde{\Omega}(\log n)$ for semi-streaming space.

The distribution of the two instances in EQPC is a *product* distribution and corresponds to finding the shortest path distance to within an *additive* factor of two. On the other hand, we introduce and analyze a *correlated* input distribution for EQPC that allows us to prove a lower bound for constant factor *multiplicative* approximation of shortest path distance. This part is the key technical contribution of our lower bound and is proven using a combination of probabilistic and combinatorial arguments to address the introduced correlation in the inputs.

Finally, similar to [GO16], we also extend this lower bound to the direct sum version using information complexity [CSWY01, BJKS02, BBCR10]. But, this part of our argument also differs almost entirely from [GO16] since our underlying hard distribution is no longer a product distribution. We implement this step by relying on the vast body of work on message compression and information complexity in a blackbox manner, instead of the adhoc arguments of [GO16].

Streaming Set Cover. This result is perhaps our simplest and most surprising result. We use a direct reduction from pointer chasing and prove an optimal three-way space-pass-approximation tradeoff for the streaming set cover problem, essentially closing this line of research. Theorem 8 shows that for some graph problems, inherently, random instances are one of the hardest instances and the barriers we mentioned in Section 2.2.2 about random instances do not hold.

A p-pass hard instance consists of a *single* $(p-1)$-pass hard instance with an *unknown* universe; instead, it will be revealed to the algorithm at the end of the first pass—by our construction of the instance—as essentially a *random subset* of size $\approx n^{1-1/p}$ from the original universe (once the universe is revealed, the $(p-1)$-pass instance simply becomes the intersection of the original sets with this new universe). Importantly, the m sets of the input are repeated throughout and serve as hard instances for all p-passes. See Figure 3.16 for an illustration.

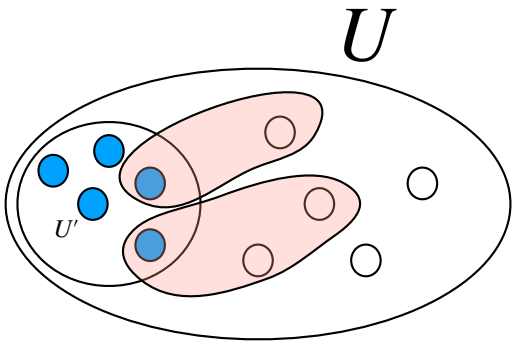


Figure 3.16: The blue elements are from a $n^{-1/p}$ fraction of the universe U. The input sets are restricted to U' after it is revealed, but initially they cover all the elements of U.

There is no hope in proving that the algorithm does not learn *any* information about the underlying $(p-1)$-pass hard instance, as the sets of the universe are unchanged. But,

we show that the algorithm does not learn anything *useful* about this underlying instance. This is similar to the arguments of pointer chasing [NW93, Yeh20, MYZ25] and we are able to prove the lower bound with a direct reduction to pointer chasing.

This concludes our (brief) overview of the techniques. Low level details can be found in the respective chapters.

4 Preliminaries

This section contains the technical preliminaries before we dive into our results.

Notation. We use $[t]$ to denote the set $\{1, 2, \ldots, t\}$ for any integer $t \geq 1$. For any graph $G = (V, E)$, we use $n := |V|$ to denote the number of vertices, and $m := |E|$ to denote the number of edges.

We use $\log(\cdot)$ to denote the binary logarithm and $\ln(\cdot)$ to denote the natural logarithm. We use $\exp(\cdot)$ to denote the exponential function, $\exp(x) = e^x$.

Some times, for a clarity of exposition, we may write $(a) \uparrow (b)$ to denote a^b. For any tuple $(x_1, \ldots, x_t)$ and $i \in [t]$, we define $x_{<i} := (x_1, \ldots, x_{i-1})$. We define $x_{>i}$ analogously. We define $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_t)$. For a set of tuples $\{(x, y) \mid x \in X, y \in Y\}$ for some sets X and Y, and $x \in X$, we define $(x, *) := \{(x, y) \mid y \in Y\}$; we define $(*, y)$ for $y \in Y$ analogously.

When there is room for confusion, we use sans-serif letters for random variables (e.g. A) and normal letters for their realizations (e.g. A). We use $\text{dist}(\mathsf{A})$ and $\text{supp}(\mathsf{A})$ to denote the distribution of A and its support, respectively. When implicit or clear from the presentation, the distribution and support will be omitted.

4.1 Models of Computation

For the purpose of our lower bounds, we shall work with a more powerful model than what is typically considered when designing streaming algorithms (this is the common approach when proving streaming lower bounds; see, e.g. [GM08, LNW14, BGW20]). In particular, we shall define streaming algorithms similar to branching programs as follows.

Definition 4.1.1 (Streaming algorithms). *For any integers $m, p, s \geq 1$, we define a p-pass s-space streaming algorithm A working on a length-m stream $\sigma = (\sigma_1, \ldots, \sigma_m)$ with entries from a universe U as follows:*

(i) The algorithm A has access to a read-only tape of uniform random bits r from a finite, but arbitrarily large range R without having to pay for the cost of storing these bits.

(ii) There is a function $f_A : \{0,1\}^s \times U \times R \to \{0,1\}^s$ that updates the state of the algorithm as follows. The algorithm starts with the state $S := 0^s$, and for $i \in [m]$, whenever A reads σ_i in the stream during its p passes, it updates its current state S to $S \leftarrow f_A(S, \sigma_i, r)$ (the algorithm is computationally unbounded when computing its next state).

(iii) At the end of the last pass, the algorithm outputs the answer as a function of its state S and random bits r.

(We note that this model is non-uniform and is defined for each choice of m, p, s individually.)

The other model of focus in this thesis is the distributed model with congestion.

Definition 4.1.2 (**Distributed Computing with Bandwidth**). *For any integers $n, r, b \geq 1$, we define an r-round b-bandwidth distributed algorithm A working on a graph $G = (V, E)$ with n vertices as follows:*

(*i*) *Each vertex from $v \in V$ has a unique identity from $[N]$ with $N = \text{poly}(n)$,*[1] *and it knows the identity of all its neighbors. Each vertex also has access to a private read-only finite tape R_v of random bits.*

(*ii*) *In any round $i \in [r]$, there is a function f_A that takes as input the identity of v, identity of all the neighbors of v, the messages received till round $(i-1)$ and the private random bits from R_v to generates messages for vertex v to send to all its neighbors in round i. All the messages sent have at most b bits.*

(*iii*) *The algorithm is computationally unbounded when computing function f_A.*

(*iv*) *At the end of the last round, the algorithm outputs the answer as a function of its identity, the identity of its neighbors, all the messages it receives and the random bits R_v.*

When the bandwidth $b = O(\log n)$, then Definition 4.1.2 gives the CONGEST model of computation from [Pel00].

Lastly, we also use many communication complexity models. As these models depend on the specific results, we postpone the definition to the relevant chapters.

4.2 Combinatorial Tools

4.2.1 Ruzsa-Szemerédi graphs

Let $G = (V, E)$ be an undirected graph, and $M \subseteq E$ be a matching in G. We say that M is an **induced matching** iff the subgraph of G induced on the vertices of M is the matching M itself; in other words, there are no other edges between the vertices of this matching.

Definition 4.2.1 (Ruzsa-Szemerédi Graphs [RS78]). *For any integers $r, t \geq 1$, a* bipartite *graph $G_{\text{rs}} = (L_{\text{rs}}, R_{\text{rs}}, E_{\text{rs}})$ is called an (r, t)-**Ruzsa-Szemerédi graph** (RS graph for short) iff its edge-set E can be partitioned into t* induced *matchings $M_1^{\text{rs}}, \ldots, M_t^{\text{rs}}$, each of size r.*

It is easy to construct bipartite (r, t)-RS graphs with n vertices on each side and parameter $t = (n/r)^2$ for any $r \geq 1$. More interestingly, and quite surprisingly[2], one can create much denser RS graphs as well; for instance, the original construction of these graphs due to Ruzsa and Szemerédi [RS78] create graphs with $r = n/2^{\Theta(\sqrt{\log n})}$ and $t = \Omega(n)$. Yet another construction is that of [FLN+02, GKK12] that shows that already for $r < n/2$, we can

[1] $[N]$ is known as the namespace of the vertices in the algorithm.

[2] The existence of dense RS graphs is surprising for the following reason: let us say an induced matching of sizd r exists in the graph. This prohibits r^2 other pairs of vertices from being edges of the graph, so as to not violate the inducedness of the matching. The RS graph constructions make sure that these "prohibited" vertex pairs overlap across many induced matchings so that the final density of the graph is large.

have $t = n^{\Omega(1/\log\log n)}$. See [BLM93, FLN+02, Alo02, TV06, AS06, AMS12, GKK12, FHS15, AB19, KKTY21] for various other constructions and applications of RS graphs. At the same time, proving upper bounds on the density of RS graphs turned out to be a notoriously difficult question (see, e.g. [Gow01, FHS15, CF13]) and the best known upper bounds, even for $r = \Omega(n)$, only imply that $t < n/2^{O(\log^* n)}$ [Fox11, FHS15].

We use the construction of RS graphs from [GKK12].

Proposition 4.2.2 ([GKK12])**.** *For any small enough constants $\delta_{\rm rs}, c_{\rm rs} > 0$ and for every large enough integer n, there exists a balanced bipartite (r,t)-RS graph with $2n$ vertices such that $r = \left(\frac{1}{2} - \delta_{\rm rs}\right) \cdot n$, $t = n^{c_{\rm rs}/\log\log(n)}$.*

4.2.2 Concentration Inequalities

We use the following standard form of Chernoff bound and its extension to negatively correlated variables in [PS97].

Proposition 4.2.3 (Chernoff Bound; cf. [DP09])**.** *Let $X_1, \ldots, X_n$ be n independent random variables in $[0,1]$ and $X := \sum_{i=1}^{n} X_i$. For any $\delta > 0$ and $\mu_{min} \leq \mathbb{E}[X] \leq \mu_{max}$,*

$$\Pr\left(X \geq (1+\delta)\cdot\mu_{max}\right) \leq \exp\left(-\frac{\delta^2\cdot\mu_{max}}{2+\delta}\right) \quad , \quad \Pr\left(X \leq (1-\delta)\cdot\mu_{min}\right) \leq \exp\left(-\frac{\delta^2\cdot\mu_{min}}{2}\right).$$

Moreover, the upper tail bound continues to hold as long $X_1, \ldots, X_n$ are negatively correlated, i.e., for every $S \subseteq [n]$,

$$\mathbb{E}\left[\prod_{i\in S} X_i\right] \leq \prod_{i\in S} \mathbb{E}\left[X_i\right].$$

4.3 Information-theoretic Tools

We now briefly introduce some definitions and facts from information theory that are needed in this thesis. We refer the interested reader to the text by Cover and Thomas [CT06] for an excellent introduction to this field, and the proofs of the statements.

- We denote the *Shannon Entropy* of a random variable A by $\mathbb{H}(\mathsf{A})$, which is defined as:

$$\mathbb{H}(\mathsf{A}) := \sum_{A\in\mathrm{supp}(\mathsf{A})} \Pr\left(\mathsf{A} = A\right)\cdot\log\left(1/\Pr\left(\mathsf{A} = A\right)\right) \tag{4.1}$$

- The *conditional entropy* of A conditioned on B is denoted by $\mathbb{H}(\mathsf{A} \mid \mathsf{B})$ and defined as:

$$\mathbb{H}(\mathsf{A} \mid \mathsf{B}) := \mathop{\mathbb{E}}_{B\sim\mathsf{B}}\left[\mathbb{H}(\mathsf{A} \mid \mathsf{B} = B)\right], \tag{4.2}$$

  where $\mathbb{H}(\mathsf{A} \mid \mathsf{B} = B)$ is defined in a standard way by using the distribution of A conditioned on the event $\mathsf{B} = B$ in Eq (4.1).

- The *mutual information* of two random variables A and B is denoted by $\mathbb{I}(\mathsf{A};\mathsf{B})$ and is defined:

$$\mathbb{I}(\mathsf{A};\mathsf{B}) := \mathbb{H}(A) - \mathbb{H}(A \mid B) = \mathbb{H}(B) - \mathbb{H}(B \mid A). \tag{4.3}$$

- The *conditional mutual information* is defined as $\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) := \mathbb{H}(\mathsf{A}\mid\mathsf{C}) - \mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C})$.

4.3.1 Useful Properties of Entropy and Mutual Information

We shall use the following basic properties of entropy and mutual information throughout. Proofs of these properties mostly follow from convexity of the entropy function and Jensen's inequality and can be found in [CT06, Chapter 2].

Fact 4.3.1. *Let* A, B, C, *and* D *be four (possibly correlated) random variables.*

1. $0 \le \mathbb{H}(\mathsf{A}) \le \log|\mathrm{supp}(\mathsf{A})|$. *The right equality holds iff* $\mathrm{dist}(\mathsf{A})$ *is uniform.*
2. $\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) \ge 0$. *The equality holds iff* A *and* B *are* independent *conditioned on* C.
3. $\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) \le \mathbb{H}(\mathsf{B})$ *for any random variables* $\mathsf{A},\mathsf{B},\mathsf{C}$.
4. Conditioning on a random variable reduces entropy*:* $\mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C}) \le \mathbb{H}(\mathsf{A}\mid\mathsf{B})$. *The equality holds iff* $\mathsf{A}\perp\mathsf{C}\mid\mathsf{B}$.
5. Subadditivity of entropy*:* $\mathbb{H}(\mathsf{A},\mathsf{B}\mid\mathsf{C}) \le \mathbb{H}(\mathsf{A}\mid C) + \mathbb{H}(\mathsf{B}\mid\mathsf{C})$.
6. Chain rule for entropy*:* $\mathbb{H}(\mathsf{A},\mathsf{B}\mid\mathsf{C}) = \mathbb{H}(\mathsf{A}\mid\mathsf{C}) + \mathbb{H}(\mathsf{B}\mid\mathsf{C},\mathsf{A})$.
7. Chain rule for mutual information*:* $\mathbb{I}(\mathsf{A},\mathsf{B}\,;\mathsf{C}\mid\mathsf{D}) = \mathbb{I}(\mathsf{A}\,;\mathsf{C}\mid\mathsf{D}) + \mathbb{I}(\mathsf{B}\,;\mathsf{C}\mid\mathsf{A},\mathsf{D})$.
8. Data processing inequality*: for a function* $f(\mathsf{A})$ *of* A, $\mathbb{I}(f(\mathsf{A})\,;\mathsf{B}\mid\mathsf{C}) \le \mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C})$.

We also use the following two standard propositions on effect of conditioning on mutual information.

Proposition 4.3.2. *For random variables* $\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}$, *if* $\mathsf{A}\perp\mathsf{D}\mid\mathsf{C}$, *then,*

$$\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) \le \mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C},\mathsf{D}).$$

Proof. Since A and D are independent conditioned on C, by Fact 4.3.1-(4), $\mathbb{H}(\mathsf{A}\mid\mathsf{C}) = \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{D})$ and $\mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{B}) \ge \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{B},\mathsf{D})$. We have,

$$\begin{aligned}\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) &= \mathbb{H}(\mathsf{A}\mid\mathsf{C}) - \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{B}) = \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{D}) - \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{B})\\ &\le \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{D}) - \mathbb{H}(\mathsf{A}\mid\mathsf{C},\mathsf{B},\mathsf{D}) = \mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C},\mathsf{D}). \qquad \blacksquare\end{aligned}$$

Proposition 4.3.3. *For random variables* $\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}$, *if* $\mathsf{A}\perp\mathsf{D}\mid\mathsf{B},\mathsf{C}$, *then,*

$$\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) \ge \mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C},\mathsf{D}).$$

Proof. Since $\mathsf{A}\perp\mathsf{D}\mid\mathsf{B},\mathsf{C}$, by Fact 4.3.1-(4), $\mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C}) = \mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C},\mathsf{D})$. Moreover, since conditioning can only reduce the entropy (again by Fact 4.3.1-(4)),

$$\begin{aligned}\mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C}) &= \mathbb{H}(\mathsf{A}\mid\mathsf{C}) - \mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C}) \ge \mathbb{H}(\mathsf{A}\mid\mathsf{D},\mathsf{C}) - \mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C})\\ &= \mathbb{H}(\mathsf{A}\mid\mathsf{D},\mathsf{C}) - \mathbb{H}(\mathsf{A}\mid\mathsf{B},\mathsf{C},\mathsf{D}) = \mathbb{I}(\mathsf{A}\,;\mathsf{B}\mid\mathsf{C},\mathsf{D}). \qquad \blacksquare\end{aligned}$$

4.3.2 Measures of Distance Between Distributions

We use two main measures of distance (or divergence) between distributions, namely the *Kullback-Leibler divergence* (KL-divergence) and the *total variation distance*.

KL-divergence. For two distributions μ and ν over the same probability space, the **Kullback-Leibler (KL) divergence** between μ and ν is denoted by $\mathbb{D}(\mu \mid\mid \nu)$ and defined as:

$$\mathbb{D}(\mu \mid\mid \nu) := \mathop{\mathbb{E}}_{a\sim\mu}\left[\log \frac{\mu(a)}{\nu(a)}\right]. \tag{4.4}$$

We also have the following relation between mutual information and KL-divergence.

Fact 4.3.4. *For random variables* $\mathsf{A}, \mathsf{B}, \mathsf{C}$*,*

$$\mathbb{I}(\mathsf{A}\,;\mathsf{B} \mid \mathsf{C}) = \mathop{\mathbb{E}}_{(B,C)\sim(\mathsf{B},\mathsf{C})}\Big[\mathbb{D}(\mathrm{dist}(\mathsf{A} \mid \mathsf{B}=B, \mathsf{C}=C) \mid\mid \mathrm{dist}(\mathsf{A} \mid \mathsf{C}=C))\Big].$$

Total variation distance. We denote the **total variation distance** between two distributions μ and ν on the same support Ω by $\|\mu-\nu\|_{\mathrm{tvd}}$, defined as:

$$\|\mu-\nu\|_{\mathrm{tvd}} := \max_{\Omega'\subseteq\Omega}\left(\mu(\Omega')-\nu(\Omega')\right) = \frac{1}{2}\cdot\sum_{x\in\Omega}\left|\mu(x)-\nu(x)\right|. \tag{4.5}$$

We use the following basic properties of total variation distance.

Fact 4.3.5. *Suppose* μ *and* ν *are two distributions for* $\mathcal{E}$*, then,* $\mu(\mathcal{E}) \le \nu(\mathcal{E}) + \|\mu-\nu\|_{\mathrm{tvd}}$.

When an event happens with small probability, conditioning on the event does not affect any distribution too much, as formalized by the following fact.

Fact 4.3.6. *For any distribution* μ *and event* $\mathcal{E}$*,*

$$\|\mu - (\mu \mid \neg\mathcal{E})\|_{\mathrm{tvd}} \le \Pr[\mathcal{E}].$$

We also have the following "over conditioning" property.

Fact 4.3.7. *For any random variables* $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$*,*

$$\|\mathsf{X}-\mathsf{Y}\|_{\mathrm{tvd}} \le \|\mathsf{XZ}-\mathsf{YZ}\|_{\mathrm{tvd}} = \mathop{\mathbb{E}}_{Z}\|(\mathsf{X} \mid \mathsf{Z}=Z) - (\mathsf{Y} \mid \mathsf{Z}=Z)\|_{\mathrm{tvd}}.$$

Proof. First, we prove the equality between the second term and the third term in the statement.

$$\begin{aligned}\|\mathsf{XZ}-\mathsf{YZ}\|_{\mathrm{tvd}} &= \frac{1}{2}\cdot\sum_{W,Z}\Pr[(W,Z)]\,|\Pr[\mathsf{XZ}=(W,Z)] - \Pr[\mathsf{YZ}=(W,Z)]| \\ &= \frac{1}{2}\cdot\sum_{W,Z}|\Pr[\mathsf{Z}=Z]\cdot(\Pr[\mathsf{X}=W \mid \mathsf{Z}=Z] - \Pr[\mathsf{Y}=W \mid \mathsf{Z}=Z])|\end{aligned}$$

$$
\begin{aligned}
&= \sum_Z \Pr[\mathsf{Z} = Z] \cdot \frac{1}{2} \sum_W |\Pr[\mathsf{X} = W \mid \mathsf{Z} = Z] - \Pr[\mathsf{Y} = W \mid \mathsf{Z} = Z]| \\
&= \sum_Z \Pr[\mathsf{Z} = Z] \cdot \|(\mathsf{X} \mid \mathsf{Z} = Z) - (\mathsf{Y} \mid \mathsf{Z} = Z)\|_{\mathrm{tvd}} \\
&= \mathop{\mathbb{E}}_Z \|(\mathsf{X} \mid \mathsf{Z} = Z) - (\mathsf{Y} \mid \mathsf{Z} = Z)\|_{\mathrm{tvd}}.
\end{aligned}
$$

Now we prove the inequality between the first term and the third term.

$$
\begin{aligned}
\|\mathsf{X} - \mathsf{Y}\|_{\mathrm{tvd}} &= \frac{1}{2} \cdot \sum_W |\Pr[\mathsf{X} = W] - \Pr[\mathsf{Y} = W]| \\
&= \frac{1}{2} \cdot \sum_W \left| \sum_Z \Pr[\mathsf{Z} = Z](\Pr[\mathsf{X} = W \mid \mathsf{Z} = Z] - \Pr[\mathsf{Y} = W \mid \mathsf{Z} = Z]) \right| \\
&\leq \frac{1}{2} \cdot \sum_W \sum_Z \Pr[\mathsf{Z} = Z] \, |\Pr[\mathsf{X} = W \mid \mathsf{Z} = Z] - \Pr[\mathsf{Y} = W \mid \mathsf{Z} = Z]| \\
&= \sum_Z \Pr[\mathsf{Z} = Z] \cdot \left(\frac{1}{2} \cdot \sum_W |\Pr[\mathsf{X} = W \mid \mathsf{Z} = Z] - \Pr[\mathsf{Y} = W \mid \mathsf{Z} = Z]| \right) \\
&= \mathop{\mathbb{E}}_Z \|\mathsf{X} \mid \mathsf{Z} = Z - \mathsf{Y} \mid \mathsf{Z} = Z\|_{\mathrm{tvd}}. \qquad \blacksquare
\end{aligned}
$$

We also have the following (chain-rule) bound on the total variation distance of joint variables.

Fact 4.3.8. *For any distributions μ and ν on n-tuples $(X_1, \ldots, X_n)$,*

$$
\|\mu - \nu\|_{\mathrm{tvd}} \leq \sum_{i=1}^{n} \mathop{\mathbb{E}}_{X_{<i} \sim \mu} \|\mu(X_i \mid X_{<i}) - \nu(X_i \mid X_{<i})\|_{\mathrm{tvd}}.
$$

The following Fano's inequality states that if a random variable Y can be used to estimate the value of X, then Y should "consume" most of entropy of X.

Proposition 4.3.9 (Fano's inequality)**.** *Given a binary random variable X and an estimator random variable Y and a function g such that $g(\mathsf{Y}) = \mathsf{X}$ with probability at least $1 - \delta$ for $\delta < 1/2$,*

$$
\mathbb{H}(\mathsf{X} \mid \mathsf{Y}) \leq H_2(\delta).
$$

The following Pinsker's inequality bounds the total variation distance between two distributions based on their KL-divergence,

Fact 4.3.10 (Pinsker's inequality)**.** *For any distributions μ and ν, $\|\mu - \nu\|_{\mathrm{tvd}} \leq \sqrt{\frac{1}{2} \cdot \mathbb{D}(\mu \mid\mid \nu)}$.*

Part II

Lower Bounds via Embedding Arguments

5 Maximal Independent Sets in Streaming

5.1 Introduction

We study the *Maximal Independent Set (MIS)* problem, namely, finding *any* independent set of the graph that is not a proper subset of another independent set in the semi-streaming model. An $O(\log n)$-pass semi-streaming algorithm for MIS follows from Luby's parallel algorithm [Lub85] (see also [LMSV11, KMVV13]). This was improved to an $O(\log \log n)$-pass algorithm in [ACG+15] (see also [GGK+18, Kon18b]). At the same time, the only streaming lower bounds known for MIS are the $\Omega(n^2)$ space lower bounds for *one*-pass algorithms obtained independently in [ACK19b, CDK19].[1]

We prove that the $O(\log \log n)$ passes in the algorithm of [ACG+15] is optimal.

Theorem 1 (restated). *For any $p \geq 1$, any p-pass streaming algorithm for finding any maximal independent set of n-vertex graphs with constant success probability requires $n^{1+1/(2^p-1)-o(1)}$ space. In particular, semi-streaming algorithms require $\Omega(\log \log n)$ passes.*

Theorem 1 fully settles the pass-complexity of MIS in the semi-streaming model. This answers a fundamental open question in the graph streaming literature—see, e.g., [CDK19, DK20]—on whether it is possible to even prove *any* multi-pass lower bound for MIS – our lower bound now almost matches the tradeoff of the algorithm of [ACG+15] for *every* number of passes.

We establish Theorem 1 by proving a stronger **rounds vs communication tradeoff** for MIS in the standard (multi-party) communication model. Here, the input graph is *edge*-partitioned between multiple players. In each round, player one sends a message to player two, who sends a message to player three, and they continue like this until the last player, who sends a message back to the first one. We would like the message of the last player in the last round to reveal an MIS of the input graph (see Section 5.3.1 for a formal definition). It is a well-known fact that communication lower bounds in this model also imply streaming lower bounds (see Proposition 5.3.2).

Result 1 (Formalized in Theorem 9)**.** *For any $r \geq 1$, any r-round $(r+1)$-party protocol for finding any maximal independent set of n-vertex graphs with constant success probability requires $n^{1+1/(2^r-1)-o(1)}$ communication.*

The bounds in Result 1 are again almost optimal in light of the algorithms in [ACG+15, GGK+18, Kon18b]. Theorem 1 now follows immediately from Result 1 using the standard reduction between the two models that we briefly discussed in Section 2.2.1. Given the

[1] There is also an $\Omega(n^{1/3})$-pass lower bound for semi-streaming algorithms that compute the *lexicographically first MIS (LFMIS)* [AGL+24, ACK19a]; however, it is known that LFMIS is a much harder (and quite different) problem than MIS (in most settings, including semi-streaming) and thus this result is not related to our discussion for finding *any* MIS.

central role MIS plays in most distributed communication models—wherein communication is often a bottleneck—our result on the communication complexity of MIS is of its own independent interest. We shall go over our techniques in detail in the streamlined overview of our approach in Section 5.2.

5.2 Technical Overview

We now present a streamlined overview of our technical approach. As stated earlier, our main result is a (multi-party) round vs communication lower bound for MIS in Result 1; we obtain our semi-streaming lower bound from this result immediately, using the standard connection between the two models (see Proposition 5.3.2). Thus, in this section, we solely focus on the communication complexity of MIS and postpone the streaming result to the formal proofs. We emphasize that this section oversimplifies many details and the discussions will be informal for the sake of intuition.

We start by presenting the intuition on what would have been the *ideal* lower bound approach for this problem. We then discuss how we are *actually* implementing this ideal approach and discuss our main technical ingredient, namely, *hierarchical embeddings*, that enables this implementation.

5.2.1 An "Ideal" Lower Bound Approach

We begin by looking at why the round elimination framework of [MNSW95] that was discussed in Section 5.2.4 cannot be applied directly.

Let us now see the *one-round* lower bounds for MIS in [ACK19b, CDK19] in this context.

Prior one-round lower bounds. The proofs in [ACK19b, CDK19] are as follows. Alice receives *two identical copies* of a random bipartite graph G with $\Theta(n^2)$ edges. Bob receives the following graph on the same set of vertices: pick a pair $\{u, v\}$ of vertices on the two sides of the bipartition of G and connect all vertices that are *not* a copy of these vertices to each other. See Figure 5.1. Computing the MIS of such a graph requires the players to figure out if (u, v) is an edge in G or not. This is because picking any vertex w other than u or v in the MIS "forces out" all the remaining vertices (possibly even the copies of u and v in the *same* copy as w), except for u, v of the other copy, that are now the only candidates to join the MIS (again, see Figure 5.1).

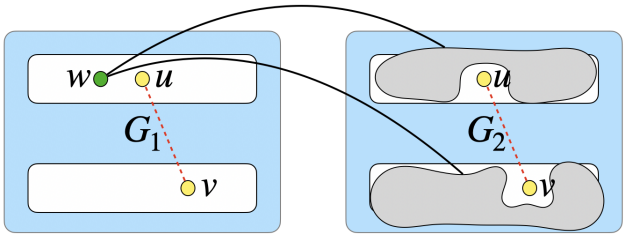


Figure 5.1: Alice's input is G_1 and G_2 which are copies of the same random bipartite graph G. Bob's input connects every vertex other than copies of u and v to each other. If we pick a vertex w in the left to join the MIS, only vertices u and v in the right remain to join the MIS (the figure only shows some of w's edges).

We can now do a round elimination to prove the lower bound[2]. Think of the 0-round "hard" problem as finding the MIS of a graph on a pair of vertices which may or may not be connected. The 1-round hard problem above consists of $\Theta(n^2)$ copies of this problem, with Bob's input making exactly one of them, corresponding to the pair $\{u, v\}$, the special one; for the players to solve this 0-round problem, Alice should have communicated enough information about $\{u, v\}$, which is not possible with $o(n^2)$ size messages, given Alice does not know which of the $\Theta(n^2)$ instances is special.

Beyond one-round lower bounds. Suppose we want to extend this approach beyond a single round. We can use the same approach of using two identical copies of the same **base graph** for Alice and give Bob an input that connects these copies. Instead of a random graph however, we need Alice's base graph to be a collection of *independent* copies of the 1-round hard instances. Here, 'independent' copies not only mean in a probabilistic sense, but also in a graph-theoretic sense: the edges of one instance should not appear between vertices of another as otherwise the instances become "corrupted", namely, they are no longer necessarily hard 1-round instances. On top of this, we also need these instances to be quite *large*, so that a 1-round lower bound for them incurs a significant communication cost in terms of the *entire* graph size (this is not a problem for the 1-round case as a 0-round protocol cannot have any communication no matter the input size).

A bit more specifically, to make this strategy work even for just a two-round $n^{1+\varepsilon}$ communication lower bound, for any constant $\varepsilon > 0$, we need Alice's base graph to have the following properties:

1. It must contain $(n^{1+\varepsilon})$ 1-round instances so Alice cannot reveal too much about a random instance;
2. Each 1-round instance needs to have $(n^{(1+\varepsilon)/2})$ vertices so that the quadratic 1-round lower bound implies that the second round of the protocol needs $(n^{(1+\varepsilon)/2})^2 = n^{1+\varepsilon}$ communication;
3. The instances cannot have edges between each others' vertices so as not to "corrupt" one another.

Given that one-round instances have quadratic number of edges, satisfying all these constraints means having a base graph on n vertices with $n^{2+2\varepsilon}$ edges or even pairs of vertices![3]

This highlights an inherent difficulty (basically, impossibility) of implementing this natural type of round elimination argument for proving communication lower bounds on graphs. In fact, almost all prior round-sensitive communication lower bounds on graphs (which are super linear in n) rely on more complicated arguments. For instance, the lower bounds in [AR20, CKP+21a, Ass22] from Section 2.2.3 require proving a lower bound for much harder intermediate problems to be able to carry out the inductive argument (e.g., see *permutation*

[2]The lower bounds in [ACK19b, CDK19] instead directly reduce the problem to the one-round communication complexity of the *Index* problem [Abl93, KNR95]; but, they can also be proven in this (almost equivalent) way. In fact, this is the same as our round elimination proof of Index in Section 3.1.1.

[3]Making 1-round instances sparser or similar modifications does not work either, as that forces us to pick even larger 1-round instances in Line 2, leading to the same exact contradiction.

hiding generators in [CKP+21a, AS23a]; see also [Ass23] for a summary of these recent advances). The only exception is the very recent work of [KN24] that shows how to extend a similar one-round lower bound approach (but, for the matching problem in [GKK12]) to *two*-rounds, and prove a two-pass semi-streaming lower bound for approximate matching (we mention similarities and differences of these two approaches throughout this section).

5.2.2 Our *Actual* Lower Bound Approach: Hierarchical Embeddings

Given the futility of the direct approach in implementing [MNSW95], we will not only start with many hard instances of the $(r-1)$-round problem in our r-round instance, but we will get the players to also solve *many* of these instances in the subsequent rounds (this is inspired by the aforementioned two-round lower bound of [KN24]). Specifically, we have $q \cdot p$ "hard" $(r-1)$-round *sub-instances* $\{I_{i,j}\}_{i\in[q],j\in[p]}$, each on b vertices for some *large* parameters q, p, and b as a function of n. These instances are again created based on a single base graph which is copied twice and given to the players; they are then connected via a clique structure, which identifies a **sub-instance group** $I_{t,*} := I_{t,1}, \ldots, I_{t,p}$ for a random chosen $t \in [q]$ as special instances. Restricting ourselves again to a two-round $n^{1+\varepsilon}$-communication lower bound, what we need from this base graph is the following:

1. We need $q \cdot p = n^{1+\varepsilon}$ so Alice's first message does not reveal too much information;
2. We also need $p \cdot b^2 = n^{1+\varepsilon}$ so Bob's message cannot solve *all* p special instances (each on b vertices admitting a 1-round $\Omega(b^2)$-communication lower bound) in the second round;
3. **Inducedness property:** None of $q \cdot p$ sub-instances can have an edge between vertices of another sub-instance so as to not "corrupt" one another. Also, each sub-instance group $I_{i,*}$ for $i \in [q]$ should be *vertex disjoint* so that finding an MIS of their vertices requires solving all sub-instances.

Satisfying these constraints requires having a base graph on n vertices with $q \cdot p \cdot b^2$ vertex-pairs. *If* we could set $p \approx q \approx n/b$, then, at least we will pass the *basic* test of "vertex-pair counting". This in turn gives us a base graph with $\approx n^2/b^2$ sub-instances among which $\approx n/b$ are special (each admitting a 1-round $\Omega(b^2)$-communication lower bound). Optimizing the parameters by setting $n^2/b^2 = (n/b) \cdot b^2 = n^{1+\varepsilon}$, implies that we can *hope for* obtaining an $\approx n^{4/3}$ lower bound (for $b \approx n^{1/3}$) for two-round algorithms this way.

Remark 5.2.1. *This is precisely the tradeoff achieved by the protocols for MIS in [ACG+15, GGK+18], albeit in three rounds. In the first round, they sample n/b vertices and send all their edges in n^2/b^2 communication and compute their MIS. In the second round, they remove the neighbors of the vertices in the MIS. They show that this reduces the maximum degree of remaining vertices to $\approx b$ only, and so in the third round they can send the remaining edges in $\approx n \cdot b$ communication and compute an MIS of the whole graph.*

This approach allows us to mimic these tradeoffs for the lower bound. However, we do not match this tradeoff exactly, because the current upper bound results use one extra pass to remove the neighbors of the vertices added to the MIS.

Let us see how a base graph with these parameters should look like, especially given that the third requirement is quite strict: the induced subgraph of the base graph on vertices of $I_{i,*}$ for $i \in [q]$, should be a vertex-disjoint union of graphs in $I_{i,*}$, see Figure 5.2.

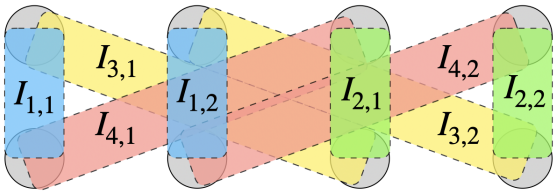


Figure 5.2: An illustration of the base graphs discussed above. Here $q = 4$ and $p = 2$ and each box denotes a single sub-instance (with different colors for different $I_{i,*}$ for $i \in [4]$). There are no edges between vertices of $I_{i,*}$ except for edges of the same sub-instances, which themselves are vertex-disjoint. While the figure may suggest sub-instances as being bipartite graphs, that crucially cannot be true in the actual construction.

The hierarchical embedding technique. Let us now reiterate how our *hierarchical embedding* technique works at a high level. For technical reasons, our r-round lower bound will be against $(r+1)$-party (number-in-hand) protocols instead of just two parties.

We pick $q_r \cdot p_r$ hard $(r-1)$-round sub-instances with r-players in a set $\mathcal{I} := \{I_{i,j}\}_{i \in [q_r], j \in [p_r]}$, each on b_r vertices, for parameters q_r, p_r, and b_r to be determined soon.

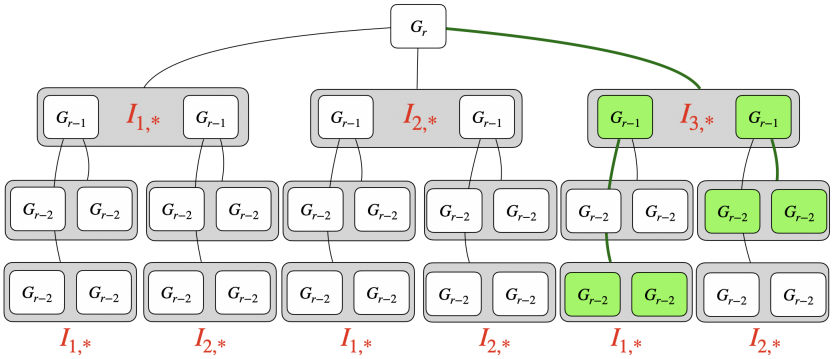
We use our **combinatorial constructions** to embed all these instances into a single base graph G with the inducedness property mentioned earlier (and then copied identically twice as before). For every $a \in [r]$, the input of the a^{th} player in the r-round instance is the collection of the inputs of all a^{th} players in sub-instances in $\mathcal{I}$. The $(r+1)^{\text{th}}$ player in the r-round instance gets a clique-subgraph as before that "points" to one of the sub-instance groups $I_{t,*}$ for a random $t \in [q_r]$.

This way, the input consists of a *hierarchy* of different-round instances of the problem, and the players have to solve multiple root-to-leaf paths of this hierarchy to obtain a solution to the entire input as well. Our **round elimination argument** then shows that the first round of the protocol cannot reveal enough information about the special sub-instance group to allow solving *all* of them in the remaining $(r-1)$ rounds. See Figure 5.3 for an illustration.

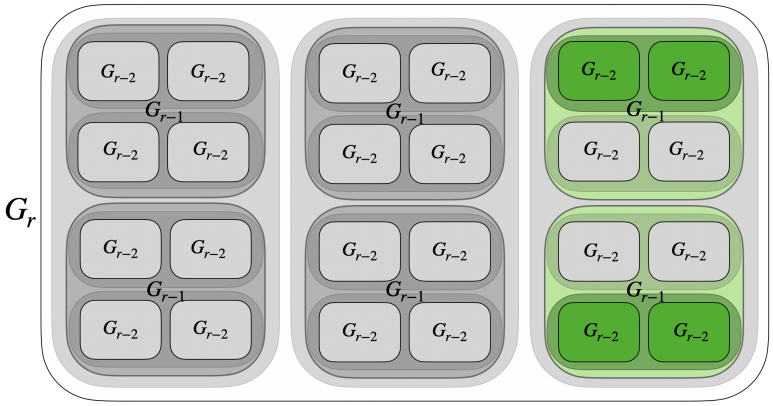
Finally, the choice of the parameters q_r, p_r and b_r is as follows. Let $C(n, r)$ denote the lower bound on the communication cost we hope to prove for r-round protocols on n-vertex graphs. Then,

$$\begin{aligned} q_r \cdot p_r \approx C(n,r) &: \text{so players cannot reveal much about} \\ &\quad\ p_r \text{ special sub-instances in the first round} \\ p_r \cdot C(b_r, r-1) \approx C(n,r) &: \text{so solving the } p_r \text{ special } b_r\text{-size} \\ &\quad\ \text{sub-instances in } r-1 \text{ rounds is hard.} \end{aligned}$$

Picking these parameters optimally, while accounting for the restrictions imposed by the combinatorial construction (and an $n^{o(1)}$-"loss" on size of these graphs compared to absolute



(a) The hierarchical instances form a conceptual tree, and the players need to "solve" certain root-to-leaf paths.



(b) The input graph is also a hierarchy of *induced* subgraphs (this figure shows these graphs in a vertex-disjoint way for simplicity of drawing, but in reality, vertices of these graphs are highly "tangled").

Figure 5.3: An illustration of the hierarchical embedding technique.

best-imaginable bounds), gives us our $\approx n^{1+1/(2^r-1)}$ communication lower bound for r-round $(r+1)$-party protocols.

5.2.3 Combinatorial Part: Dense Graphs with Many Induced Subgraphs

We now discuss the construction of our *base graphs* for our hierarchical embeddings. Recall that we need an n-vertex graph G with $q \approx n/b$ **subgraph groups** $\{H_{i,*}\}_{i\in[q]}$, each with $p \approx n/b$ *vertex-disjoint* subgraphs $\{H_{i,1}, \ldots, H_{i,p}\}_{i\in[q]}$ on b vertices each. In addition, the **inducedness property** requires that there are no other edges between the vertices of each subgraph group (other than the edges of the subgraphs themselves). Refer back to Figure 5.2 from earlier for an illustration.

Our constructions work with an *outer* graph, which, instead of induced matchings in RS graphs, contains many **Unique Path Collections (UPCs)** that can almost be seen as "stretching" an induced matching (a collection of paths of length 1), to longer paths (see Definition 5.4.3 for the formal definition and Figures 5.4 and 5.7 for illustrations). The 'almost' part however is due to a subtle difference: the property of UPCs goes beyond only

their *induced* subgraphs – instead, these paths are not only induced on their vertices, but also are the *unique* shortest path between their endpoints even if one uses outside vertices. See Figure 5.4 for an example.

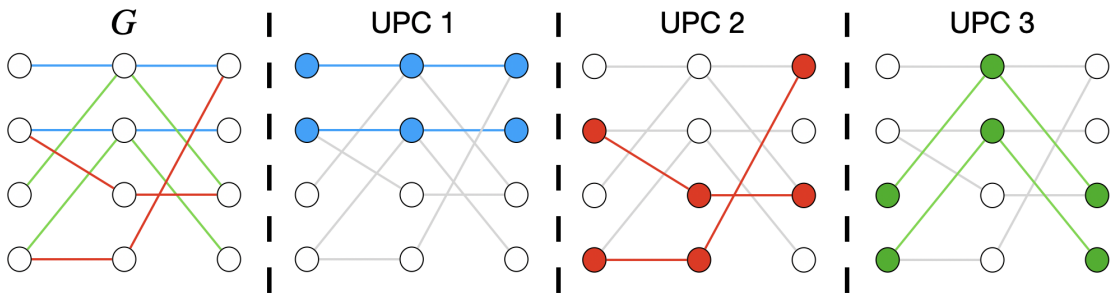


Figure 5.4: An example of UPCs in the graph G on the left. In each UPC, there is no path of length 2 from a starting vertex to an end vertex unless the path belongs to the UPC itself.

Such outer graphs have been already studied extensively in the context of shortcutting sets and spanners; see, e.g., [Hes03, AB16, CE06, HP21, BH22, LWWX22, WXX23] and references therein. For our purpose, we need a slight variation that follows from the approach in [AB16], which is itself based on the original RS graph constructions in [RS78] (the variation allows for "packing" many paths *together* in a UPC, as opposed to the *path-wise* guarantee of prior work). Nevertheless, it appears that the way we use these graphs is completely different from prior work. We are also not aware of any prior applications of these graphs to streaming or communication lower bounds, or to the maximal independent set problem in any other computational model[4].

Having obtained these outer graphs, our **embedding product** works by replacing, for every $i \in [q]$, each *path* in the i^{th} UPC of these graphs with the subgraphs in $H_{i,*}$ that we would like to have in our base graph. The only requirement we need from subgraphs $\{H_{i,j}\}_{i,j}$ is that they should have a small *chromatic number*; in particular, as long as they are k-*colorable* graphs, we can also embed them in UPCs with paths of length $k-1$ by mapping each of their color classes to one of the vertices of the path in a careful way. The strong guarantee of our outer graphs can then be used to argue that these k-colorable graphs cannot have inserted an edge between vertices of each other (i.e., break the inducedness property), as such an edge can be traced backed to a "shortcutting path" in the outer graph which cannot exist. See Figure 5.5 for an illustration.

Finally, in our application to MIS, we start with 2-colorable graphs as our base case instances that are hard for 0-round protocols. The identical copying plus adding a clique between some parts of the two copies that we mentioned earlier, with some modifications, leads to a 4-colorable hard instance for 1-round protocols. Following the same pattern thus leads to 2^{r+1}-colorable graphs for r-round protocols. Our construction allows for setting

[4]In terms of parameters, we are working with almost the exact opposite of prior work: we aim to *maximize* their density and size of UPCs—crucially of size $n^{2-o(1)}$ and $n^{1-o(1)}$, respectively—while *minimizing* their diameter—crucially of size $n^{o(1)}$—unlike prior work that aim to achieve a diameter of $n^{\Omega(1)}$; we only *have to* increase the diameter per each level of our hierarchical embedding to accommodate the recursive strategy.

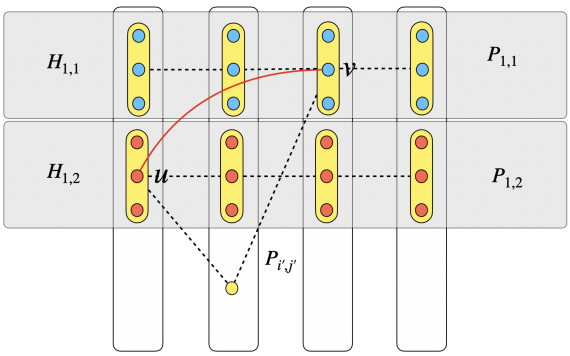


Figure 5.5: If the solid (red) edge between u and v is inserted by any other k-colorable graph $H_{i,j}$ for $i \neq 1$, we will also find a shortcut between "outer" versions of u and v, which cannot exist in the outer graph.

$q \approx p \approx n^{1-o(1)}$ for embedding any $n^{o(1)}$-colorable graphs, and since we will only need to continue the lower bound for $r \approx \log\log n$ rounds, we will have more than enough room to carry out the entire plan.

5.2.4 Information-Theoretic Part: Round Elimination via Direct-Sum

We will now discuss the information-theoretic parts of our lower bound. The combinatorial parts of our lower bound gives us a quite strong boost here: we can now *solely* focus on the ability of the protocols in "traversing" the hierarchical tree of instances we discussed earlier (see Figure 5.3), without worrying about *any* interference between underlying instances of the problem (there is none owing to the inducedness property of our construction). In other words, the communication lower bound here is practically independent of the MIS problem and is essentially (although not quite) only about the abstract problem of traversing the hierarchical tree:

- The first r players receive $q_r \cdot p_r$ "hard" $(r-1)$-round sub-instances

$$\mathcal{I} := \{I_{i,j} \mid i \in [q_r], j \in [p_r]\}$$

  ; the last player receives an index $t \in [q_r]$;
- The goal for the players is to solve *all* **special** sub-instances of $I_{t,*}$ in r rounds.

This problem is similar in spirit to the *tree-pointer jumping* problem of [CCM08]. The crucial difference is that tree-pointer jumping only has *one* special instance per level (think of $p_r = 1$ above), whereas for us, we need the players to solve *many* "small" sub-instances—which, individually, are not hard to solve within the budget of even the $(r-1)$-round protocols (this is precisely our deviation from the ideal-type lower bounds to our actual lower bound). A similar problem was also very recently introduced by [KN24] only for 2-round protocols as their *HiddenStrings* problem. However, the techniques in [KN24]—based on the two-way communication complexity of the Index problem [JRS09]—are inherently only for 2-round protocols and do not generalize beyond.

Our second technical contribution is a lower bound for this problem (in a bit less abstract form).

An intuitive "proof". Suppose we have a protocol π with $s = o(p_r \cdot q_r)$ communication on the distribution $\mathcal{G}_r$ of r-round instances. We can first show that, in its first round, π can only reveal $O(s/q_r) = o(p_r)$ bits in total about all instances in $I_{t,*}$ (as these are an unknown $1/q_r$ fraction of the inputs in $\mathcal{I}$). This means that *on average* π has only revealed $o(1)$ bits about any one instance $I_{t,j}$ for $j \in [p_r]$. Thus, in the remaining $r-1$ rounds, it needs to solve p_r copies of the $(r-1)$-round problem, where each copy, on average, is only $o(1)$ statistical distance away from the hard $(r-1)$-round distribution $\mathcal{G}_{r-1}$. This second problem has a flavor of a direct-sum result[5]. Hence, using a direct-sum style argument (say, as in [BBCR10]), we should also be able to say that in the remaining $(r-1)$ rounds, π needs to communicate p_r times the communication cost of solving one $(r-1)$-round hard instance. This gives us the tradeoff stated at the very end of Section 5.2.2.

Unfortunately, making this intuition precise is hindered by the fact that in this direct-sum scenario, the p_r $(r-1)$-round instances, in addition to their marginal deviation of $o(1)$ from $\mathcal{G}_{r-1}$, also have become *correlated* with each other for a total of $o(p_r)$ bits. In other words, the first message of π can also change their *joint* distribution by revealing $o(p_r)$ bits about them collectively. In general, one cannot hope for a direct-sum result to hold over such non-independent instances.

Our actual proof. To tackle this obstacle, we apply an *information complexity direct-sum* to the previous round elimination arguments of [ANRW15, AKZ22]. Specifically, we show how to obtain a $(r-1)$-round protocol π' for a *single* hard $(r-1)$-round instance using the protocol π:

High level description of the protocol π' for $I_{r-1} \sim \mathcal{G}_{r-1}$:

1. **Sampling step:** *Sample* the first message Π_1 of π and a family of sub-instances $\mathcal{I}$ from $\mathcal{G}_r$, i.e., $(\Pi_1, \mathcal{I}) \sim \mathcal{G}_r$. Then, sample $k \in [p_r]$ uniformly at random and *replace* $I_{t,k}$ in $\mathcal{I}$ with the input instance I_{r-1} (where t is the index of the special sub-instance group of $\mathcal{I}$).
2. **Simulation step:** Run the protocol π on the modified $\mathcal{I}$ from its *second* round onwards and output the answer of π on the special sub-instance $I_{t,k}$.

The analysis consists of two parts, one for each step of the protocol. The first part shows how the players can sample the joint variables $(\Pi_1, \mathcal{I})$ and embed I_{r-1} inside $\mathcal{I}$—which crucially is independent of Π_1 in this sampling, although it should *not* be in the distribution of $(\Pi_1, \mathcal{I})$ in π—using a distribution that is *close* to $\mathcal{G}_r \mid I_{i,j} = I_{r-1}$. This sampling is inspired by [ANRW15, AKZ22] with various modifications to account for the considerable differences between the two communication models. Proving this part is based on a similar argument as

[5]Recall that a direct-sum result argues that solving m *independent* copy of a problem becomes m times harder than solving one copy; in this context, solving m copies needs m times more communication than one copy.

in the intuitive "proof" above that says that the marginal distribution of $I_{i,j}$ is not affected that much by the first-round message Π_1.

The second part of the argument, however, deviates entirely from [ANRW15, AKZ22], because in their communication model, the new protocol π' already has the desired communication cost for an $(r-1)$-round protocol (given their focus on maximum message-size *per* vertex of the graph); thus, their arguments finish at this point. However, for us, the protocol π' has the same communication cost as π even though it is being run on a *much smaller* instance. Hence, there is no contradiction that π' can solve $\mathcal{G}_{r-1}$ in $(r-1)$ rounds as its communication cost is way above the bar anyway.

To handle this part, we instead work with the *information cost* of the protocols (in place of their communication cost) for a natural *multi*-party generalization of two-party *external* information cost in [BBCR10] (see Section 5.3.2 for the definition). Our approach is then to show that even though π' may communicate as much as π, its information cost is a factor $1/p_r$ smaller than the information cost of π. This is because I_{r-1} is embedded in one of the p_r-many special sub-instances randomly whose identity is *hidden* to the protocol π. But, this creates another challenge: information cost, unlike communication, is a function of the underlying distribution, and in π', the distribution of the inputs that π is simulated on is not the original distribution $\mathcal{G}_r$, only statistically close to it. Thus, even though we know that π has a *low* information cost on $\mathcal{G}_r$, we cannot guarantee it also has a low information cost in the simulation step and conclude that π' has a low information cost[6].

The challenge outlined above is not unique to our problem and has been studied before also, e.g., for direct-*product* results in [BRWY13b, BRWY13a, BW15] but typically for *internal* information cost and two-party protocols. For our purpose, we use a *message compression* technique due to [HJMR07] (see also [BG14]) to reduce the *communication* cost of the protocol π in the simulation step down to its information content. This leads to some minimal loss in the parameters that can be easily handled in our setting. But now, the change in the underlying distribution of the simulation step cannot affect the communication cost, which allows us to obtain a too-good-to-be-true protocol π' for $\mathcal{G}_{r-1}$, which leads to our desired contradiction.

This concludes our overview. In the rest of the paper, we formalize the ideas and approaches discussed in this section and prove Theorem 1 and Result 1 formally.

5.3 Preliminaries

5.3.1 (Multi-Party) Communication Complexity

We work in the standard multi-party number-in-hand communication model; we provide some basic definitions here and refer the interested reader to the excellent textbooks [KN97, RY20] on communication complexity for more details.

There are $k \geq 1$ players $P_1, \ldots, P_k$ in this model who receive a partition of the edges of an input graph $G = (V, E)$. The players follow a protocol π to solve a given problem on G, say, finding an MIS. They have access to a *shared* tape of randomness, referred to as *public randomness*, in addition to their own *private randomness*. In each *round*, P_1 writes a message

[6] In general, information cost of a protocol on two statistically close distributions can be vastly different.

on a board visible to all parties, followed by P_2, all the way to P_k. Each message can only depend on the private input of the sender, content of the blackboard, public randomness, and private randomness of the sender. At the end of the last round, the last player P_k writes the answer to the blackboard.

Definition 5.3.1. *For any protocol π, the **communication cost** of π, denoted by $CC(\pi)$, is defined as the worst-case (maximum) total length of messages, measured in bits, communicated by all players on any input. We assume that all **transcripts**, i.e., the set of all messages written onto the blackboard, in π have the same worst-case length (by padding).*

Communication complexity and streaming. The following standard result—dating back (at the very least) to the seminal paper of [AMS96] that introduced the streaming model—relates communication cost of protocols and space of streaming algorithms.

Proposition 5.3.2 (cf. [AMS96])**.** *For any $p \geq 1, s \geq 1$, and $\delta \in (0,1)$, suppose there is a p-pass s-space δ-error streaming algorithm A for some problem $\mathbb{P}$. Then, for any integer $k \geq 1$, there also exists a k-party protocol π with p rounds, communication cost $CC(\pi) \leq p \cdot k \cdot s$, and error probability at most δ, for the same problem $\mathbb{P}$.*

Proof. Consider the stream $\sigma = \sigma_1 \circ \cdots \circ \sigma_k$ where σ_i is the input to the player P_i in π (ordered arbitrarily in the stream). Player P_1 runs A on σ_1 and writes the memory content on the board, which allows player P_2 to continue running A on σ_2, and so on and so forth. This allows the players to run one pass of A using communication cost at most $k \cdot s$. The players can continue this in p rounds, faithfully simulating the p passes of the algorithm, and at the end P_k can read the answer of A from the memory and outputs it on the blackboard.

This way, π will have the same probability of success as A, uses at most $p \cdot k \cdot s$ communication and p rounds of communication. ∎

Proposition 5.3.2 allows us to translate communication lower bounds into streaming lower bounds. In particular, this result reduces proving Theorem 1 to proving Result 1.

5.3.2 Information Cost and Message Compression

We also work with the notion of *information cost* of protocols that originated in [CSWY01] and has since found numerous applications (see, e.g. [Wei15] for an excellent survey of this topic).

There are various definitions of information cost that have been considered depending on the application. The definition we use—which is a natural multi-party variant of the standard *external information cost* [BBCR10]—is best suited for our application.

Definition 5.3.3. *For any multi-party protocol π whose inputs are distributed according to the distribution μ, the **(external) information cost** is defined as:*

$$IC(\pi, \mu) := \mathbb{I}(\mathsf{G}\,; \Pi \mid \mathsf{R}),$$

where G denotes the random variable for the input graph sampled from μ, Π denotes the set of all messages written on the board, and R is the public randomness.

In words, multi-party external information cost measures the information revealed by the entire protocol about the entire graph to an *external* observer. The following result follows from [BBCR10].

Proposition 5.3.4 (cf. [BBCR10])**.** *For any multi-party protocol π on any input distribution μ,*

$$IC(\pi, \mu) \leq CC(\pi).$$

Proof. We have,

$$\mathrm{IC}(\pi,\mu) \underset{(1)}{=} \mathbb{I}(\mathsf{G}\,;\mathsf{\Pi} \mid \mathsf{R}) \underset{(2)}{=} \mathbb{H}(\mathsf{\Pi} \mid \mathsf{R}) - \mathbb{H}(\mathsf{\Pi} \mid \mathsf{G},\mathsf{R}) \underset{(3)}{\leq} \mathbb{H}(\mathsf{\Pi} \mid \mathsf{R}) \underset{(4)}{\leq} \mathbb{H}(\mathsf{\Pi}) \underset{(5)}{\leq} \log|\mathrm{supp}(\mathsf{\Pi})| \underset{(6)}{=} \mathrm{CC}(\pi);$$

here, (1) is by the definition information cost, (2) is by the definition of mutual information, (3) is by the non-negativity of entropy (Fact 4.3.1-(1)), (4) is because conditioning can only reduce the entropy (Fact 4.3.1-(4)), (5) is because uniform distribution has the highest entropy (Fact 4.3.1-(1)), and (6) is by the definition of worst-case communication cost. ∎

Message compression. We use *message compression* to reduce communication cost of protocols close to their information cost in certain settings. The following result is due to [HJMR07] which was further strengthened slightly in [BG14]. We follow the textbook presentation in [RY20].

Proposition 5.3.5 (c.f. [RY20, Theorem 7.6])**.** *Suppose Alice knows two distributions $\mathcal{A}, \mathcal{B}$ over the same set U and Bob only knows $\mathcal{B}$. Then, there is a protocol for Alice and Bob to sample an element according to $\mathcal{A}$ by Alice sending a single message of size*

$$\mathbb{D}(\mathcal{A} \mid\mid \mathcal{B}) + 2\log(1 + \mathbb{D}(\mathcal{A} \mid\mid \mathcal{B})) + O(1) \tag{5.1}$$

bits in expectation. This protocol has no error.

We note that somewhat weaker bounds on the KL-Divergence in Eq (5.1) already suffice for our purposes. Thus, to simplify the exposition, we use,

$$\begin{aligned}\mathbb{D}(\mathcal{A} \mid\mid \mathcal{B}) + 2\log(1 + \mathbb{D}(\mathcal{A} \mid\mid \mathcal{B})) + O(1) &\leq \mathbb{D}(\mathcal{A} \mid\mid \mathcal{B}) + 10\mathbb{D}(\mathcal{A} \mid\mid \mathcal{B}) + O(1) \\ &\qquad (\text{as } \log(1+x) \leq 5x \text{ for every } x \geq 0) \\ &\leq c_{\mathrm{comp}} \cdot (\mathbb{D}(\mathcal{A} \mid\mid \mathcal{B}) + 1)\,, \end{aligned} \tag{5.2}$$

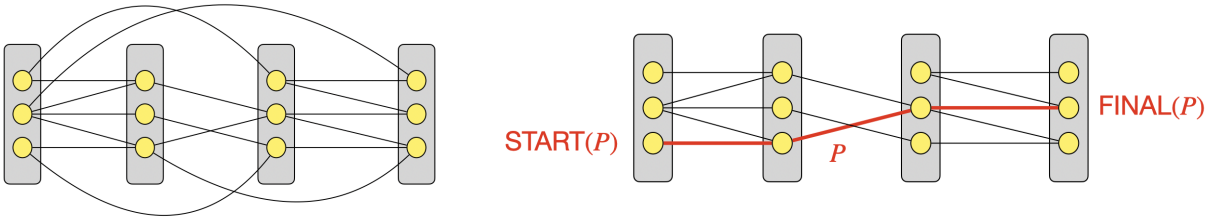
for some **absolute constant** $c_{\mathrm{comp}} \geq 1$ that we use from now on in our proofs.

5.4 Disjoint-Unique-Paths Graphs and Embedding Products

Our lower bound for MIS relies on two graph-theoretic ingredients: (*i*) an extremal family of graphs which can be seen as a generalization of *RS graphs* [RS78], and (*ii*) a graph product which we call the *embedding product* for this particular family of graphs. We now define these constructions and establish their key properties. To continue, we need the following basic definitions (see Figure 5.6).

Definition 5.4.1. *For any $n, k \geq 1$, a graph $G = (V, E)$ is called a (n, k)-**layered graph** if there exists an equipartition of V into k* independent sets *$(V_1, V_2, \ldots, V_k)$ of size n/k each, referred to as **layers** of G. We additionally say that G is a **strictly-layered graph** if all edges of G are between consecutive layers.*

Definition 5.4.2. *For any $n, k \geq 1$, a path P in an (n, k)-strictly-layered graph G is called a **layered path** if it has edges between consecutive layers and exactly one vertex per layer. For any layered path P, we define* $\textsc{start}(P)$ *as the vertex of P in the layer V_1 of G and* $\textsc{final}(P)$ *as the vertex of P in the layer V_k.*



(a) A $(12, 4)$-layered graph.

(b) A $(12, 4)$-strictly-layered graph and a layered path inside it.

Figure 5.6: An illustration of layered graphs and strictly-layered graphs with a layered path.

5.4.1 Disjoint-Unique-Paths (DUP) Graphs

The following definition captures the key concept we need from our extremal graphs.

Definition 5.4.3. *A set of paths $\mathcal{P}$ in a* strictly-*layered graph G is called a **unique path collection (UPC)** iff (i) $\mathcal{P}$ is a vertex-disjoint set of layered paths, and, (ii) for any pair of vertices s in $\textsc{start}(\mathcal{P}) := \{\textsc{start}(P) \mid P \in \mathcal{P}\}$ and t in $\textsc{final}(\mathcal{P}) := \{\textsc{final}(P) \mid P \in \mathcal{P}\}$, the only layered path between s and t in the entire graph G is a path in $\mathcal{P}$ (if one exists, s and t are the two ends of the same path $P \in \mathcal{P}$).*

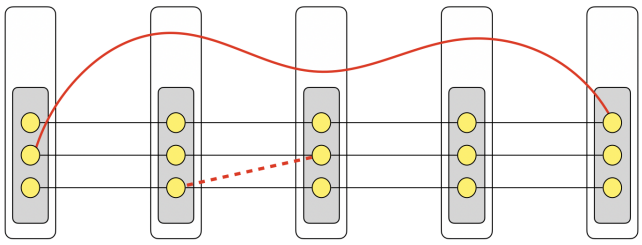


Figure 5.7: An illustration of a UPC in a strictly-layered graph. Neither the dashed red edges (between the vertices of different paths in a UPC) nor the edges of the solid red path (using outside vertices) are allowed.

See Figure 5.7 for an example of UPCs. We use UPCs to define our extremal graph family.

Definition 5.4.4. *For any $n, k, p, q \geq 1$, a (p, q, k)-**disjoint-unique-paths (DUP)** graph is a $(n, k+1)$-strictly-layered graph G whose edges can be partitioned into q UPCs $\mathcal{P}_1, \ldots, \mathcal{P}_q$, each consisting of exactly p layered paths $P_{i,1}, \ldots, P_{i,p} \in \mathcal{P}_i$ for $i \in [q]$.*

The following result shows the existence of (p, q, k)-DUP graphs with $p, q = n^{1-o(1)}$ (that is necessary for our application) and $k = \omega(\text{polylog}(n))$ (that is sufficient for our application). The existence of such results is a-priori surprising as UPCs of size p are "locally sparse" and restrict many edges from appearing in the graph, and yet large q implies that the graph is "globally dense". Nevertheless, we can use the same ideas as they are used in the construction of RS graphs [RS78] or for 'graphs with many long disjoint shortest paths' in [AB16] to construct DUP graphs.

Proposition 5.4.5. *For any sufficiently large $n \in \mathbb{N}$ and any given $k \geq 1$ satisfying $k \leq 2^{(\log n)^{1/4}}$, there exists a (p, q, k)-DUP graph on n vertices with the following parameters for two absolute constants $\eta_p, \eta_q > 0$,*

$$p = \frac{n}{\exp\left(\eta_p \cdot (\ln n)^{3/4}\right)} \qquad \text{and} \qquad q = \frac{n}{\exp\left(\eta_q \cdot (\ln n)^{3/4}\right)}.$$

Given that this result does not seem to have appeared in prior work, we provide a self-contained proof in Section 5.4.3, which may also provide some more intuition about these graphs.

5.4.2 Embedding Products

The next key ingredient of our construction is a graph product that treats DUP graphs as an "outer" graph and replace their *entire* paths in UPCs with different "inner" layered graphs (not necessarily strictly-layered ones). See Figure 5.8 for an illustration.

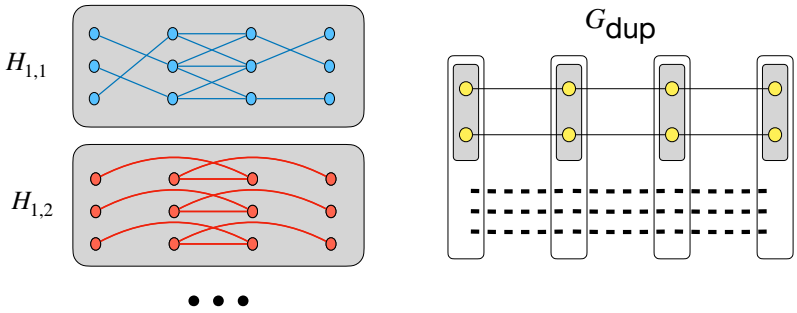
Definition 5.4.6. *Let G_{DUP} be a (p, q, k)-DUP graph on n_1 vertices and layers $U_1, \ldots, U_{k+1}$ for some $p, q, k \geq 1$. Let $\mathcal{H} := \{H_{i,j} \mid i \in [q], j \in [p]\}$ be a family of $(n_2, k+1)$-layered graphs where all the graphs in $\mathcal{H}$ are on the same layers $W_1, \ldots, W_{k+1}$ but may have different edges.*

*We define the **embedding of $\mathcal{H}$ into** G_{DUP}, denoted by $G := \text{EMBED}(\mathcal{H} \to G_{\text{DUP}})$, as the following $((n_1 \cdot n_2)/(k+1), k+1)$-layered graph:*

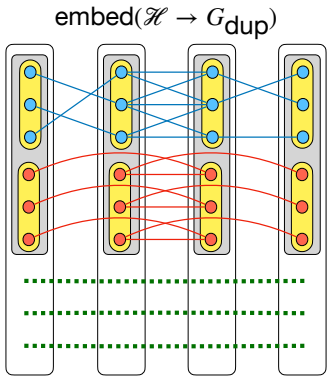
- ***Vertices:** We have layers $V_1, \ldots, V_{k+1}$ where, for every $\ell \in [k+1]$, $V_\ell := U_\ell \times W_\ell$;*
- ***Edges:** For any layered path $P_{i,j} = (u_1, u_2, ..., u_{k+1})$ of UPC $\mathcal{P}_i$ in G_{DUP} with $i \in [q]$ and $j \in [p]$, and any edge $(x, y) \in H_{i,j}$ between layers W_{ℓ_x} and W_{ℓ_y}, we add an edge e between (u_{ℓ_x}, x) and (u_{ℓ_y}, y) to G. We say that the edge e is added* w.r.t. *the path $P_{i,j}$.*

The following lemma captures the main property of these embeddings.

Lemma 5.4.7. *Let G_{DUP} be a (p, q, k)-DUP graph for some $p, q, k \geq 1$, $\mathcal{H} := \{H_{i,j} \mid i \in [q], j \in [p]\}$ be a family of layered graphs with size $p \cdot q$, and $G := \text{EMBED}(\mathcal{H} \to G_{\text{DUP}})$. For any $i \in [q]$, the* **induced subgraph** *of G on vertices corresponding to $\mathcal{P}_i$, i.e., $\{(v, *) \mid v \in P_{i,j}$ for some $j \in [p]\}$, is a* **vertex-disjoint union** *of graphs $H_{i,j}$ for $j \in [p]$ in $\mathcal{H}$.*



(a) A family $\mathcal{H}$ including $(12, 4)$-layered graphs $H_{1,1}$ and $H_{1,2}$ (remaining graphs are unspecified) on the left and a DUP graph G_{DUP} with $p = 2$, unspecified q, and $k = 3$ on the right.



(b) The resulting layered graph as the embedding of $\mathcal{H}$ into G_{DUP} – the drawing only shows the effect of the embedding on one UPC of G_{DUP} and the remainder of the graph is unspecified.

Figure 5.8: An illustration of the embedding product. For simplicity of exposition, this figure only shows a part of the final graph obtained via embedding inside a single UPC of the DUP graph; in the actual construction, every UPC is replaced by an embedded graph.

Before getting to the proof of Lemma 5.4.7, an important remark is in order. Recall that our layered graphs can have edges between any pairs of layers and not only consecutive ones. As such, to be able to establish the property in Lemma 5.4.7, we crucially rely on the fact that DUP graphs not only disallow any additional edges between vertices of a UPC, but in fact, no layered paths can connect them also even if the paths are using vertices outside the UPC. This is the key property of DUP graphs that is needed for our constructions.

Proof of Lemma 5.4.7. For any $i \in [q]$, let $U_i := \{(v, *) \mid v \in P_{i,j} \text{ for some } j \in [p]\}$ denote the vertices of G corresponding to the UPC $\mathcal{P}_i$. Let $G_i := G[U_i]$ be the induced subgraph of G on U_i. We first prove that (i) G_i contains the vertex-disjoint union of $H_{i,j}$ for $j \in [p]$, and then, that (ii) it does not contain any other edges. See Figure 5.9 for an illustration of this proof.

Part (i). Fix $j \in [p]$ and observe that all edges of $H_{i,j}$ are added to G_i by the embedding product w.r.t. layered path $P_{i,j}$ in a way that "preserves its structure". Formally, since $P_{i,j} = (u_1, \ldots, u_{k+1})$ is a *layered* path (with edges between consecutive layers and one vertex

per layer), the edges between layers W_ℓ and $W_{\ell'}$ in $H_{i,j}$ are added as edges between vertices $\{u_\ell\} \times W_\ell \subseteq V_\ell$ and $\{u_{\ell'}\} \times W_{\ell'} \subseteq V_{\ell'}$ of G_i. Also, since the layered paths of the UPC $\mathcal{P}_i$ must be vertex-disjoint (Definition 5.4.3), the edges added w.r.t. paths in $\mathcal{P}_i$ form a vertex-disjoint union of $H_{i,j}$'s for $j \in [p]$ in G_i.

Part (ii). It remains to argue that no other edges are added to G_i by the embedding product. For a contradiction, suppose that G_i contains an edge e that is *not* added w.r.t. $\mathcal{P}_i$. As such, it must be between vertices that correspond to $u \in P_{i,j_1}$ and $v \in P_{i,j_2}$ for some $j_1, j_2 \in [p]$ (possibly $j_1 = j_2$), respectively. By the embedding product, the edge e is added w.r.t. a layered path $P_{i',j'}$ and layered graph $H_{i',j'}$ for some $i' \neq i \in [q]$ and $j' \in [p]$. The layered graph $H_{i',j'}$ may contain any edge between layers even non-consecutive ones. Thus, the existence of edge e implies that there is a *subpath* of $P_{i',j'}$ that connects the vertices $u \in P_{i,j_1}$ and $v \in P_{i,j_2}$ in G_{DUP}, which is a strictly-layered graph and only has edges between consecutive layers. This creates a layered path from $\text{START}(P_{i,j_1})$ to $\text{FINAL}(P_{i,j_2})$ (or $\text{START}(P_{i,j_2})$ to $\text{FINAL}(P_{i,j_1})$) that uses one or more edges of $P_{i',j'}$ and is thus not in $\mathcal{P}_i$, which is a contradiction by the definition of a UPC (Definition 5.4.3). ∎

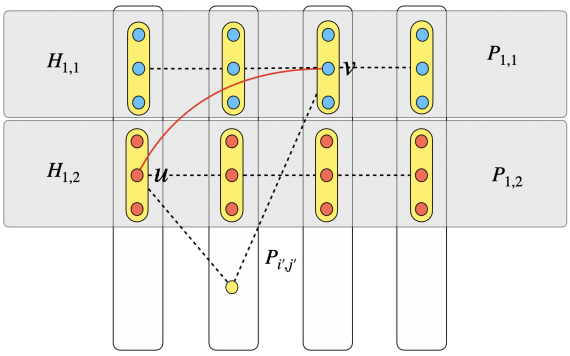


Figure 5.9: The gray boxes denote the subgraphs $H_{1,1}$ and $H_{1,2}$ that are created w.r.t. paths $P_{1,1}$ and $P_{1,2}$ in the same UPC of G_{DUP}. *If* there is an edge between vertices u and v that is added w.r.t. another path $P_{i',j'}$, it implies that some subpath of $P_{i',j'}$ connects two vertices in paths $P_{1,1}$ and $P_{1,2}$ together – this is because any edge of G is added "alongside" a path of G_{DUP}, although it is possible that it *shortcuts* a couple of layers in between as $H_{i',j'}$ may have edges between non-consecutive layers. This in turn creates another layered path between $\text{START}(P_{1,2})$ and $\text{FINAL}(P_{1,1})$, which is not allowed in a UPC, a contradiction.

5.4.3 A Construction of DUP Graphs

We now present a construction of DUP graphs and prove Proposition 5.4.5. The construction of these graphs follows a standard approach in the literature, e.g., in [AB16], for creating graphs with many long disjoint shortest paths. For completeness, we provide the construction to incorporate UPCs explicitly (beyond just bounding the number of "critical paths" as in [AB16]) and fine tune the parameters to the range that we need (which is different from the typical range of parameters used in other applications of these graphs that we are aware of).

We first need a basic claim on the existence of a large set of "average-free" *vectors*.

Claim 5.4.8. *For any $\ell, d \geq 1$, there exists a set $A \subseteq [\ell]^d$ with $|A| \geq \frac{\ell^d}{d \cdot \ell^2}$ such that for every multi-set of $t \geq 1$ not-all-equal vectors $y_1, \ldots, y_t \in A$, their average $(1/t)\sum_{i=1}^{t} y_i$ is not in A.*

Proof. Let A be the largest subset of $[\ell]^d$ such that all vectors in A have the same ℓ_2-norm. Since the number of different values possible for squared ℓ_2-norm of vectors in $[\ell]^d$ is at most $d \cdot \ell^2$, we obtain the lower bound in the claim statement on the size of A by the pigeonhole principle.

Let s denote the ℓ_2-norm of the vectors in A. Consider any multi-set of $t > 1$ not-all-equal vectors $y_1, \ldots, y_t \in A$. We have,

$$\|\frac{1}{t}\sum_{i=1}^{t} y_i\|_2^2 = \frac{1}{t^2}\left(\sum_{i=1}^{t}\|y_i\|^2 + \sum_{i\neq j}\langle y_i\ ,\ y_j\rangle\right) < \frac{1}{t^2}\left(\sum_{i=1}^{t}\|y_i\|^2 + \sum_{i\neq j}\|y_i\|\|y_j\|\right)$$

(the Cauchy-Schwartz inequality is strict for the pair $y_i \neq y_j$ which exists as y_i's are not all equal)

$$= \frac{1}{t^2}\left(t \cdot s^2 + t \cdot (t-1) \cdot s^2\right) = s^2.$$

The ℓ_2-norm of the average is strictly less than s, hence, it cannot be in A. ∎

We now use the existence of the set A in Claim 5.4.8 to construct our DUP graphs.

A construction of (p, q, k)-DUP graphs:

(*i*) Set $d := \sqrt{\log\left(\frac{n}{k+1}\right)}$ and $\ell := \left(\frac{n}{(k+1)\cdot(k+2)^d}\right)^{1/d}$ and let $A \subseteq [\ell]^d$ be the set of vectors in Claim 5.4.8.

(*ii*) Fix the vertex set $V = V_1 \cup V_2 \cup \ldots \cup V_{k+1}$ with $V_i := [(k+2)\cdot\ell]^d$ for each $i \in [k+1]$.

(*iii*) For every $x \in [\ell]^d$ and $y \in A$, and all $i \in [k]$, add an edge between

$$x + i \cdot y \in V_i \quad \text{and} \quad x + (i+1) \cdot y \in V_{i+1}.$$

Proof of Proposition 5.4.5. The partition of the edge set into $q = \ell^d$ UPCs of size $p = |A| \geq \ell^d/(d \cdot \ell^2)$ each is as follows. For every $x \in [\ell]^d$, we have a UPC $\mathcal{P}_x := \{P_{x,y} \mid y \in A\}$ of p paths. Each path $P_{x,y} \in \mathcal{P}_x$ for $y \in A$ is defined as:

$$P_{x,y} := (x + y\ ,\ x + 2y\ ,\ x + 3y\ ,\ \ldots\ ,\ x + (k+1) \cdot y),$$

where edge $(x+i\cdot y, x+(i+1)\cdot y)$ is from the layer V_i to layer V_{i+1} for $i \in [k]$. See Figure 5.10 for an illustration. We now prove that G is indeed a DUP graph with the UPCs $\mathcal{P}_x$ for $x \in [\ell]^d$.

Firstly, any edge of the graph G is between some $x + i \cdot y$ and $x + (i+1) \cdot y$ for $x \in [\ell]^d$ and $y \in A$ and $i \in [k]$. This edge thus belongs to the path $P_{x,y}$ in the UPC $\mathcal{P}_x$. Hence, the edges of the graph are partitioned between the UPCs.

Secondly, by the definition, each $P_{x,y}$ is a layered path. Furthermore, the paths in $\mathcal{P}_x$ are vertex-disjoint since, for any pair of paths $P_{x,y}$ and $P_{x,y'}$ for some $y, y' \in A$, their respective

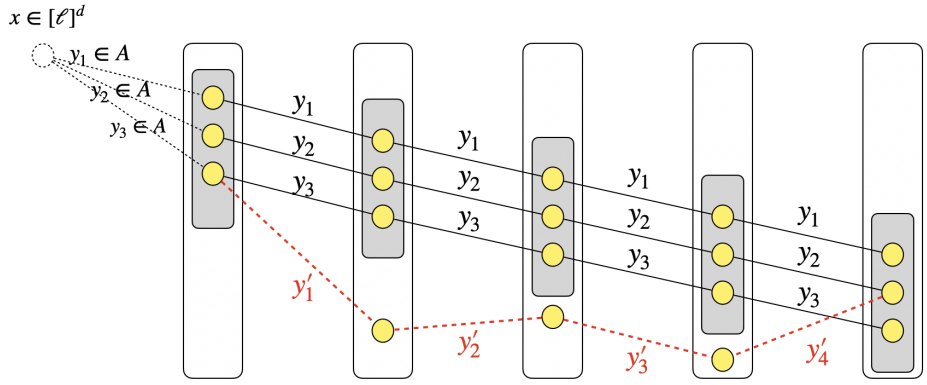


Figure 5.10: An illustration of the construction of DUP graphs. The circle x at left does not correspond to a vertex and instead is used to define a UPC. The red path cannot exist because it requires $y_3 + y'_1 + y'_2 + y'_3 + y'_4 = 5 \cdot y_2$ which cannot happen because A is average free.

vertices in layer i, given by $x + i \cdot y$ and $x + i \cdot y'$, are only the same if $y = y'$.

Finally, consider $z_s := \textsc{start}(P_{x,y})$ and $z_f := \textsc{final}(P_{x,y'})$ for some $y, y' \in A$; we prove that z_s has no layered path to z_f in G unless $y = y'$, in which case there is only one layered path $P_{x,y}$ (in the entire graph G), between z_s and z_f.

By definition, we have

$$z_s = x + y \qquad \text{and} \qquad z_f = x + (k+1) \cdot y'.$$

Suppose there exists a layered path between z_s and z_f. As each edge of the graph connects its both endpoints via a vector in A, this means that there exist vectors $y_1, \ldots, y_k \in A$ such that $z_f - z_s = \sum_{i=1}^{k} y_i$. This implies that

$$x + (k+1) \cdot y' = z_f = z_s + \sum_{i=1}^{k} y_i = x + y + \sum_{i=1}^{k} y_i,$$

which in turn means

$$y' = \frac{1}{k+1} \cdot \left(y + \sum_{i=1}^{k} y_i \right).$$

By Claim 5.4.8, since $y' \in A$, the only possibility is that $(y, y_1, \ldots, y_k)$ are all-equal, which means that they are also all equal to y'. Thus, the only possible layered path between z_s and z_t is if they belong to the same path $P_{x,y} \in \mathcal{P}_x$.

To finalize the proof, we can work out the parameters as follows. The number of vertices is

$$(k+1) \cdot ((k+2) \cdot \ell)^d = (k+1) \cdot \left((k+2) \cdot \left(\frac{n}{(k+1) \cdot (k+2)^d} \right)^{1/d} \right)^d = n,$$

as it should be, hence the choice of ℓ and d are consistent with n. Moreover, by the upper

bound on k and the choice of d, we have,

$$q = \ell^d = \left(\left(\frac{n}{(k+1)\cdot(k+2)^d}\right)^{1/d}\right)^d \geq \frac{n}{(k+2)^{d+1}} \geq \frac{n}{2^{\Theta(1)\cdot(\log n)^{1/4}\cdot(\log n)^{1/2}}} \geq \frac{n}{2^{\Theta((\log n)^{3/4})}};$$

$$p \geq \frac{\ell^d}{\ell^2\cdot d} = \frac{q}{\ell^2\cdot d} \geq \frac{q}{n^{2/d}\cdot\sqrt{\log n}} = \frac{q}{2^{2\log n/\sqrt{\log n/k}}\cdot\sqrt{\log n}} \geq \frac{q}{2^{\Theta(\sqrt{\log n})}} \geq \frac{n}{2^{\Theta((\log n)^{3/4})}},$$

as desired. This concludes the proof for all graph sizes determined this way from k and ℓ by picking η_p and η_q to match the hidden constants in the Θ-notation above.

To conclude the proof, we need the construction to work for all large enough integers n as in the statement of the proposition. However, this can be fixed easily using a padding argument, which we postpone to Section A.3. ∎

5.5 A Round vs Communication Tradeoff for MIS

We now switch to proving a multi-party communication lower bound for MIS, defined formally as:

Definition 5.5.1. *For any integers $n, t \geq 1$, we define $\mathbf{MIS}_{n,t}$ as the communication game of outputting* any *MIS of a given n-vertex graph whose edges are partitioned between t players.*

We prove an almost optimal round vs communication tradeoff for MIS in the model of Section 5.3.1, formalizing Result 1 from Section 5.1.

Theorem 9. *For any $r \geq 1$ and sufficiently large $n \in \mathbb{N}$, any r-round $(r+1)$-party protocol π for $\mathbf{MIS}_{n,r+1}$ with any constant probability of success strictly more than zero has communication cost*

$$CC(\pi) = \Omega\left(\frac{1}{2^{\Theta(r\log^{5/6}(n))}}\cdot n^{1+1/(2^r-1)}\right).$$

The lower bound for semi-streaming algorithms in Theorem 1, restated below, now follows.

Corollary 5.5.2. *For any integer $p \geq 1$ and sufficiently large $n \in \mathbb{N}$, any p-pass streaming algorithm for finding any maximal independent set with any constant probability of success strictly more than zero has space*

$$\Omega\left(\frac{1}{2^{\Theta(p\log^{5/6}(n))}}\cdot n^{1+1/(2^p-1)}\right).$$

In particular, semi-streaming algorithms require $\Omega(\log\log n)$ passes.

Proof. Follows from Theorem 9 and Proposition 5.3.2 since the number of players and rounds are $O(\log\log n)$ each, and hence, the space lower bound is at most smaller than the communication one by an $O((\log\log n)^2)$ factor; this is subsumed by the $2^{\Theta(p\log^{5/6}(n))}$ term of the lower bound. ∎

We define our hard input distribution in the proof of Theorem 9 in this section, and list its main properties. We then use this distribution in the next section to conclude the proof of Theorem 9.

5.5.1 A Family of Hard Distributions for MIS

The distributions are defined recursively as a family $\{\mathcal{G}_r(n)\}_{r\geq 0}$ where $\mathcal{G}_r(n)$ is a hard distribution for r-round $(r+1)$-party protocols on n-vertex graphs. To avoid ambiguity with the notation for paths, we use $Q_1, Q_2, \ldots, Q_{r+1}$ to denote the players in $\mathcal{G}_r(n)$. In addition, the input to player Q_a for $a \in [r+1]$ is denoted by G_a.

Base case for $r = 0$. The base case is defined vacuously for $r = 0$ and 0-round "protocols" that are supposed to solve a simple yet non-trivial MIS problem (and are hence trivially impossible).

The distribution $\mathcal{G}_0(n)$ for 0-round protocols on n-vertex graphs for $n \geq 2$.

1. Let $G = (V, E)$ be a graph on n vertices $V = \{u_i \mid i \in [p_0]\} \cup \{v_i \mid i \in [p_0]\}$ with $p_0 = n/2$.
2. For each $i \in [p_0]$, the edge (u_i, v_i) is independently included in E with probability $1/2$ (independent of the inclusion of the other potential edges (u_j, v_j), for $j \neq i$).
3. Player Q_1 receives $G(1) := G$ as the input (this is the only player when $r = 0$).

The answer to MIS on $\mathcal{G}_0$ contains either one of the vertices u_i *or* v_i (but not both) with probability half, and otherwise contains *both* u_i and v_i with the remaining probability for each $i \in [p_0]$ independently. Thus, a 0-round "protocol", namely, one that has to commit to a fixed answer always without checking the input, can only succeed with probability $1/2^{p_0}$ on $\mathcal{G}_0$.

Distributions for $r \geq 1$. A graph generated by $\mathcal{G}_r(n)$ will be an $(n, 2^{r+1})$-layered graph created by *embedding* many n_{r-1}-vertex hard instances for $(r-1)$ rounds into $(p_r, q_r, 2^r - 1)$-DUP graphs with b_r vertices in each layer. We will determine the values of the parameters p_r, q_r, b_r and n_{r-1} based on n shortly, after defining the distribution itself. Throughout, if clear from the context or not relevant, we may omit the subscripts on p_r, q_r, and b_r, as well as n from $\mathcal{G}_r(n)$.

Distribution $\mathcal{G}_r(n)$ for r-round protocols for $r \geq 1$.

1. Pick a fixed $(p_r, q_r, 2^r - 1)$-DUP graph $G_{\text{DUP}} := (W = W_1 \cup \ldots \cup W_{2^r}, E_{\text{DUP}})$ with b_r vertices in each layer and thus $b_r \cdot 2^r$ vertices in total.
2. Sample $p_r \cdot q_r$ independent instances $\mathcal{H} := \{H_{i,j} \mid i \in [q_r], j \in [p_r]\}$ from $\mathcal{G}_{r-1}(n_{r-1})$.
3. Create two identical copies of the graph $\widetilde{G}_L = \widetilde{G}_R := \text{EMBED}(\mathcal{H} \to G_{\text{DUP}})$.

4. For $a \in [r]$, player Q_a receives the edges in both $\widetilde{G}_L$ and $\widetilde{G}_R$ that correspond to $H_{i,j,a}$ for $i \in [q_r], j \in [p_r]$ where $H_{i,j,a}$ is the subgraph of $H_{i,j}$ given to the a^{th} player in $\mathcal{G}_{r-1}(n_{r-1})$.

5. Pick t uniformly at random from $[q_r]$ and let $\mathcal{P}^\star := \mathcal{P}_t$ be the t-th UPC in G_{DUP}, referred to as the *special UPC*. Let $W^\star \subseteq W$ be the set of vertices in G_{DUP} incident to $\mathcal{P}^\star$.

6. Player Q_{r+1} receives a complete bipartite graph between $\{(u, *) \mid u \in W \setminus W^\star\}$ in $\widetilde{G}_L$ and $\widetilde{G}_R$, i.e., there is an edge between every pair of vertices $(u_1, v_1) \in \widetilde{G}_L$ and $(u_2, v_2) \in \widetilde{G}_R$ with both u_1 and u_2 not belonging to $W^\star$.

Figure 5.11 gives an illustration of our hard distribution.

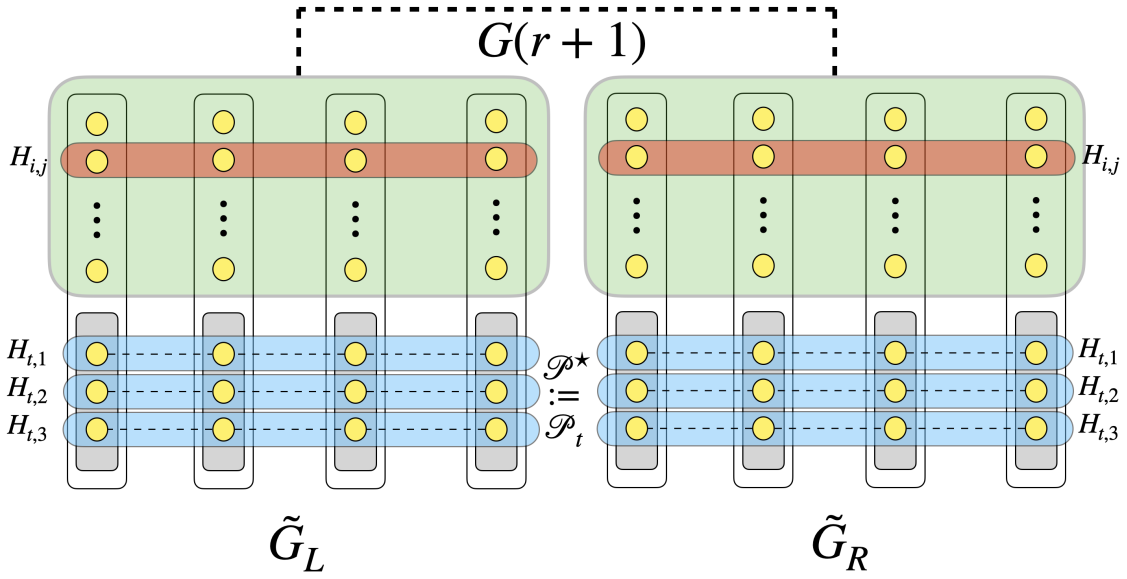


Figure 5.11: An illustration of the distribution $\mathcal{G}_r(n)$. The paths at the bottom correspond to the UPC $\mathcal{P}^\star$; each path is "replaced" with an instance H from $\mathcal{G}_{r-1}(n_{r-1})$ via the embedding product. Since the embedding product affects the entire graph, the paths in other UPCs are also replaced by instances from $\mathcal{G}_{r-1}(n_{r-1})$, e.g., for the top (red) block (remaining instances are not drawn). The last player Q_{r+1} receives a bipartite clique between the vertices corresponding to $W \setminus W^\star$ in $\widetilde{G}_L$ and $\widetilde{G}_R$, denoted by the top (green) blocks. Other players Q_a for $a \in [r]$ receive the inputs of the a^{th} player in *all* instances H sampled from $\mathcal{G}_{r-1}(n_{r-1})$ when placed inside graphs $\widetilde{G}_L$ and $\widetilde{G}_R$.

Definitions and Notation for Distribution $\mathcal{G}_r(n)$

We set up the following notation for the distribution $\mathcal{G}_r(n)$ for $r \geq 1$. For any $a \in [r+1]$, let $\mathcal{G}_{r,a}(n)$ denote the distribution of edges in the input graph that are given to player Q_a. Thus, the distribution $\mathcal{G}_r(n)$ is a joint distribution of $(\mathcal{G}_{r,1}(n), \ldots, \mathcal{G}_{r,r+1}(n))$.

An instance of the $\mathbf{MIS}_{n,r+1}$ sampled from $\mathcal{G}_r(n)$ is a graph G with n vertices together with its edge partitioning between the $r+1$ players. With a slight abuse of notation, we sometimes refer to G sampled from $\mathcal{G}_r(n)$, i.e., $G \sim \mathcal{G}_r(n)$, as the entire instance, where the edge partitioning between the players is implicit. Any $G \sim \mathcal{G}_r(n)$ contains $q_r \cdot p_r$ instances $H_{1,1}, \ldots, H_{q_r,p_r}$ sampled from $\mathcal{G}_{r-1}(n_{r-1})$ that are embedded as part of both $\widetilde{G}_L$ and $\widetilde{G}_R$. We refer to these $(r-1)$-round instances as the **sub-instances** of G. For any $i \in [q_r], j \in [p_r]$, we further write $\widetilde{E}_{L,i,j}$ and $\widetilde{E}_{R,i,j}$ to denote the edges inserted to, respectively, $\widetilde{G}_L$ and $\widetilde{G}_R$ as part of the embedding of $H_{i,j}$ into G_{DUP}.

Among the sub-instances of any $G \sim \mathcal{G}_r(n)$, there are p_r sub-instances that correspond to the paths in the special UPC $\mathcal{P}^\star$; we denote them by $\mathcal{H}^\star$ and call them the **special sub-instances** of G. We enumerate these special sub-instances by $H_{t,j}$ for $t \in [q_r]$ being the index of the special UPC $\mathcal{P}^\star = \mathcal{P}_t$ and j ranging over $[p_r]$.

For any $j \in [p_r]$, define the **special subgraphs** $G^\star_{L,j}$ and $G^\star_{R,j}$ as the subgraphs of $\widetilde{G}_L$ and $\widetilde{G}_R$ on edges of special sub-instance $H_{t,j}$, i.e., with edges $\widetilde{E}_{L,t,j}$ and $\widetilde{E}_{R,t,j}$, respectively.

The Choice of Parameters and Their Ranges

We now specify how the parameters p_r, q_r, b_r, and n_{r-1} are chosen, the restriction we have on their ranges, and their "validity" in their recursive calls (and Proposition 5.4.5).

Parameters b_r, n_{r-1}. Let n_0 be an arbitrarily large constant to be determined later (this controls the probability of success of the protocols). We work with large $n \gg n_0$ for all $r \geq 1$.

The distribution $\mathcal{G}_r(n)$ for $r \geq 0$ rounds requires that

$$n \geq (1/2) \cdot (2n_0)^{2^{r-1}}. \tag{5.3}$$

For $r = 1$, n_0 is already defined as a constant and b_1 is defined from n as

$$b_1 = \left(\frac{n}{2 \cdot n_0}\right). \tag{5.4}$$

For $r > 1$, b_r and n_{r-1} are defined from n via the following equations (recall that $(a) \uparrow (b) = a^b$)

$$\begin{aligned} b_r &= (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right), \\ 2 \cdot b_r \cdot n_{r-1} &= n, \end{aligned} \tag{5.5}$$

which implies that

$$\begin{aligned} n_{r-1} &= \left(\frac{n}{2}\right) \uparrow \left(\frac{2^{r-1}-1}{2^r-1}\right) \\ b_r &= \left(\frac{n}{2}\right) \uparrow \left(\frac{2^{r-1}}{2^r-1}\right). \end{aligned} \tag{5.6}$$

In the distribution $\mathcal{G}_r(n)$ for any $r \geq 1$, we call the distribution $\mathcal{G}_{r-1}(n_{r-1})$. We will show that when input n satisfies Eq (5.3) in $\mathcal{G}_r(n)$, calls to $\mathcal{G}_{r-1}(n_{r-1})$ also satisfy Eq (5.3).
For $r = 1$, $\mathcal{G}_1(n)$ calls $\mathcal{G}_0(n_0)$ which satisfies Eq (5.3) for $r = 0$ as

$$(1/2) \cdot (2n_0)^{2^0-1} = (1/2), \qquad \text{and} \qquad n_0 \geq 1.$$

For $r > 1$, $\mathcal{G}_r(n)$ calls $\mathcal{G}_{r-1}(n_{r-1})$ and by the choice of n_{r-1} in Eq (5.6), we have,

$$n_{r-1} = \left(\frac{n}{2}\right) \uparrow \left(\frac{2^{r-1}-1}{2^r-1}\right) \underset{\text{Eq (5.3)}}{\geq} \left(\frac{1}{4} \cdot (2n_0)^{2^r-1}\right) \uparrow \left(\frac{2^{r-1}-1}{2^r-1}\right) \geq \frac{1}{2} \cdot (2n_0)^{2^{r-1}-1}, \quad (5.7)$$

which means the call to $\mathcal{G}_{r-1}(n_{r-1})$ also satisfies Eq (5.3) for $r-1$. This gives us the following.

Observation 5.5.3. *For any $r \geq 1$, if the distribution $\mathcal{G}_r(n)$ satisfies Eq (5.3) for n, then any distribution $\mathcal{G}_{r'}(n')$ for $r' < r$ called recursively in $\mathcal{G}_r$ also satisfies Eq (5.3) for r', n'.*

Parameters p_r, q_r for $r \geq 1$. Recall that we use a $(p_r, q_r, 2^r - 1)$-DUP graph with b_r vertices in each layer in $\mathcal{G}_r$. This means the the total number of vertices in this DUP graph is $b_r \cdot 2^r$. We define the parameters p_r and q_r as follows:

$$\begin{aligned} p_r &:= \frac{b_r \cdot 2^r}{\exp(\eta_p \cdot (\ln(b_r \cdot 2^r))^{3/4})}, \\ q_r &:= \frac{b_r \cdot 2^r}{\exp(\eta_q \cdot (\ln(b_r \cdot 2^r))^{3/4})}. \end{aligned} \quad (5.8)$$

Firstly, these bounds match those of Proposition 5.4.5 and thus we can apply the proposition to prove the existence of the required DUP graphs in our distribution. We should also verify that

$$2^r - 1 \leq 2^{(\log(b_r \cdot 2^r))^{1/4}},$$

which is required by Proposition 5.4.5. For $r = 1$, the LHS is 1 and this holds trivially. For $r > 1$, we have by Eq (5.6) that

$$b_r = \left(\frac{n}{2}\right) \uparrow \left(\frac{2^{r-1}}{2^r-1}\right) \underset{\text{Eq (5.3)}}{\geq} \left(\frac{1}{4} \cdot (2n_0)^{2^r-1}\right) \uparrow \left(\frac{2^{r-1}}{2^r-1}\right) \geq \frac{1}{4} \cdot \left((2n_0)^{2^{r-1}}\right).$$

This satisfies the above equation because for $r \geq 2$,

$$\log(2^r - 1) \leq r \qquad \text{whereas} \qquad (\log(b_r \cdot 2^r))^{1/4} \geq (2^{r-1} \cdot \log(n_0))^{1/4}$$

and thus we obtain the equation for any large constant n_0 (with quite some room to spare).

Observation 5.5.4. *For any $r \geq 1$, if the distribution $\mathcal{G}_r(n)$ satisfies Eq (5.3) for n, then the DUP graphs created in $\mathcal{G}_r$ (and in any recursive call to $\mathcal{G}_{r'}$ for $r' < r$) with the given parameters p_r and q_r do exist (by Proposition 5.4.5).*

5.5.2 Basic Properties of the Distributions $\{\mathcal{G}_r(n)\}_{r\geq 1}$

We establish some basic properties of the distribution $\mathcal{G}_r(n)$ for $r \geq 1$ in this subsection. We then use these to define *search predicates* that reduce our task of proving the lower bound for MIS to determining the ability of protocols in figuring out a certain predicate for graphs sampled from $\mathcal{G}_r$.

The first property identifies the "real input" to the players.

Property 5.5.5. *For any $a \in [r]$, the input of player Q_a in $G_a \sim \mathcal{G}_{r,a}(n)$ is determined deterministically by the input of the a^{th} player in all sub-instances in $\mathcal{H}$, i.e.,*

$$\mathcal{H}_a := \{H_{i,j,a} \mid i \in [q_r], j \in [p_r]\}$$

. The input of player Q_{r+1} is deterministically determined by the index $t \in [q_r]$ of the special UPC.

Proof. The choice of G_{DUP} is fixed in $\mathcal{G}_r(n)$ and does not include any randomness (think of it as being "hardcoded" in the definition of the distribution). For any $a \in [r]$, given only $\mathcal{H}_a$, player Q_a can know the edges of $\widetilde{G}_L$ and $\widetilde{G}_R$ that belong to its input. Similarly, given only t, player Q_{r+1} can know what vertices of $\widetilde{G}_L$ and $\widetilde{G}_R$ are in $W^\star$ and thus which edges should be added to the bipartite clique between vertices corresponding to $W \setminus W^\star$ in $\widetilde{G}_L$ and $\widetilde{G}_R$. ∎

The next property shows that the distributions $\{\mathcal{G}_r(n)\}_{r\geq 0}$ are product distributions – a fact which is used crucially in our lower bound arguments.

Property 5.5.6. *For any $r \geq 0$, the distribution $\mathcal{G}_r(n)$ is a product distribution, i.e.,*

$$\mathcal{G}_r(n) = \mathcal{G}_{r,1}(n) \times \mathcal{G}_{r,2}(n) \times \cdots \times \mathcal{G}_{r,r+1}(n).$$

Proof. The proof is by induction on r. The base case when $r = 0$ is trivial as the entire graph is given to Q_1. Let us assume that the statement is true for $r = \ell$ and we prove it for $r = \ell + 1$.

The input to players in $\mathcal{G}_{\ell+1}(n)$, by Property (5.5.5) is determined by $q_{\ell+1} \cdot p_{\ell+1}$ sub-instances sampled from $\mathcal{G}_\ell(n_\ell)$ (for players Q_1 to $Q_{\ell+1}$) and by $t \in [q_{\ell+1}]$ (for player $Q_{\ell+2}$). The distribution of each sub-instance $H_{i,j}$ for $i \in [q_{\ell+1}]$ and $j \in [p_{\ell+1}]$ is a product distribution of the inputs to Q_a for all $a \in [\ell + 1]$ by the induction hypothesis. These sub-instances are also all sampled independently of each other, thus making the collective input to players Q_1 to $Q_{\ell+1}$ independent of each other. Finally, the index $t \in [q_{\ell+1}]$ is sampled independent of all other variables, implying the input to player $Q_{\ell+2}$ is also independent of the rest, completing the proof. ∎

The next two properties together identify the key properties of special subgraphs and their role w.r.t. any MIS of the input graph.

Property 5.5.7. *The special subgraphs $G^\star_{L,j}$ and $G^\star_{R,j}$ for $j \in [p_r]$ are all vertex-disjoint from each other. Moreover, the induced subgraph of $\widetilde{G}_L$ (resp. $\widetilde{G}_R$) on the vertices corresponding*

to the special subgraphs contains only the edges of these subgraphs, i.e., the edges of $G^{\star}_{L,j}$ *(resp.* $G^{\star}_{R,j}$*) for all* $j \in [p_r]$.

Proof. The special subgraphs inserted to either $\widetilde{G}_L$ or $\widetilde{G}_R$ are part of the embedding product of $\textsc{embed}(\mathcal{H} \to G_{\text{DUP}})$ corresponding to the special UPC $\mathcal{P}^{\star}$. Thus, by Lemma 5.4.7, we know that they are vertex-disjoint and no other edges from any other subgraphs are present in the vertices of the special subgraphs. ∎

Property 5.5.8. *For any graph* $G \sim \mathcal{G}_r(n)$ *for* $r \geq 1$*, any MIS* S *of* G *also contains an MIS for every special subgraph* $G^{\star}_{L,1}, \ldots, G^{\star}_{L,p_r}$ *in* $\widetilde{G}_L$ *or every special subgraph* $G^{\star}_{R,1}, \ldots, G^{\star}_{R,p_r}$ *in* $\widetilde{G}_R$.

Proof. We consider three possible cases for the proof.

Case 1. Suppose first that there exists a vertex $v \in S$ which (1) belongs to $\widetilde{G}_L$ and (2) corresponds to some vertex in $W \setminus W^{\star}$. Figure 5.12 gives an illustration of this case.

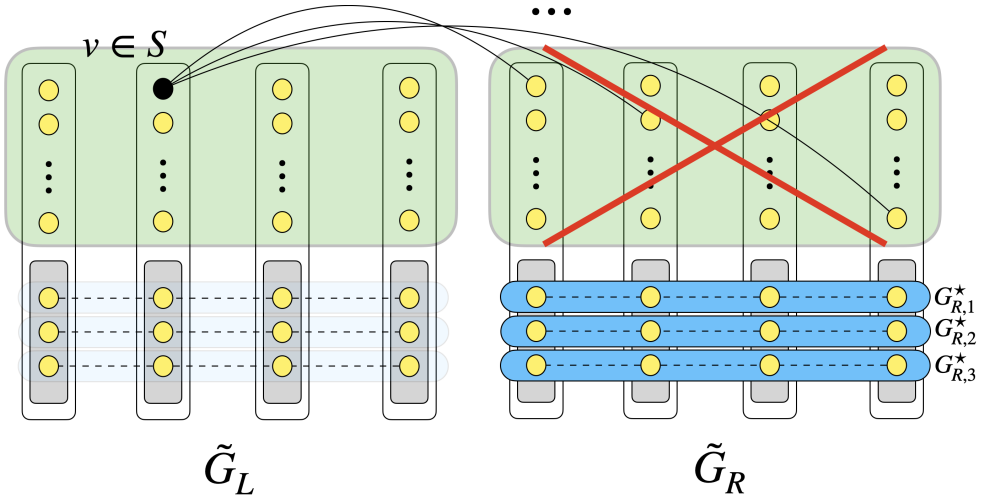


Figure 5.12: An illustration of case 1: picking vertex v in the MIS "removes" all vertices on $\widetilde{G}_L$ that are not part of the special subgraphs as the remaining ones are all neighbor to v (not all edges of v are drawn). Since the special subgraphs are now isolated, the MIS of G now needs to contain a separate MIS for each of the special subgraphs in $\widetilde{G}_R$ in bottom right (drawn in blue).

Since $G_{r+1} \subseteq G$ is a bipartite clique between vertices corresponding to $W \setminus W^{\star}$ in $\widetilde{G}_L$ and $\widetilde{G}_R$, we have that all vertices corresponding to $W \setminus W^{\star}$ in $\widetilde{G}_R$ are now incident on v and cannot be part of the MIS S. The remaining vertices in $\widetilde{G}_R$ are incident on $W^{\star}$ and thus S also needs to contain an MIS of the induced subgraph of $\widetilde{G}_R$ on vertices corresponding to $W^{\star}$ or equivalently $\mathcal{P}^{\star}$ – this is because these vertices do not have any edge to $\widetilde{G}_L$ and also no vertex of $\widetilde{G}_R$ outside these can be part of S.

We can now apply the main property of the embedding product, i.e., Lemma 5.4.7, captured in Property (5.5.7) to have that the induced subgraph of $\widetilde{G}_R$ on the vertices corresponding to the UPC $\mathcal{P}^{\star}$ is a vertex-disjoint union of subgraphs $G^{\star}_{R,1}, \ldots, G^{\star}_{R,p_r}$. This

immediately implies that the MIS of this subgraph of $\widetilde{G}_R$ should be a union of the MISes of $G^\star_{R,1}, \ldots, G^\star_{R,p_r}$, which means the MIS S contains an MIS for each of these special subgraphs.

Case 2. Suppose the symmetric case that there exists a vertex $v \in S$ which (1) belongs to $\widetilde{G}_R$ and (2) corresponds to some vertex in $W \setminus W^\star$. The same exact argument implies that now S should contain an MIS for every special subgraph $G^\star_{L,1}, \ldots, G^\star_{L,p_r}$ instead.

Case 3. Finally, suppose that no vertex in S corresponds to a vertex of $W \setminus W^\star$ in either of $\widetilde{G}_L$ or $\widetilde{G}_R$. This, similar to the above, implies that S should contain an MIS for the induced subgraph of $\widetilde{G}_L$ on vertices corresponding to $W^\star$ *and* the induced subgraph of $\widetilde{G}_R$ on vertices corresponding to $W^\star$. This in turn, again, as above, implies that S now contains an MIS for every special subgraph $G^\star_{L,1}, \ldots, G^\star_{L,p_r}$ *and* every $G^\star_{R,1}, \ldots, G^\star_{R,p_r}$. ∎

5.5.3 Search Sequences and Predicates

Before getting to analyze the distributions $\{\mathcal{G}_r(n)\}_{r\geq 1}$, we need one other set of definitions, which capture the notion of *hierarchical embeddings* in our lower bounds.

At a high level, the analysis goes as follows. We are *hiding* the choice of the special UPC of the instance from the first r players in the first round, then UPC of special sub-instances from the first $r-1$ players in the second round, and so on and so forth, until at the end of the last round; at that point, we reach the "inner most" graphs sampled from $\mathcal{G}_0$, whose edges are still hidden from the players, and now there is no more round to compute the answer. Making this intuition precise is going to take work, but hopefully this provides some insight for the following definitions.

Notation. Fix any $r \geq 1$. We use p_j, q_j, n_j for $0 \leq j < r$ to denote the parameters p, q, n in the instances sampled from the hard distribution for j rounds, i.e., $\mathcal{G}_j$, in the construction of the hard distribution on r rounds with n vertices, i.e., $\mathcal{G}_r(n)$.

Definition 5.5.9. *For $r \geq 0$, we define $K = (k_r, k_{r-1}, \ldots, k_1)$ to be a valid **search sequence** if $k_i \in [p_i]$ for each $i \in [r]$ ($K = \emptyset$ is the only valid search sequence for $r = 0$). We interpret a search sequence on a graph $G = G_r \sim \mathcal{G}_r(n)$ as:*

- *k_r points to the k_r^{th} special sub-instance of G_r, namely, $G_{r-1} := H_{t_r,k_r} \sim \mathcal{G}_{r-1}$ where t_r is the index of the special UPC of G_r;*
- *k_{r-1} points to the k_{r-1}^{th} special sub-instance of G_{r-1}, namely, $G_{r-2} := H_{t_{r-1},k_{r-1}} \sim \mathcal{G}_{r-2}$ where t_{r-1} is the index of the special UPC of G_{r-1};*
- *We continue like this until k_1 points to the k_1^{th} special sub-instance of G_1, namely, $G_0 := H_{t_1,k_1} \sim \mathcal{G}_0$ where t_1 is the index of the special UPC of G_1.*
- *We now have a unique instance G_0 of $\mathcal{G}_0$ defined by K.*

*Finally, we define the **search predicate** of a $G \sim \mathcal{G}_r(n)$ and a search sequence K as:*

$$\mathbb{P}_r(G, K) := \text{a string in } \{0,1\}^{p_0} \text{ where the } i\text{-th bit is 1 iff } (u_i, v_i) \text{ is an edge in } G_0.$$

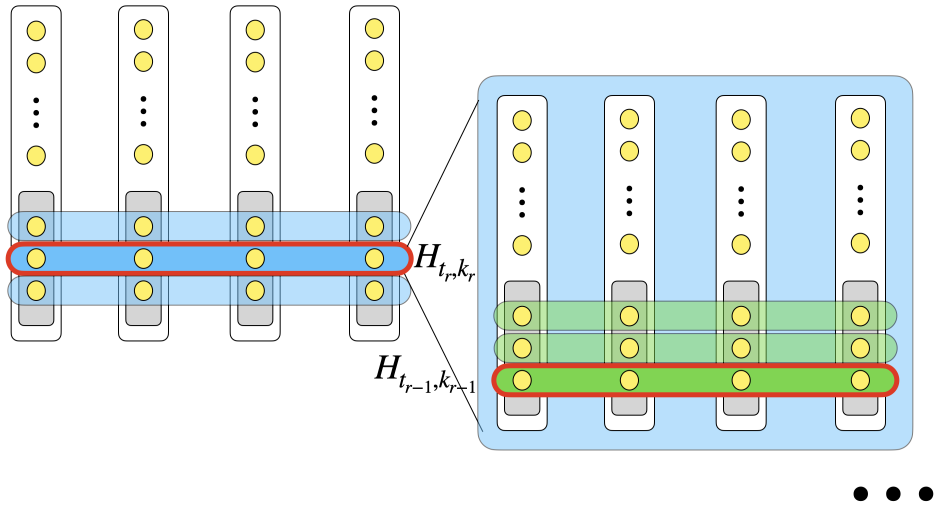


Figure 5.13: An illustration of a search sequence. Here, k_r picks a special sub-instance H_{t_r,k_r} from $\mathcal{G}_{r-1}$ (with t_r being the index of the special UPC of the outer instance), and then "inside" that instance, k_{r-1} picks another sub-instance $H_{t_{r-1},k_{r-1}}$ from $\mathcal{G}_{r-2}$ (with t_{r-1} being the index of the special UPC of the instance H_{t_r,k_r}), and this continues on.

Figure 5.13 gives an illustration of this definition.

Let π be any $(r+1)$-party communication protocol on inputs sampled from $\mathcal{G}_r(n)$, and let $\Pi = \Pi(G)$ be the transcript of π at the end of r rounds on an input $G \sim \mathcal{G}_r(n)$. We say that the protocol π **solves the search predicate** $\mathbb{P}_r$ for input graph G, if only given the transcript Π (with no further access to G), $\mathbb{P}_r(G, K)$ can be deterministically determined for *all* valid search sequences K. We shall emphasize that the search sequence K is *not* a part of the input to the protocol π (and is in fact even independent of the input graph G).

The following lemma shows why search sequences are important to us.

Lemma 5.5.10. *For every $\delta \in (0,1)$, any $(r+1)$-party protocol π that outputs an MIS of $G \sim \mathcal{G}_r(n)$ with probability at least δ, also solves the search predicate $\mathbb{P}_r(G,K)$ for* every *valid search sequence K with probability at least δ.*

Proof. We claim that:

> For any $r \geq 0$, any MIS of $G \sim \mathcal{G}_r(n)$ uniquely determines $\mathbb{P}_r(G,K)$ for all valid search sequences K.

We can then conclude the proof as follows. Protocol π will be computing some MIS of G with probability at least δ. Thus, whenever π succeeds, the transcript Π contains an MIS that can be used to solve the predicate $\mathbb{P}_r(G,K)$ for all K with probability at least δ as well.

We now prove the above statement by induction on r. The base case when $r = 0$ is a graph G_0 consisting of $n_0 = 2p_0$ vertices u_i and v_i for each $i \in [p_0]$ and $K = \emptyset$. Each such u_i and v_i pair for $i \in [p_0]$ is possibly an edge in G_0. If the edge does exist, any MIS necessarily should have exactly one of u_i or v_i but not both, whereas if the edge does not exist, then any MIS of G_0 should contain both u_i and v_i. Thus, given any MIS of $G_0 \sim \mathcal{G}_0$, we can determine the predicate $\mathbb{P}_0(G, \emptyset)$.

For the induction step, suppose the statement is true for $r-1$ and we prove it for r. Let $K := (k_r, \ldots, k_1)$ be any search sequence. By Property (5.5.8), any MIS of $G \sim \mathcal{G}_r$ also contains an MIS either for *every* special subgraph $G^\star_{L,1}, \ldots, G^\star_{L,p_r}$ or for *every* special subgraph $G^\star_{R,1}, \ldots, G^\star_{R,p_r}$. In particular, it contains an MIS for either $G^\star_{L,k_r}$ or $G^\star_{R,k_r}$. Both these graphs are identical and correspond to the k_r^{th} special sub-instance H_{t,k_r} in G where t is the index of the special UPC. As $H_{t,k_r} \sim \mathcal{G}_{r-1}$, by the induction hypothesis, an MIS of H_{t,k_r} uniquely determines

$$\mathbb{P}_{r-1}(H_{t,k_r}, (k_{r-1}, k_{r-2}, \ldots, k_1)).$$

But, by definition, we also have

$$\mathbb{P}_r(G, K) = \mathbb{P}_{r-1}(H_{t,k_r}, (k_{r-1}, k_{r-2}, \ldots, k_1)),$$

as $k_r \in K$ is just pointing to the special sub-instance H_{t,k_r}. This finalizes the proof. ∎

5.6 New Round Elimination

We are given a family of problems $\{\mathbb{F}_r\}_{r\geq 0}$ such that $\mathbb{F}_r$ is a communication complexity problem with $(r+1)$ players, $Q_1, Q_2, \ldots, Q_{r+1}$, and the players are restricted to r-rounds. We prove the round elimination lemma in this section for this general family of problems.

For $r \geq 1$, we are given a distribution $\mathcal{D}_r$ over hard instances of $\mathbb{F}_r$ for r-round protocols. The problem and the distribution have the following properties:

Properties of $\mathbb{F}_r$:

- (i) In problem $\mathbb{F}_{r-1}$, the players Q_ℓ for $\ell \in [r]$ have a source of public randomness. Each player Q_ℓ for $\ell \in [r]$ also has a source of private randomness.
- (ii) The distribution $\mathcal{D}_r, \mathcal{D}_{r-1}$ are product distributions over the $(r+1)$ players and r players respectively, i.e., the input of Q_ℓ can be sampled independently of the input of $Q_{\ell'}$ for all $\ell \neq \ell'$, $\ell, \ell' \in [(r+1)]$, and similarly for $\mathcal{D}_{r-1}$.
- (iii) The distribution $\mathcal{D}_r$ is constructed by sampling $q_r \cdot p_r$ instances $B_{i,j}$ for $i \in [q_r], j \in [p_r]$ independently from $\mathcal{D}_{r-1}$, such that Q_ℓ for $\ell \in [r]$ receives the input that they would receive in $B_{i,j}$ for all $i \in [q_r], j \in [p_r]$.
- (iv) Player Q_{r+1} has a special index $t \in [q_r]$, and this is the only input of Q_{r+1}.
- (v) If any r-round protocol succeeds on instance $A_r \sim \mathcal{D}_r$, the transcript of the protocol contains answers to at least ρ-fraction of $B_{t,j}$ for $j \in [p_r]$ for some $\rho \in [0,1]$.[a]

[a] In the context of MIS, the transcript in fact contains answers to all $B_{t,j}$ for $j \in [p_r]$. Our round elimination lemma holds even in the weaker case when only a large fraction of $B_{t,j}$ are solved in a successful transcript.

Then, we prove the following theorem:

Theorem 10. *For every $0 < \varepsilon < \delta < 1$, parameter $\rho \in [0,1]$, and integer $s \geq 1$, the following is true. Suppose there is a deterministic r-round $(r+1)$-party protocol π_r with communication cost $CC(\pi_r) \leq s$ that solves problem $\mathbb{F}_r$ with probability at least δ for an instance $A_r \sim \mathcal{D}_r$. Then, for constant c_{comp} (from Eq (5.2)), there is an $(r-1)$-round r-party deterministic protocol π_{r-1} with communication cost*

$$CC(\pi_{r-1}) \leq \frac{c_{\text{comp}}}{\varepsilon} \cdot (\frac{s}{p_r} + r^2)$$

that solves problem $\mathbb{F}_{r-1}$ for any instance $A_{r-1} \sim \mathcal{D}_{r-1}$ with probability of success at least

$$\delta \cdot \rho - \varepsilon - \sqrt{\frac{s}{2p_r \cdot q_r}}.$$

This section is dedicated to proving Theorem 10.

5.6.1 The Setup for the Proof of Theorem 10

We now start the proof of Theorem 10 which is the most technical part of the paper. Fix any $r \geq 1$ and let π_r be a r-round $(r+1)$-party protocol for solving $\mathbb{F}_r$ with the parameters specified in Theorem 10. We further define (or recall) the following notation:

- Let $\mathcal{B}$ denote the family of smaller instances $B_{i,j}$ for $i \in [q_r], j \in [p_r]$.
- $\mathcal{D}_{r,\ell}$ for $\ell \in [r+1]$ to denote the distribution of the input instance A_r given to player Q_ℓ; we further use B_ℓ to denote the input of Q_ℓ in a specific sub-instance $B \in \mathcal{B}(A)$.
- $\mathtt{t}$ to denote the random variable for the index $t \in [q_r]$ of the special row of instances picked in $\mathcal{D}_r$.
- $B_{i,*}$ for any $i \in [q_r]$ to denote the sub-instances $(B_{i,1}, B_{i,2}, \ldots, B_{i,p_r})$ together in $A \sim \mathcal{D}_r$. The sub-instances given to player Q_ℓ for $\ell \in [r+1]$ are denoted by $B_{i,*,\ell} = (B_{i,1,\ell}, B_{i,2,\ell}, \ldots, B_{i,p_r,\ell})$.
- $\Pi = (\Pi_1, \ldots, \Pi_r)$ as the set of messages communicated by the players in rounds 1 to r. Similarly, we use $\Pi_{i,\ell}$ for $\ell \in [r+1]$ to denote the messages of player Q_ℓ in round $i \in [r]$, and $\Pi_{*,\ell}$ to denote all messages of Q_ℓ. We use $\mathsf{\Pi}, \mathsf{\Pi}_i$, and $\mathsf{\Pi}_{i,\ell}$ as the corresponding random variables for these messages.

Our goal is to go from a r-round $(r+1)$-party protocol π_r for $\mathbb{F}_r$ with communication cost s and probability of success δ, to a $(r-1)$-round r-party protocol π_{r-1} for $\mathbb{F}_{r-1}$ with communication cost $\approx s/(\varepsilon \cdot p_r)$ and success probability $\approx \delta \cdot \rho - \varepsilon - \sqrt{s/(p_r \cdot q_r)}$. We do this in two steps:

- **Step 1:** We first *shave off* one player from π_r and obtain an intermediate r-party protocol π' with communication cost $\approx s/(\varepsilon \cdot p_r)$ and success probability $\approx \delta \cdot \rho - \varepsilon$ for $\mathbb{F}_{r-1}$. However, π' still has r rounds instead of our desired $(r-1)$ rounds.

- **Step 2:** We then *shave off* one round from π' and obtain the protocol π_{r-1} without increasing the communication but by decreasing the success probability with another $\approx \sqrt{s/(p_r \cdot q_r)}$ term.

We implement each step in the following two subsections. We do emphasize that these steps are not entirely blackbox and this partitioning into the two steps is more for the simplicity of exposition (and there will be some intermediate steps as well). Figure 5.14 gives a schematic organization of these steps, in particular, the protocols we build along the way and their properties.

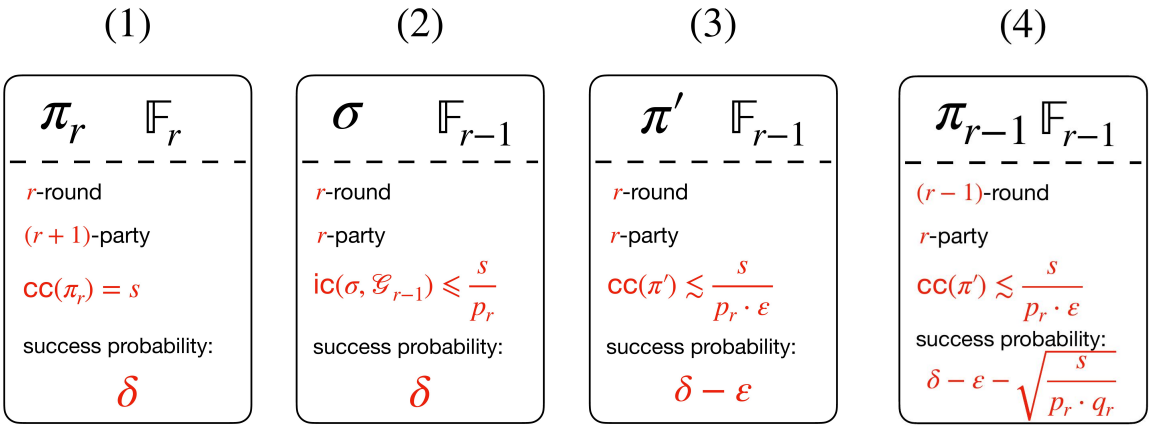


Figure 5.14: A schematic organization of the proof of Theorem 10: step 1 encompasses moving from part 1 to part 3 here, and step 2 is for moving from part 3 to part 4.

5.6.2 Step 1: A Low Communication Protocol for $\mathbb{F}_{r-1}$ in r Rounds

Fix $r \geq 1$, suppose we have an instance A_{r-1} sampled from $\mathcal{D}_{r-1}$, and we want to use π_r, which is for $\mathcal{D}_r$, to solve $\mathbb{F}_{r-1}$. Consider the following direct way of doing this:

An r-round r-party protocol σ for $\mathbb{F}_{r-1}$ on input instance A_{r-1}:

1. Sample indices $j^\star \in [p_r]$ and $t \in [q_r]$ uniformly at random using public randomness.
2. For every $\ell \in [r]$, each player Q_ℓ samples $\left(A_{r,\ell} \sim \mathcal{D}_r \mid \mathsf{t} = t, \mathsf{B}_{t,j^\star,\ell} = B_{r-1,\ell}\right)$ using private randomness. Let $A_r = (A_{r,1}, A_{r,2}, \ldots, A_{r,r+1})$ be the input of all players in $\mathcal{D}_r$ where, by item (iv), the input of the "simulated" player Q_{r+1} is fixed by $\mathsf{t} = t$.
3. The players run $\pi_r(A_r)$ by simulating the messages of player Q_{r+1} in π_r given that t is public knowledge.
4. At the end, to solve $\mathbb{F}_{r-1}(A_{r-1})$, we use the answer of π_r for $\mathbb{F}_r(A_r)$. By item (v), we know that this transcript contains the answer to $B_{t,j^\star} = A_{r-1}$ when $j^\star \in [p_r]$ is randomly chosen, with probability ρ.

Define the **distribution** $\mathcal{D}^{\text{REAL}}$ as the distribution of the instance A_r obtained in the protocol σ when $A_{r-1} \sim \mathcal{D}_{r-1}$. We claim that this is the "right" distribution.

Observation 5.6.1. *Distribution $\mathcal{D}^{\text{REAL}}$ is the same as $\mathcal{D}_r$.*

Proof. We know from item *(ii)* that $\mathcal{D}_r$ is a product distribution of $\mathcal{D}_{r,1} \times \ldots \mathcal{D}_{r,r+1}$. Moreover, each $\mathcal{D}_{r,i}$ for $i \in [r]$ is, by item *(iii)*, a collection of *independent* instances $B_{i,j,a} \sim \mathcal{D}_{r-1,a}$ and $t \in [q_r]$ is chosen uniformly at random in $\mathcal{D}_r$ to define the input of player Q_{r+1}. This matches exactly the distribution of $\mathcal{D}^{\text{REAL}}$ in σ, given that $B_{r-1,\ell}$ used by player Q_ℓ is sampled from $\mathcal{D}_{r-1,\ell}$ by definition. ∎

As protocol π_r can solve $\mathbb{F}_r$, the transcript of π_r can determine $\mathbb{F}_{r-1}(A_{r-1})$ when the output is correct with probability at least ρ, because $j^\star$ is randomly chosen. By Observation 5.6.1, we know that the probability of success of σ in solving $\mathbb{F}_{r-1}$ is the same as that of π_r for solving $\mathbb{F}_r$ times ρ and is thus at least $\delta \cdot \rho$ by the statement of Theorem 10.

It seems however that we have done nothing yet: σ is a r-round protocol with the *same* communication cost as π_r, so effectively we made no progress. The silver lining is that we can actually prove σ has a much lower *information cost* compared to π_r, using a direct-sum style argument.

Claim 5.6.2. *The information cost of protocol σ on the distribution $\mathcal{D}_{r-1}$ is at most*

$$IC(\sigma, \mathcal{D}_{r-1}) \leq \frac{1}{p_r} \cdot IC(\pi, \mathcal{D}_r).$$

Proof. Let j denote the random variable for the index $j^\star$ in σ and Π denote both the messages of players in σ as well as π_r – this is because, the players in σ communicate exactly the same messages as π_r (without loss of generality, we can assume Q_r also writes the message of Q_{r+1} on the board, even though all players can calculate that message on their own also). And, by Observation 5.6.1, we obtain that these messages are also distributed exactly the same way as π_r is a deterministic function of the samples from $\mathcal{D}_r = \mathcal{D}^{\text{REAL}}$. Moreover, since π_r is a deterministic protocol, the only *public* randomness of σ is j and t. Thus,

$$\begin{aligned}
\text{IC}(\sigma, \mathcal{D}_{r-1}) &= \mathbb{I}(A_{r-1}\,; \Pi \mid \mathsf{j}, \mathsf{t}) && \text{(by Definition 5.3.3)} \\
&= \sum_{j=1}^{p_r} \frac{1}{p_r} \cdot \mathbb{I}(\mathsf{B}_{\mathsf{t},j}\,; \Pi \mid \mathsf{t}, \mathsf{j} = j) \\
&\text{(by the distribution of } \mathsf{j} \text{ and since } A_{r-1} = B_{t,k} \text{ in } \sigma \text{ for any choice of } t \in [q_r]) \\
&= \sum_{j=1}^{p_r} \frac{1}{p_r} \cdot \mathbb{I}(\mathsf{B}_{\mathsf{t},j}\,; \Pi \mid \mathsf{t}), && (5.9)
\end{aligned}$$

where the final equality holds because of the following: the joint distribution of $(\mathsf{B}_{\mathsf{t},j}, \Pi, \mathsf{t})$ is a deterministic function of the choice of $\mathsf{A}_r \sim \mathcal{D}^{\text{REAL}}$ as fixing the input A_r also fixes the sub-instance $B_{t,j}$ (where t is the index of the special row of subinstances of A_r) as well as all the messages of protocol $\pi_r = \pi_r(A_r)$ which is deterministic. On the other hand, even given a fixed choice of A_r, we can still pick j uniformly at random from $[p_r]$ as it is entirely

independent of A_r. Thus, the joint distribution of $(\mathsf{B}_{\mathsf{t},j}, \Pi, \mathsf{t})$ is independent of the event $\mathsf{j} = j$ and we can drop the conditioning. [7]

The next observation is that for every $j \in [p_r]$, we have $\mathsf{B}_{\mathsf{t},j} \perp \mathsf{B}_{\mathsf{t},<j} \mid \mathsf{t}$ because the sub-instances are sampled independently and, after conditioning on t, the choice of $\mathsf{B}_{\mathsf{t},j}$ and $\mathsf{B}_{\mathsf{t},<j}$ only depend on the sub-instances. Thus, we can apply Proposition 4.3.2 to each of the mutual information terms above and get:

$$\begin{aligned}
\sum_{j=1}^{p_r} \mathbb{I}(\mathsf{B}_{\mathsf{t},j}\,; \Pi \mid \mathsf{t}) &\leq \sum_{j=1}^{p_r} \mathbb{I}(\mathsf{B}_{\mathsf{t},k}\,; \Pi \mid \mathsf{t}, \mathsf{B}_{\mathsf{t},<k}) \\
&= \mathbb{I}(\mathsf{B}_{\mathsf{t},*}\,; \Pi \mid \mathsf{t}) \quad \text{(by the chain rule of mutual information in Fact 4.3.1-(7))} \\
&\leq \mathbb{I}(\mathsf{B}_{\mathsf{t},*}\,; \Pi \mid \mathsf{t}) + \mathbb{I}(\mathsf{B}_{-\mathsf{t},*}\,; \Pi \mid \mathsf{B}_{\mathsf{t},*}, \mathsf{t}) + \mathbb{I}(\mathsf{t}\,; \Pi) \\
&\qquad \text{(as mutual information is non-negative (Fact 4.3.1-(2)))} \\
&= \mathbb{I}(\{\mathsf{B}_{i,j}\}_{i\in[q_r], j\in[p_r]}\,, \mathsf{t}\,; \Pi) \\
&\qquad \text{(by the chain rule of mutual information in Fact 4.3.1-(7))} \\
&= \mathbb{I}(\mathsf{A}_r\,; \Pi) \qquad \text{(as } A_r \text{ fixes } \{B_{i,j}\}_{i,j}\,, T \text{ and vice-versa by item } (iii)) \\
&= \mathrm{IC}(\pi_r, \mathcal{D}_r). \qquad \text{(by Definition 5.3.3)}
\end{aligned}$$

Plugging in this bound in Eq (5.9), we conclude that,

$$\mathrm{IC}(\sigma, \mathcal{D}_{r-1}) \leq \frac{1}{p_r} \cdot \mathrm{IC}(\pi_r, \mathcal{D}_r). \qquad \blacksquare$$

We are not done however as in the next step, we *really* need the communication cost of the protocol to be small and not only its information cost (see Remark 5.6.11). We do this by "compressing" the messages of σ down to their information cost. This part uses standard ideas except for the fact that we are applying them to a multi-party protocol instead of their typical two-party application.

In the following, we design a protocol $\overline{\sigma}$ by compressing the messages of σ on an input $A_{r-1} \sim \mathcal{D}_{r-1}$ down to their information cost. To avoid ambiguity, here, for every $\ell \in [r]$ and $i \in [r]$, we use $\Sigma_{i,\ell}$ to denote the message sent by player Q_ℓ of σ in round i. We also use R_σ to denote the public randomness of σ. We define $\overline{\Sigma}_{i,\ell}$ and $\mathsf{R}_{\overline{\sigma}}$ analogously for $\overline{\sigma}$. Protocol $\overline{\sigma}$ is as follows:

A communication efficient implementation of σ as a protocol $\overline{\sigma}$:

- For $i = 1$ to r rounds and $\ell = 1$ to r players in this order:
  1. Suppose the players at this point *all* know R_σ as well as
  $$\Sigma_{<i} := \Sigma_{<i,1}, \ldots, \Sigma_{<i,r}, \quad \text{and} \quad \Sigma_{i,<\ell} := \Sigma_{i,1}, \ldots, \Sigma_{i,\ell-1},$$

[7] The conditioning on t cannot be dropped in the same manner, because at the end of the first round, all the players know value of t, as it can be sent by Q_r. All subsequent messages depend on t.

i.e., the public randomness of σ and the messages that "should have been" communicated by σ in the first $i-1$ rounds plus the ones by $Q_1, \ldots, Q_{\ell-1}$ in round i.

2. Define the distributions

$$\mathcal{U}_{i,\ell} := \mathbf{\Sigma}_{i,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, R_\sigma, A_{r-1,\ell}, \quad \text{and} \quad \mathcal{V}_{i,\ell} := \mathbf{\Sigma}_{i,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, R_\sigma;$$

notice that the difference is only that in $\mathcal{U}_{i,\ell}$ we also condition on the input of player Q_ℓ in A_{r-1}, i.e., $A_{r-1,\ell}$ but in $\mathcal{V}_{i,\ell}$, we do not.

3. The player Q_ℓ plays the role of Alice in Proposition 5.3.5, who knows both $\mathcal{U}_{i,\ell}$ and $\mathcal{V}_{i,\ell}$ and all other players play the role of Bob who only knows $\mathcal{V}_{i,\ell}$. The players use public randomness and Alice (i.e., player Q_ℓ) writes the message of Proposition 5.3.5, denoted by $\overline{\sigma}_{i,\ell}$, on the blackboard which allows *all* players to sample a *single* message

$$\Sigma_{i,\ell} \sim \mathcal{A}_{i,\ell} = \mathbf{\Sigma}_{i,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, A_{r-1,\ell}, R_\sigma.$$

This allows the players to obtain the message $\Sigma_i(\ell)$ and continue this for-loop

The fact that $\overline{\sigma}$ is *faithfully* simulating the protocol σ follows immediately from Proposition 5.3.5 as the message compression approach has no error. We now bound the length of messages communicated by $\overline{\sigma}$ *in expectation.*

Claim 5.6.3. *The* expected *length of the messages communicated by the protocol $\overline{\sigma}$ is at most*

$$c_{\text{comp}} \cdot \Big(IC(\sigma, \mathcal{D}_{r-1}) + r^2 \Big),$$

where $c_{\text{comp}} \geq 1$ is an absolute constant defined in Eq (5.2) (derived from Proposition 5.3.5).

Proof. With a slight abuse of notation, we denote $\left|\overline{\Sigma}\right|$ to denote the length of the messages communicated by $\overline{\sigma}$ (on a particular input and realization of randomness) and $\left|\overline{\Sigma}_{i,\ell}\right|$ for $i \in [r]$ and $\ell \in [r]$ as the length of the message $\overline{\Sigma}_{i,\ell}$. Thus, by linearity of expectation,

$$\mathbb{E}\left|\overline{\Sigma}\right| = \sum_{i=1}^{r} \sum_{\ell=1}^{r} \mathbb{E}\left|\overline{\Sigma}_{i,\ell}\right|.$$

Moreover, by Proposition 5.3.5, for every $i \in [r]$ and $\ell \in [r]$, we have,

$$\mathbb{E}\left|\overline{\Sigma}_{i,\ell}\right| \leq c_{\text{comp}} \cdot \Big(\mathop{\mathbb{E}}_{\Sigma_{<i}, \Sigma_{i,<\ell}, A_{r-1,\ell}, R_\sigma} \mathbb{D}(\mathcal{U}_{i,\ell} \mid\mid \mathcal{V}_{i,\ell}) + 1 \Big)$$

(as the distributions of Proposition 5.3.5 are $\mathcal{U}_{i,\ell}, \mathcal{V}_{i,\ell}$ (conditioned on the prior messages))

$$\leq c_{\text{comp}} \cdot \Big(\mathop{\mathbb{E}}_{\Sigma_{<i}, \Sigma_{i,<\ell}, A_{r-1,\ell}, R_\sigma} \mathbb{D}(\mathbf{\Sigma}_{i,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, R_\sigma, A_{r-1,\ell} \mid\mid \mathbf{\Sigma}_{i,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, R_\sigma) + 1 \Big)$$

(by the definition of $\mathcal{U}_{i,\ell}, \mathcal{B}_{i,\ell}$)

$$= c_{\text{comp}} \cdot \Big(\mathbb{I}(\Sigma_{i,\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, \mathsf{R}_\sigma) + 1 \Big). \qquad \text{(by Fact 4.3.4)}$$

Plugging these bounds in the expectation-term above implies

$$\begin{aligned}
\mathbb{E}\left|\overline{\Sigma}\right| &\le \sum_{i=1}^{r}\sum_{\ell=1}^{r} c_{\text{comp}} \cdot \Big(\mathbb{I}(\Sigma_{i,\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, \mathsf{R}_\sigma) + 1 \Big) \\
&\le \sum_{i=1}^{r}\sum_{\ell=1}^{r} c_{\text{comp}} \cdot \Big(\mathbb{I}(\Sigma_{i,\ell}\,; \mathsf{A}_{r-1} \mid \Sigma_{<i}, \Sigma_{i,<\ell}, \mathsf{R}_\sigma) + 1 \Big) \\
&\qquad \text{(by Fact 4.3.1-(8) as } \mathsf{A}_{r-1,\ell} \text{ is a deterministic function of } \mathsf{A}_{r-1}) \\
&= c_{\text{comp}} \cdot \Big(\mathbb{I}(\{\Sigma_{i,\ell}\}_{i\in[r],\ell\in[r]}\,; \mathsf{A} \mid \mathsf{R}_\sigma) + r^2 \Big) \\
&\qquad \text{(by the chain rule of mutual information in Fact 4.3.1-(7))} \\
&= c_{\text{comp}} \cdot \Big(\mathrm{IC}(\sigma, \mathcal{D}_{r-1}) + r^2 \Big),
\end{aligned}$$

by the definition of information cost of σ in Definition 5.3.3. ∎

Finally, we do a basic clean up of $\overline{\sigma}$ to bound its communication cost (in worst-case, not in expectation). This step is entirely straightforward and is basically by truncating the protocol $\overline{\sigma}$ whenever a player is going to communicate more than their "budget". We only provide the full protocol here for completeness.

Protocol π': the truncated version of the protocol $\overline{\sigma}$ for a given parameter $\varepsilon > 0$

- Run the protocol $\overline{\sigma}$ as is. If at any point, a player is going to send a message that results in the communication cost of the protocol to become more than

$$\frac{c_{\text{comp}}}{\varepsilon} \cdot \Big(\mathrm{IC}(\sigma, \mathcal{D}_{r-1}) + r^2 \Big),$$

  terminate the protocol and return 'fail'.

We list the properties of the protocol π' in the following.

Claim 5.6.4. *For every $\varepsilon > 0$, protocol π' is a r-round r-party protocol for solving $\mathbb{F}_{r-1}$ on an instance $A_{r-1} \sim \mathcal{D}_{r-1}$ with probability of success at least $\delta - \varepsilon$ and communication cost*

$$CC(\pi') \le \frac{c_{\text{comp}}}{\varepsilon} \cdot \Big(\frac{s}{p_r} + r^2 \Big).$$

Proof. The fact that π' is r-round and r-party is by definition. A direct application of Markov bound, plus Claim 5.6.3 also implies that the probability that π' terminates is at most ε. This in turn implies that the error probability of π' compared to that $\overline{\sigma}$ and equivalently σ is at most ε more. Thus, π' succeeds with probability at least $\delta - \varepsilon$. Finally,

$$\mathrm{CC}(\pi') \underset{(1)}{\le} \frac{1}{\varepsilon} \cdot c_{\text{comp}} \cdot \Big(\mathrm{IC}(\sigma, \mathcal{D}_{r-1}) + r^2 \Big) \underset{(2)}{\le} \frac{c_{\text{comp}}}{\varepsilon} \cdot \Big(\frac{1}{p_r} \cdot \mathrm{IC}(\pi_r, \mathcal{D}_r) + r^2 \Big) \underset{(3)}{\le} \frac{c_{\text{comp}}}{\varepsilon} \cdot \Big(\frac{s}{p_r} + r^2 \Big),$$

where (1) holds because of the truncation step in π', (2) holds by Claim 5.6.2, and (3) holds since by Proposition 5.3.4, $\mathrm{IC}(\pi_r, \mathcal{D}_r) \leq \mathrm{CC}(\pi_r)$, which is the parameter s in Theorem 10. ∎

5.6.3 Step 2: A Protocol for $\mathbb{F}_{r-1}$ in $(r-1)$ Rounds via Round Elimination

Up until this point, we managed to obtain a protocol π' which has the desired communication cost and probability of success, but the main issue remains: it still requires r rounds as opposed to $(r-1)$ rounds. In the following, we try a more nuanced way of creating an $(r-1)$-round protocol π_{r-1} which follows the same approach as π' but no longer uses the first round of messages of π_r at all – instead, the players simply sample those messages using public randomness.

Before we proceed however, we need to establish an important property of the original protocol π_r using the fact the distribution $\mathcal{D}_r$ is a product distribution.

Conditional Independence of Inputs Even After Messages

We can prove that the distribution of the inputs to the players in $\mathcal{D}_r$ *remains* a product distribution (see item *(ii)*), even conditioned on the messages of the first round of π_r. This is a direct consequence of the rectangle property of protocols (and is a standard fact, which is proven here merely for completeness given we are conditioning on a subset of the input and not all of it).

Lemma 5.6.5. *Let Π_1 denote all messages of π_r in the first round. For any $i \in [q_r]$ and $j \in [p_r]$,*

$$\mathrm{dist}(B_{i,>j}, B_{>i,*} \mid B_{i,\leq j}, \Pi_1, B_{<i,*}) = \prod_{\ell=1}^{r} \mathrm{dist}(B_{i,>j,\ell}, B_{>i,*,\ell} \mid B_{i,\leq j,\ell}, \Pi_1, B_{<i,*}).$$

We will first show some necessary independence of distributions before proving the lemma.

Claim 5.6.6. *For any $\ell \in [r], j \in [p_r]$, $i \in [q_r]$, and any choice of $\Pi_1, B_{i,\leq j,\ell}, B_{<i,*}$,*

$$\mathsf{B}_{i,\leq j,-\ell}, \mathsf{B}_{i,>j,<\ell}, \mathsf{B}_{>i,*,<\ell} \perp \mathsf{B}_{i,>j,\ell}, \mathsf{B}_{>i,*,\ell} \mid \Pi_1, B_{i,\leq j,\ell}, B_{<i,*}.$$

Proof. We will show that the following mutual information term is zero, which immediately implies the claim by Fact 4.3.1-(2):

$$\begin{aligned}
&\mathbb{I}(\mathsf{B}_{i,\leq j,-\ell}, \mathsf{B}_{i,>j,<\ell}, \mathsf{B}_{>i,*,<\ell}\,; \mathsf{B}_{i,>j,\ell}, \mathsf{B}_{>i,*,\ell} \mid \Pi_1, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \\
&\quad\leq \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Pi_1, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \\
&\qquad\qquad \text{(by the data processing inequality of Fact 4.3.1-(8) and Item (iii))} \\
&\quad= \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Pi_{1,\leq\ell}, \Pi_{1,>\ell}, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \qquad \text{(by splitting } \Pi_1 = \Pi_{1,\leq\ell}, \Pi_{1,>\ell}) \\
&\quad\leq \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Pi_{1,\leq\ell}, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \\
&\text{(by Proposition 4.3.3 as } \Pi_{1,>\ell} \perp \mathsf{A}_{r-1,\ell} \mid A_{r-1,-\ell}, \Pi_{1,\leq\ell}, B_{i,\leq j,\ell}, B_{<i,*}, \text{ and } \Pi_{1,>\ell} \text{ is now fixed)} \\
&\quad= \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Pi_{1,<\ell}, \Pi_{1,\ell}, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \quad \text{(by further splitting } \Pi_{1,\leq\ell} = \Pi_{1,<\ell}, \Pi_{1,\ell}) \\
&\quad\leq \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,; \mathsf{A}_{r-1,\ell} \mid \Pi_{1,<\ell}, \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*}) \\
&\text{(by Proposition 4.3.3 as } \Pi_{1,\ell} \perp \mathsf{A}_{r-1,-\ell} \mid A_{r-1,\ell}, \Pi_{1,<\ell}, B_{i,\leq j,\ell}, B_{<i,*}, \text{ and } \Pi_{1,\ell} \text{ is now fixed)}
\end{aligned}$$

$$\leq \mathbb{I}(\mathsf{A}_{r-1,-\ell}\,;\mathsf{A}_{r-1,\ell} \mid \mathsf{B}_{i,\leq j,\ell}, \mathsf{B}_{<i,*})$$
(by Proposition 4.3.3 as $\mathsf{\Pi}_{1,<\ell} \perp \mathsf{A}_{r-1,\ell} \mid A_{r-1,-\ell}, B_{i,\leq j,\ell}, B_{<i,*}$; $\mathsf{\Pi}_{1,<\ell}$ is now fixed)

$= 0.$ ∎ (by item *(ii)* all $B_{i,j}$'s are independent and we can apply Fact 4.3.1-(2))

Lemma 5.6.5 follows easily now.

Proof of Lemma 5.6.5. We have that,

$$\begin{aligned}
&\operatorname{dist}(B_{i,>j}, B_{>i,*} \mid B_{i,\leq j}, \Pi_1, B_{<i,*}) \\
&\quad= \prod_{\ell=1}^{r} \operatorname{dist}(B_{i,>j,\ell}, B_{>i,*,\ell} \mid B_{i,>j,<\ell}, B_{>i,*,<\ell}, B_{i,\leq j}, \Pi_1, B_{<i,*}) && \text{(by chain rule)} \\
&\quad= \prod_{\ell=1}^{r} \operatorname{dist}(B_{i,>j,\ell}, B_{>i,*,\ell} \mid B_{i,>j,<\ell}, B_{>i,*,<\ell}, B_{i,\leq j,-\ell}, B_{i,\leq j,\ell}, \Pi_1, B_{<i,*}) \\
&\qquad\qquad && \text{(by splitting } B_{i,\leq j} = B_{i,\leq j,-\ell}, B_{i,\leq j,\ell}) \\
&\quad= \prod_{\ell=1}^{r} \operatorname{dist}(B_{i,>j,\ell}, B_{>i,*,\ell} \mid \Pi_1, B_{i,\leq j,\ell}, B_{<i,*}), && \text{(by Claim 5.6.6)}
\end{aligned}$$

completing the proof. ∎

Eliminating the First Round of the Protocol π_r

We are now ready to proceed with designing our $(r-1)$-round r-party protocol π_{r-1} for $\mathbb{F}_{r-1}$ using π_r (and the intermediate protocol π' designed in the previous step). Recall the intermediate protocol σ in the previous step. This protocol (and subsequent ones $\overline{\sigma}$ and π') consists of two separate parts: (1) an *embedding* part that created an entire graph A_r from $\mathcal{D}_r$ by placing the input $A_{r-1} \sim \mathcal{D}_{r-1}$ as one of its special sub-instances; and, (2) a *simulation* part that ran the protocol π_r (either directly in σ or indirectly in $\overline{\sigma}$ and π') on this input. In the following, we will change the embedding part (thoroughly) but stick with the same simulation part except that we will only run those simulations from the second round onwards. Formally,

An $(r-1)$-round r-party protocol π_{r-1} for $\mathbb{F}_{r-1}(A_{r-1})$:

1. Sample first-round messages $\Pi_{1,1}, \ldots, \Pi_{1,r}$ of π_r from $\mathcal{D}_r$ and indices $j^\star \in [p_r]$ and $t \in [q_r]$ uniformly at random and independently using public randomness[a].
2. Sample $B_{<t,*}, B_{t,<j^\star}$ from $\mathcal{D}_r \mid \Pi_{1,1}, \ldots, \Pi_{1,r}, \mathsf{t} = t$ using public randomness.
3. For every $\ell \in [r]$, each player Q_ℓ sets $B_{t,j^\star,\ell} = A_{r-1,\ell}$ and samples
$$\left(A_{r,\ell} \sim \mathcal{D}_r \mid \Pi_{1,1}, \ldots, \Pi_{1,r}, \mathsf{t} = t, B_{<t,*}, B_{t,<j^\star}, \mathsf{B}_{t,j^\star,\ell} = A_{r-1,\ell}\right)$$
using private randomness. Let $A_r = (A_{r,1}, A_{r,2}, \ldots, A_{r,r+1})$ be the input of all players where, by item *(iv)*, the input of the simulated player Q_{r+1} is fixed by $\mathsf{t} = t$.

4. The players run $\pi_r(A_r)$ from its second round onwards, following the protocol π' of Claim 5.6.4.[b]

5. At the end, to solve $\mathbb{F}_{r-1}(A_{r-1})$, use the transcript of π_r for $\mathbb{F}_r(A_r)$, and this will have the answer to $\mathbb{F}_{r-1}(B_{t,j^\star})$ with probability at least ρ for random $j^\star$ (see item (v), and π' generates the messages of $\pi_r(A_r)$ implicitly).

[a] We emphasize that we are *not* sampling the message $\Pi_{1,r+1}$ in this step (as it is determined by t).

[b] Protocol σ runs protocol π_r, and the first round messages are sampled from the same distribution. The simulation of protocol σ in protocol $\overline{\sigma}$ can be started from the second round also, if the players already know the messages of the first round, as is happening here. They truncate any large message from the second round which is larger than the threshold given in π'.

Similar to step one and distribution $\mathcal{D}^{\text{REAL}}$, we also define the **distribution** $\mathcal{D}^{\text{FAKE}}$ as the distribution of the graph A_r obtained in the protocol π_{r-1} when $A_{r-1} \sim \mathcal{D}_{r-1}$. Unlike before, however, it is no longer the case that $\mathcal{D}^{\text{FAKE}}$ is actually the "right" distribution of the input that π_r expects; for instance, we now *embedded* $B_{t,j^\star} = A_{r-1}$ *without* conditioning on the messages $\Pi_{1,1}, \ldots, \Pi_{1,r}$ in the protocol[8]. In particular, we can write these distributions as the following (where the random variables on the RHS of each term is distributed according to $\mathcal{D}_r$):

$$\begin{aligned}\text{dist}(\Pi_1, j^\star, A_r \sim \mathcal{D}^{\text{REAL}}) := &(\Pi_1, \mathsf{t}, \mathsf{j}) \times (\mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}} \mid \Pi_1, \mathsf{t}, \mathsf{j}) \\ &\times (\mathsf{B}_{\mathsf{t},\mathsf{j}} \mid \Pi_1, \mathsf{t}, \mathsf{j}, \mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}}) \times (\mathsf{A}_r \mid \Pi_1, \mathsf{t}, \mathsf{j}, \mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}}, \mathsf{B}_{\mathsf{t},\mathsf{j}})\end{aligned} \tag{5.10}$$

$$\begin{aligned}\text{dist}(\Pi_1, j^\star, A_r \sim \mathcal{D}^{\text{FAKE}}) := &(\Pi_1, \mathsf{t}, \mathsf{j}) \times (\mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}} \mid \Pi_1, \mathsf{t}, \mathsf{j}) \\ &\times (\mathsf{B}_{\mathsf{t},\mathsf{j}}) \times \prod_{\ell=1}^{r} (\mathsf{A}_{r,\ell} \mid \Pi_1, \mathsf{t}, \mathsf{j}, \mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}}, \mathsf{B}_{\mathsf{t},\mathsf{j},\ell}) \,.\end{aligned} \tag{5.11}$$

In the above, with a slight abuse of notation, we wrote a triple $(\Pi, t, j^\star, A_r) \sim \mathcal{D}^{\text{REAL}}$ (or $\sim \mathcal{D}^{\text{FAKE}}$) to denote the joint distribution of *all* these variables when the input graph A_r is sampled from $\mathcal{D}^{\text{REAL}}$ versus when they are sampled from $\mathcal{D}^{\text{FAKE}}$.

Nevertheless, we are going to prove that $\mathcal{D}^{\text{FAKE}}$ is not that different from $\mathcal{D}^{\text{REAL}}$ either. In particular, we prove the following lemma.

Lemma 5.6.7 ("distributions induced by $\mathcal{D}^{\text{REAL}}$ and $\mathcal{D}^{\text{FAKE}}$ are close")**.**

$$\|\text{dist}(\Pi_1, j^\star, A_r \sim \mathcal{D}^{\text{REAL}}) - \text{dist}(\Pi_1, j^\star, A_r \sim \mathcal{D}^{\text{FAKE}})\|_{\text{tvd}} \leq \sqrt{\frac{s}{2\, p_r \cdot q_r}}.$$

We will bound this difference term by term in Eq (5.10) and Eq (5.11). The first two terms are equal in both. The next, and the main, claim bounds the difference between the

[8] This is inevitable: we are given $A_{r-1} \sim \mathcal{D}_{r-1}$ as part of the input and *not* from $\mathcal{D}_{r-1} \mid \Pi_1$ (and we have to sample Π_1 instead of spending a whole round computing it). So, even though clearly in the protocol π_r, $\Pi_1 \not\perp B_{t,j^\star}$ can happen, these two variables are always independent in $\mathcal{D}^{\text{FAKE}}$.

third terms. This is yet another application of direct-sum style arguments along the lines of Claim 5.6.2, although fundamentally different as we need a much stronger guarantee (that only holds for the first round of the protocol π_r). We emphasize that the following claim is talking about π_r and *not* π_{r-1} (as the variables in RHS of Eq (5.10) and Eq (5.11) are distributed according to π_r).

Claim 5.6.8.

$$\mathop{\mathbb{E}}_{\Pi_1,\mathsf{t},\mathsf{j},\mathsf{B}_{<t,*},\mathsf{B}_{t,<j^\star}} \|(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=j^\star) - \mathsf{B}_{t,j^\star}\|_{\mathrm{tvd}} \leq \sqrt{\frac{s}{2\,p_r \cdot q_r}}.$$

Proof. We follow the standard plan of bounding the KL-divergence of the above distributions and applying Pinsker's inequality (Fact 4.3.10) at the end. In this lemma, *all* variables are with respect to the protocol π_r on the distribution $\mathcal{D}_r$. The only exception is that of j which is chosen uniformly from $[p_r]$ (and is not defined explicitly in $\mathcal{D}_r$ beforehand).

By item *(ii)*, the first r players do not have any information about random variable t, and thus their messages in the first round cannot reveal too much about *all* of the special sub-instances, and in particular $B_{t,j^\star}$. The last player knows t but has no access to *any* of the sub-instances, so should not be able to reveal anything about the special sub-instances. Overall, the first message cannot change the distribution of $\mathsf{B}_{t,j^\star}$, for a random chosen $t \in [q_r]$ and $j^\star \in [p_r]$ by much. Our proof formalizes this idea to get a bound on the KL-Divergence. In the following, we denote

$$\Pi_{1,\leq r} := (\Pi_{1,1}, \Pi_{1,2}, \ldots, \Pi_{1,r}),$$

i.e., the first-round messages of all players except for the last one.

For every $i \in [q_r]$ and $j \in [p_r]$,

$$\mathsf{B}_{i,j} \perp \mathsf{B}_{<i,*}, \mathsf{B}_{i,<j}, \mathsf{t}=i, \mathsf{j}=j, \tag{5.12}$$

because sub-instances are chosen independently in $\mathcal{D}_r$ from each other, as well as the special UPC selected by t; the choice of j is also independent of the entire graph. We can now use this and have,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\Pi_1,\mathsf{t},\mathsf{j},\mathsf{B}_{<t,*},\mathsf{B}_{t,<j^\star}} \mathbb{D}(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=j^\star \;||\; \mathsf{B}_{t,j^\star}) \\
&\quad = \mathop{\mathbb{E}}_{\Pi_1,\mathsf{t},\mathsf{j},\mathsf{B}_{<t,*},\mathsf{B}_{t,<j^\star}} \mathbb{D}(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=j^\star \;||\; \mathsf{B}_{t,j^\star} \mid B_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=j^\star) \\
&\qquad\qquad \text{(the distribution is the same by Eq (5.12))} \\
&\quad = \mathop{\mathbb{E}}_{\Pi_{1,\leq r},\mathsf{t},\mathsf{j},\mathsf{B}_{<t,*},\mathsf{B}_{t,<j^\star}} \mathbb{D}(\mathsf{B}_{t,j^\star} \mid \Pi_{1,\leq r}, \mathsf{B}_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=j^\star \;||\; \mathsf{B}_{t,j^\star} \mid B_{<t,*}, B_{t,<j^\star}, \mathsf{t}=t, \mathsf{j}=k) \\
&\quad \text{(because } \Pi_{1,r+1} \text{ is fixed by } t \text{ and } \Pi_{1,\leq r} \text{ because it fixes the input of } Q_{r+1} \text{ by item } (iv)) \\
&\quad = \mathbb{I}(\Pi_{1,\leq r}\,; \mathsf{B}_{\mathsf{t},\mathsf{j}} \mid \mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}}, \mathsf{t}, \mathsf{j}). \qquad \text{(by Fact 4.3.4)}
\end{aligned}$$

In words, the change in the KL-divergence of the distribution of the sub-instance $B_{t,j^\star}$ from its original distribution by the messages of first round is bounded by the information revealed by $\Pi_{1,\leq r}$ about this sub-instance.

To bound this mutual information term, we use the following simple observation:

Observation 5.6.9. *We have, for any $i \in [q_r], j \in [p_r]$,*

$$\Pi_{\leq r}, \mathsf{B}_{i,j}, \mathsf{B}_{<i,*}, \mathsf{B}_{i,<j} \perp \mathsf{t}, \mathsf{j}.$$

This observation holds because the sub-instances $\mathsf{B}_{i,j}$ for $i \in [q_r]$ and $j \in [p_r]$ are sampled independently of t and j in distribution $\mathcal{D}_r$, and this independence continues to hold even conditioned on messages by the rectangle property. Nonetheless, we show this formally with a mutual information based proof.

Proof of Observation 5.6.9. We have,

$$\begin{aligned}
&\mathbb{I}(\Pi_{\leq r}, \mathsf{B}_{i,j}, \mathsf{B}_{<i,*}, \mathsf{B}_{i,<j} \,;\, \mathsf{t}, \mathsf{j}) \\
&\leq \mathbb{I}(\mathsf{B}_{\leq q_r,*} \,;\, \mathsf{t}, \mathsf{j}) && \text{(by Fact 4.3.1-(8) as } \Pi_{\leq r} \text{ is fixed by all the sub-instances)} \\
&\leq 0. && \text{(as the sub-instances are independent of } \mathsf{t} \text{ and } \mathsf{j}\text{, by Fact 4.3.1-(2))}
\end{aligned}$$

Fact 4.3.1-(2) now completes the proof. ∎

We now bound the information term as follows:

$$\begin{aligned}
\mathbb{I}(\Pi_{1,\leq r} \,;\, \mathsf{B}_{\mathsf{t},\mathsf{j}} \mid \mathsf{B}_{<\mathsf{t},*}, \mathsf{B}_{\mathsf{t},<\mathsf{j}}, \mathsf{t}, \mathsf{j}) &= \mathop{\mathbb{E}}_{\mathsf{t},\mathsf{j}} \left[\mathbb{I}(\Pi_{1,\leq r} \,;\, \mathsf{B}_{t,j^\star} \mid \mathsf{B}_{<t,*}, \mathsf{B}_{t,<j^\star}, \mathsf{t} = t, \mathsf{j} = j^\star) \right] \\
&\text{(by the definition of conditional mutual information)} \\
&= \mathop{\mathbb{E}}_{\mathsf{t},\mathsf{j}} \left[\mathbb{I}(\Pi_{1,\leq r} \,;\, \mathsf{B}_{t,j^\star} \mid \mathsf{B}_{<t,*}, \mathsf{B}_{t,<j^\star}) \right] \\
&\text{(as } (\Pi_{1,\leq r}, \mathsf{B}_{t,j^\star}, \mathsf{B}_{<t,*}, \mathsf{B}_{t,<j^\star}) \text{ are all independent of } \mathsf{t}, \mathsf{j}\text{, see Observation 5.6.9)} \\
&= \frac{1}{q_r} \cdot \frac{1}{p_r} \cdot \sum_{t=1}^{q_r} \sum_{j^\star=1}^{p_r} \mathbb{I}(\Pi_{1,\leq r} \,;\, \mathsf{B}_{t,j^\star} \mid \mathsf{B}_{<t,*}, \mathsf{B}_{t,<j^\star}) \\
&\text{(as the distribution of both } \mathsf{t}, \mathsf{j} \text{ is uniform and independent of each other)} \\
&= \frac{1}{q_r \cdot p_r} \cdot \mathbb{I}(\Pi_{1,\leq r} \,;\, \{\mathsf{B}_{i,j}\}_{i \in [q_r], j \in [p_r]}) \\
&\text{(by the chain rule of mutual information in Fact 4.3.1-(7))} \\
&\leq \frac{1}{q_r \cdot p_r} \cdot \mathbb{I}(\Pi \,;\, \mathsf{A}) \\
&\text{(by the data processing inequality of Fact 4.3.1-(8) and item } \textit{(iii)} \text{ on inputs of players)} \\
&= \frac{1}{q_r \cdot p_r} \cdot \mathrm{IC}(\pi_r, \mathcal{D}_r) \\
&\text{(by the definition of external information (Definition 5.3.3))} \\
&\leq \frac{1}{q_r \cdot p_r} \cdot \mathrm{CC}(\pi_r) && \text{(by Proposition 5.3.4)}
\end{aligned}$$

which is $s/(q_r \cdot p_r)$ as s is the total communication cost of π_r. We thus have,

$$\mathop{\mathbb{E}}_{\Pi_1, \mathsf{t}, \mathsf{j}, \mathsf{B}_{t,<j^\star}, \mathsf{B}_{<t,*}} \mathbb{D}(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{t,<j^\star}, B_{<t,*} \,||\, \mathsf{B}_{t,j^\star}) \leq \frac{s}{q_r \cdot p_r}. \tag{5.13}$$

We can finish the proof as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\Pi_1,\mathsf{t},\mathsf{j},\mathsf{B}_{t,<j^\star},\mathsf{B}_{<t,*}} \|(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{t,<j^\star}, B_{<t,*}) - \mathsf{B}_{t,j^\star}\|_{\mathrm{tvd}} \\
&\quad\le \mathop{\mathbb{E}}_{\Pi_1,\mathsf{t},\mathsf{j},\mathsf{B}_{t,<j^\star},\mathsf{B}_{<t,*}} \left[\sqrt{1/2 \cdot \mathbb{D}(\mathsf{B}_{t,j^\star} \mid \Pi_1, B_{t,<j^\star}, B_{<t,*} \mid\mid \mathsf{B}_{t,j^\star})}\right] \\
&\qquad\qquad\qquad

We can now conclude the proof of Theorem 10, by showing that the protocol π_{r-1} can solve $\mathbb{F}_{r-1}$ with the desired probability.

Proof of Theorem 10. Suppose we sample an instance $A_{r-1} \sim \mathcal{D}_{r-1}$ and run protocol π' of step one in Section 5.6.2 on this instance by creating the messages and instance $(\Pi_1, A_r) \sim \mathcal{D}^{\text{REAL}}$; then, by Claim 5.6.4,

$$\Pr_{\substack{A_{r-1}\sim\mathcal{D}_{r-1}\\(\Pi_1,j^\star,A_r)\sim\mathcal{D}^{\text{REAL}}}}\Big(\pi' \text{ succeeds in solving } \mathbb{F}_{r-1}\Big) \geq \delta\cdot\rho - \varepsilon, \tag{5.14}$$

for the parameters δ and ε in the lemma's statement.

On the other hand, what we are actually doing in π_{r-1} is to sample $A_{r-1} \sim \mathcal{D}_{r-1}$ but then run π' of step one on this graph by creating messages and instance $(\Pi_1, A_r) \sim \mathcal{D}^{\text{FAKE}}$ instead. Thus,

$$\Pr_{A_{r-1}\sim\mathcal{D}_{r-1}}\Big(\pi_{r-1} \text{ succeeds in solving } \mathbb{F}_{r-1}\Big) = \Pr_{\substack{A_{r-1}\sim\mathcal{D}_{r-1}\\(\Pi_1,j^\star,A_r)\sim\mathcal{D}^{\text{FAKE}}}}\Big(\pi' \text{ succeeds in solving } \mathbb{F}_{r-1}\Big). \tag{5.15}$$

Nevertheless, the RHS of this equation and the LHS of the above one are quite close to each other by Lemma 5.6.7 so we should be able to extend the first equation to the second one as well. Formally,

$$\begin{aligned}
&\Pr_{\substack{A_{r-1}\sim\mathcal{D}_{r-1}\\(\Pi_1,j^\star,A_r)\sim\mathcal{D}^{\text{FAKE}}}}\Big(\pi' \text{ succeeds in solving } \mathbb{F}_{r-1}\Big)\\
&\quad\geq \Pr_{\substack{A_{r-1}\sim\mathcal{D}_{r-1}\\(\Pi_1,j^\star,A_r)\sim\mathcal{D}^{\text{REAL}}}}\Big(\pi' \text{ succeeds in solving } \mathbb{F}_{r-1}\Big) - \|(\Pi_1, j^\star, A_r)\sim\mathcal{D}^{\text{REAL}} - (\Pi_1, j^\star, A_r)\sim\mathcal{D}^{\text{FAKE}}\|_{\text{tvd}}\\
&\qquad\qquad\qquad

of the proof above. Had we directly used σ, then, in the last step, we should have bounded the information cost of $\overline{\sigma}$ on the new *distribution $\mathcal{D}^{\text{FAKE}}$ instead of $\mathcal{D}^{\text{REAL}}$ – while, by Lemma 5.6.7, these two distributions are statistically close, this does* not *imply that the information cost of $\overline{\sigma}$ on $\mathcal{D}^{\text{FAKE}}$ will be small also, which stops us from applying the induction hypothesis.*

5.7 Using New Round Elimination

In this section, we use Theorem 10 to analyze the lower bound for distributions $\{\mathcal{G}_r\}_{r\geq 1}$ in this section, and conclude the proof of Theorem 9. By Lemma 5.5.10, we need to focus on the ability of protocols for solving the search predicate. The following lemma captures our lower bound for this task.

Lemma 5.7.1. *For any $r \geq 1$, any r-round protocol π that given $G \sim \mathcal{G}_r(n)$ (for n, r satisfying Eq (5.3)), can solve $\mathbb{P}_r$ on input graph G with probability of success at least*

$$2^{-p_0} \cdot \left(1 + \frac{r}{20 \cdot (r+1)}\right)$$

has communication cost

$$CC(\pi) \geq s_r(n) := \frac{1}{{n_0}^2 \cdot (2^{p_0} \cdot 40 \cdot c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot e^{3r \ln^{5/6}(n)}} \cdot \left((n) \uparrow \left(1 + \frac{1}{2^r - 1}\right)\right).$$

We do note that ignoring all extra (and lower order) terms in the lemma (that are needed for a proper inductive argument), the lemma simply says that obtaining any probability of success better than 2^{-p_0} (an arbitrarily small constant) requires roughly $n^{1+1/(2^r-1)}$ communication.

We prove Lemma 5.7.1 inductively using the *round elimination* argument: if we have a "very good" r-round protocol, then we should be able to eliminate its first round and also obtain a "good enough" $(r-1)$-round protocol; keep doing this then eventually bring us to a "non-trivial" protocol for the $r = 0$ case, which we know cannot exist. The following lemma—which is the heart of the proof—allows us to establish the induction step.

We establish that the problelm $\mathbb{P}_r$, obeys the properties required of $\mathcal{D}_r$ in Theorem 10.

Claim 5.7.2. *The problem of solving a predicate $\mathbb{P}_r$ in graphs $G \sim \mathcal{G}_r(n)$ (for n, r satisfying Eq (5.3)), and $\mathbb{P}_{r-1}$ on graphs $G' \sim \mathcal{G}_{r-1}(n_{r-1})$, for n_{r-1} from Eq (5.6) obey the properties (i) to (v) expected of $\mathcal{D}_r$.*

Proof. Property (*i*) holds by definition. Property (*ii*) holds by Property (5.5.6) of $\mathcal{G}_r(n)$.

Property (*iii*) is true by the definition of $\mathcal{G}_r(n)$, specifically, line (2) in the description of $\mathcal{G}_r(n)$. Moreover, Property (5.5.7) ensures that the independence of the subinstances is preserved, as the subgraphs $H_{t,j}$ for $j \in [p_r]$ are vertex disjoint. Property (*iv*) is true by line (5) in the description. This is stated in Property (5.5.5) explicitly also.

The last property (*v*) is given by Lemma 5.5.10. This lemma proves that for every valid search sequence $K = (k_r, k_{r-1}, \ldots, k_1)$, the MIS uniquely determines $\mathbb{P}_r(G, K)$. For any

subgraph $H_{t,j}$ for $j \in [p_r]$, for any valid search sequence $K' = (k'_{r-1}, \ldots, k'_1)$ for $H_{t,j}$, we have:

$$\mathbb{P}_{r-1}(H_{t,j}, (k'_{r-1}, \ldots, k'_1)) = \mathbb{P}_r(G, (j, k'_{r-1}, \ldots, k'_1)),$$

hence the transcript can be used to solve $\mathbb{P}_{r-1}(H_{t,j})$ also for all $j \in [p_r]$. Hence, property (v) holds with $\rho = 1$. ∎

A simple corollary of Claim 5.7.2 and Theorem 10 gives us the following round elimination result for $\mathcal{G}_r(n)$.

Corollary 5.7.3. *For every $0 < \varepsilon < \delta < 1$ and integer $s \geq 1$, the following is true. Suppose there is a deterministic r-round $(r+1)$-party protocol π_r with communication cost $CC(\pi_r) \leq s$ that solves predicate $\mathbb{P}_r$ with probability at least δ for a graph $G \sim \mathcal{G}_r(n)$ (for n, r satisfying Eq (5.3)). Then, for constant c_{comp} (from Eq (5.2)), there is an $(r-1)$-round r-party deterministic protocol π_{r-1} with communication cost*

$$CC(\pi_{r-1}) \leq \frac{c_{\text{comp}}}{\varepsilon} \cdot \left(\frac{s}{p_r} + r^2\right)$$

that solves $\mathbb{P}_{r-1}$ for $G' \sim \mathcal{G}_{r-1}(n_{r-1})$ (for n_{r-1} from Eq (5.6)) with probability of success at least

$$\delta - \varepsilon - \sqrt{\frac{s}{2p_r \cdot q_r}}.$$

5.7.1 Proof of Lemma 5.7.1

We now use Corollary 5.7.3 to complete the proof of Lemma 5.7.1 (restated below). At this stage, the main arguments have been made and the remainder of the proof, for the most part, is a tedious calculation based on the parameters set in Eq (5.6) and Eq (5.8).

Lemma (Restatement of Lemma 5.7.1)**.** *For any $r \geq 1$, any r-round protocol π that given $G \sim \mathcal{G}_r(n)$ (for n, r satisfying Eq (5.3)), can solve $\mathbb{P}_r$ on input graph G with probability of success at least*

$$2^{-p_0} \cdot \left(1 + \frac{r}{20 \cdot (r+1)}\right)$$

has communication cost

$$CC(\pi) \geq s_r(n) := \frac{1}{{n_0}^2 \cdot (2^{p_0} \cdot 40 \cdot c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot e^{3r \ln^{5/6}(n)}} \cdot \left((n) \uparrow \left(1 + \frac{1}{2^r - 1}\right)\right).$$

Proof. Given that there is a distribution over G and we are working with communication cost (which is a worst-case measure), it suffices to prove Lemma 5.7.1 for *deterministic* protocols; the result for randomized protocols then follows immediately by the easy direction of Yao's minimax principle (namely, by fixing the randomness of the protocol to its "best" choice using an averaging argument). We prove the lemma using induction on r.

Base Case. We prove the base case when $r = 1$. Assume towards a contradiction that there exists a protocol π_1 for $\mathcal{G}_1(n)$ which uses communication

$$\mathrm{CC}(\pi_1) < s_1(n) = \frac{1}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}}) \cdot (2!)^2 \cdot e^{3\ln^{5/6}(n)}} \cdot n^2,$$

and success probability at least $2^{-p_0} \cdot 41/40$. We will use Corollary 5.7.3 on π_1 to get a 0-round protocol.

We know that a 0-round protocol cannot change the prior distribution of $\mathbb{P}_0(G, \emptyset)$ for $G \sim \mathcal{G}_0(n_0)$ which is uniform over $\{0,1\}^{p_0}$. Thus, the probability that the 0-round protocol can solve the predicate $\mathbb{P}_0$ is 2^{-p_0} as it returns a fixed answer to $\mathbb{P}_0(G, \emptyset)$ independent of G.

We use Corollary 5.7.3 on π_1 with the following parameters (from Eq (5.8) for $r = 1$):

$$\epsilon = \frac{1}{160 \cdot 2^{p_0}} \qquad p_1 = \frac{2b_1}{\exp(\eta_p \cdot (\ln(2b_1))^{3/4})} \qquad q_1 = \frac{2b_1}{\exp(\eta_q \cdot (\ln(2b_1))^{3/4})}.$$

We get a 0-round protocol π_0, which cannot communicate any bits. We will just analyze the probability of success of π_0. We have,

$$\begin{aligned}
\frac{s_1(n)}{p_1 \cdot q_1} &= \underbrace{\frac{1}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}}) \cdot (2!)^2 \cdot e^{3\ln^{5/6}(n)}} \cdot n^2}_{=s_1} \cdot \underbrace{\frac{\exp(\eta_p \cdot (\ln(2b_1))^{3/4})}{2b_1}}_{=1/p_1} \cdot \underbrace{\frac{\exp(\eta_q \cdot (\ln(2b_1))^{3/4})}{2b_1}}_{=1/q_1} \\
&= \frac{1}{16 \cdot 40 \cdot 2^{p_0} \cdot c_{\mathrm{comp}}} \cdot \frac{(2 \cdot b_1 n_0)^2}{4 \cdot b_1^2 \cdot n_0^2} \cdot \exp((\eta_p + \eta_q) \cdot \ln^{3/4}(2b_1) - 3\ln^{5/6}(n)) \\
&\qquad \text{(by choice of } b_1 = (n/2n_0) \text{ from Eq (5.4))} \\
&\leq \frac{1}{16 \cdot 40 \cdot 2^{p_0} \cdot c_{\mathrm{comp}}} \cdot \exp(-2\ln^{5/6}(n)) \\
&\qquad \text{(as } \eta_p = O(1), \eta_q = O(1),\ 2b_1 < n, \text{ and } O(\ln^{3/4}(n)) < \ln^{5/6}(n) \text{ for large } n) \\
&\leq \left(\frac{1}{160 \cdot 2^{p_0}}\right)^2. \qquad \text{(taking a loose but sufficient upper bound)}
\end{aligned}$$

Thus, by Corollary 5.7.3, the probability of success of π_0 is at least,

$$\frac{41}{40 \cdot 2^{p_0}} - \underbrace{\frac{1}{160 \cdot 2^{p_0}}}_{\varepsilon} - \underbrace{\frac{1}{160 \cdot 2^{p_0}}}_{\sqrt{s_1(n)/(p_1 \cdot q_1)}} = \frac{81}{80 \cdot 2^{p_0}} > \frac{1}{2^{p_0}},$$

which is a contradiction. This proves the induction base for $r = 1$.

Induction Step. We now prove the induction step wherein we assume the statement is true for $r - 1$ and we prove it for $r > 1$. Suppose towards a contradiction that there exists a protocol π_r with probability of success

$$\delta := \frac{1}{2^{p_0}} \left(1 + \frac{r}{20 \cdot (r+1)}\right)$$

for solving $\mathbb{P}_r$ as in the lemma statement, but with communication cost $\mathrm{CC}(\pi_r) < s_r(n)$ for the parameter $s_r(n)$ in the lemma statement. Now, recall from Eq (5.5) and Eq (5.8) that any graph sampled from distribution $\mathcal{G}_r$ has the following parameters:

$$\begin{aligned}
n &= 2 \cdot b_r \cdot n_{r-1} = 2 \cdot (n_{r-1}) \uparrow \left(\frac{2^r - 1}{2^{r-1} - 1}\right), \\
b_r &= (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1} - 1}\right) \\
p_r &= \frac{b_r \cdot 2^r}{\exp(\eta_p \cdot (\ln(b_r \cdot 2^r))^{3/4})} = \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right) \cdot 2^r}{\exp\left(\eta_p \cdot (r \ln 2 + \ln(b_r))^{3/4}\right)}, \\
q_r &= \frac{b_r \cdot 2^r}{\exp(\eta_q \cdot (\log(b_r \cdot 2^r))^{3/4})} = \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right) \cdot 2^r}{\exp\left(\eta_q \cdot (r \ln 2 + \log(b_r))^{3/4}\right)}.
\end{aligned} \tag{5.16}$$

We use Corollary 5.7.3 on protocol π_r with parameter

$$\varepsilon := \frac{1}{2^{p_0} \cdot 40 \cdot (r+1)^2}$$

to get a protocol π_{r-1} for $\mathbb{P}_{r-1}$ over $\mathcal{G}_{r-1}(n_{r-1})$. We want to analyze the communication cost and probability of success of this protocol. We start by getting an upper bound on $s_r(n)/p_r$.

$$\begin{aligned}
\frac{s_r(n)}{p_r} &= \underbrace{\frac{(n) \uparrow \left(1 + \frac{1}{2^r - 1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot e^{3r \ln^{5/6}(n)}}}_{=s_r} \cdot \underbrace{\frac{\exp\left(\eta_p \cdot (r \ln 2 + \ln(b_r))^{3/4}\right)}{b_r \cdot 2^r}}_{=1/p_r} \\
&= \frac{2^{\frac{2^r}{2^r-1}} \cdot (n_{r-1}) \uparrow \left(\frac{2^r}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \frac{1}{b_r} \cdot \exp\left(\eta_p \cdot (r \ln 2 + \ln(b_r))^{3/4} - 3r \ln^{5/6}(n)\right) \\
&\text{(by the choice of } n_{r-1} \text{ from Eq (5.16) and moving around the terms } 2^r \text{ and } b_r \text{ in the denominator)} \\
&= \frac{2^{\frac{2^r}{2^r-1}} \cdot (n_{r-1}) \uparrow \left(\frac{2^r}{2^{r-1}-1} - \frac{2^{r-1}}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(\eta_p \cdot (r \ln 2 + \ln(b_r))^{3/4} - 3r \ln^{5/6}(n)\right) \\
&\text{(by the choice of } b_r \text{ from Eq (5.16) and writing it in the exponent of } n_{r-1}) \\
&= \frac{2^{\frac{2^r}{2^r-1}} \cdot (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(\eta_p \cdot (r \ln 2 + \ln(b_r))^{3/4} - 3r \ln^{5/6}(n)\right) \\
&\text{(by canceling the terms in the exponent of } n_{r-1}) \\
&\leq \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(O(\ln^{3/4}(n)) - 3r \ln^{5/6}(n)\right) \\
&\text{(as } 2^{\frac{2^r}{2^r-1}} = O(1), \ b_r \leq n \text{ by Eq (5.16)}, \ \eta_p = \Theta(1) \text{ by Proposition 5.4.5, and } r \ll \ln n \text{ by Eq (5.3))} \\
&\leq \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(-(3r-1) \ln^{5/6}(n)\right). \\
&\text{(as } O(\log^{3/4}(n)) < \ln^{5/6}(n_r) \text{ for large } n)
\end{aligned}$$

By Corollary 5.7.3, the communication cost of π_{r-1} is at most,

$$\begin{aligned}
\mathrm{CC}(\pi_{r-1}) &\leq \frac{c_{\mathrm{comp}}}{\varepsilon} \cdot \left(\frac{s_r(n)}{p_r} + r^2 \right) \\
&\leq \frac{c_{\mathrm{comp}}}{\varepsilon} \cdot \left(\frac{2 \cdot (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(-(3r-1)\ln^{5/6}(n)\right) \right) \\
&\text{(by our bound on } s_r(n)/p_r \text{ from above and since } r^2 \text{ is (much) smaller than this bound by Eq (5.3))} \\
&= c_{\mathrm{comp}} \cdot \underbrace{2^{p_0} \cdot 40 \cdot (r+1)^2}_{=1/\varepsilon} \cdot \left(\frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^{r-1}} \cdot \exp\left(-(3r-2)\ln^{5/6}(n)\right) \right) \\
&\text{(by our choice of } \varepsilon \text{ and canceling the 2-term in the nominator via } 2^r \text{ in the denominator)} \\
&= \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^{r-1} \cdot (r!)^2 \cdot 2^{r-1}} \cdot \exp\left(-(3r-1)\ln^{5/6}(n)\right) \\
&\text{(by canceling the extra terms via the ones in the denominator)} \\
&\leq \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^{r-1} \cdot (r!)^2 \cdot 2^{r-1} \cdot e^{3 \cdot (r-1) \cdot \ln^{5/6}(n_{r-1})}} \\
&\text{(by dropping an extra } e^{2\ln^{5/6}(n_{r-1})} \text{-term and using the fact } n_{r-1} < n) \\
&= s_{r-1}(n_{r-1}).
\end{aligned}$$

Thus, the communication cost of π_{r-1} is less than the bounds of the induction hypothesis for $r-1$.

It remains to bound the probability of success of π_{r-1}. By Corollary 5.7.3,

$$\begin{aligned}
\Pr\left(\pi_{r-1} \text{ solves } \mathbb{P}_{r-1}\right) &\geq \delta - \varepsilon - \sqrt{\frac{s_r(n)}{p_r \cdot q_r}} \\
&\geq \frac{1}{2^{p_0}} \left(1 + \frac{r}{20(r+1)}\right) - \frac{1}{2^{p_0} \cdot 40(r+1)^2} - \sqrt{\frac{s_r(n)}{p_r \cdot q_r}}. \\
&\text{(by the choice of } \varepsilon \text{ and } \delta \text{ earlier in the proof)}
\end{aligned}$$

We thus only need to bound the last term to conclude the proof.

We can upper bound this last term as follows:

$$\frac{s_r(n)}{p_r \cdot q_r} \leq \frac{(n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\mathrm{comp}})^r \cdot ((r+1)!)^2 \cdot 2^r} \cdot \exp\left(-(3r-1)\ln^{5/6}(n)\right) \cdot \underbrace{\frac{\exp\left(\eta_q \cdot (r \ln 2 + \ln b_r)^{3/4}\right)}{b_r \cdot 2^r}}_{=1/q_r}$$

(by our upper bound on $s_r(n)/p_r$ established above and the choice of q_r in Eq (5.16))

$$= \frac{\exp\left(-(3r-1)\ln^{5/6}(n) + \eta_q \cdot (r\ln 2 + \ln b_r)^{3/4}\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot 2^{2r}}$$

(by the choice of $b_r = (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1}-1}\right)$ in Eq (5.16) and moving around the terms)

$$\leq \frac{\exp\left(-(3r-1)\ln^{5/6}(n) + O(\ln^{3/4} n)\right)}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot 2^{2r}}$$

(as $b_r \leq n$ by Eq (5.16), $\eta_q = \Theta(1)$ by Proposition 5.4.5, and $r \ll \ln n$ by Eq (5.3))

$$\leq \frac{\exp(-(3r-2)\ln^{5/6}(n))}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot 2^{2r}}. \qquad \text{(as } O(\ln^{3/4}(n)) < \log^{5/6}(n) \text{ for large } n)$$

$$\ll \left(\frac{1}{2^{p_0} \cdot 40 \cdot (r+1)^2}\right)^2.$$

(using a crude upper bound sufficient for the analysis using $\ln^{5/6} \gg r$ by Eq (5.3))

Thus, the probability that π_{r-1} solves $\mathbb{P}_{r-1}$ is at least

$$\frac{1}{2^{p_0}}\left(1 + \frac{r}{20(r+1)}\right) - \frac{1}{2^{p_0} \cdot 40(r+1)^2} - \frac{1}{2^{p_0} \cdot 40(r+1)^2} > \frac{1}{2^{p_0}}\left(1 + \frac{r-1}{20r}\right).$$

But, now π_{r-1} is a $(r-1)$-round protocol with communication cost $\text{CC}(\pi_{r-1}) < s_{r-1}(n_{r-1})$ and probability of success strictly larger than $1/2^{p_0} \cdot (1 + (r-1)/20r)$ which contradicts the induction hypothesis for $r-1$ (recall that by Observation 5.5.3, n_{r-1} and $(r-1)$ also satisfy Eq (5.3) and thus we can indeed apply the induction hypothesis). This implies that our choice of the protocol π_r was contradictory, completing the proof. ∎

5.7.2 Concluding the Proof of Theorem 9

Theorem 9 follows from combining the results in the previous section and the following argument.

Lemma 5.5.10 implies that $\mathbf{MIS}_{n,r+1}$ is as hard as solving $\mathbb{P}_r(G, K)$ for $G \sim \mathcal{G}_r$ and every valid search sequence K. Lemma 5.7.1 then shows that, for the r-round protocol π in the theorem statement, there is a lower bound of

$$\text{CC}(\pi) \geq s_r(n) := \frac{1}{n_0^2 \cdot (2^{p_0} \cdot 40 c_{\text{comp}})^r \cdot ((r+1)!)^2 \cdot e^{3r\ln^{5/6}(n)}} \cdot \left((n) \uparrow \left(1 + \frac{1}{2^r - 1}\right)\right)$$

for solving $\mathbb{P}_r$ on an n-vertex input graph $G \sim \mathcal{G}_r(n)$ with probability of success at least

$$\frac{1}{2^{p_0}}\left(1 + \frac{r}{20(r+1)}\right) \leq \frac{21}{20 \cdot 2^{p_0}} < \frac{21}{20 \cdot 2^{(n_0/2)-1}}$$

for any $r \geq 1$. We choose n_0 to be a large enough constant to handle any constant probability

of success strictly larger than zero. Thus, for $\mathbf{MIS}_{n,r+1}$, we obtain a lower bound of

$$\Omega\left(\frac{1}{2^{\Theta(r\ln^{5/6}(n))}}\cdot n^{1+1/(2^r-1)}\right)$$

bits on the communication cost of the problem for any r-round protocol as long as $n \geq (n_0)^{2^r-1}$.

This concludes the proof of Theorem 9. ∎

6 Matchings in Streams

The maximum matching problem is arguably the most studied problem in the semi-streaming model and has been considered from numerous angles (this list is by no means a comprehensive summary of prior results):

- single-pass algorithms [FKM+05, GKK12, Kap13, Kap21, ABKL23],
- constant-pass algorithms [KMM12, EHM16, KT17, Kon18b, KN21, Ass22, KNS23],
- $(1-\varepsilon)$-approximation algorithms [McG05, AG11, EKMS12, AG18, Tir18, GKMS19, ALT21, FMU22, AJJ+22, Ass23],
- random-order streams [KMM12, Kon18b, ABB+19, GKMS19, FHM+20, Ber20, AS23b],
- dynamic streams [Kon15, CCHM15, AKLY16, CCE+16, AKL17, DK20, AS22],
- weighted or submodular matchings [FKM+05, CS14, CK14, CGQ15, PS17, BDL21, LW21],
- matching size estimation [KKS14, EHL+15, BS15, MV16, CJMM17, MV18, AKL17, KMNT20, AKSY20, AN21, AS23b],
- and, exact algorithms [FKM+05, GO16, AR20, LSZ20, CKP+21a, AJJ+22].

In this chapter, we focus on proving multi-pass graph streaming lower bounds for approximate matchings in both insertion-only streams and dynamic streams.

Insertion-only Streams

The question of understanding the approximation ratio achievable by multi-pass semi-streaming algorithms for matchings was posed by [FKM+05] alongside the introduction of the semi-streaming model itself. Moreover, [FKM+05] also gave a $(2/3-\varepsilon)$-approximation algorithm for this problem in $O(1/\varepsilon)$ passes, which was soon after improved by [McG05] to a $(1-\varepsilon)$-approximation in $(1/\varepsilon)^{O(1/\varepsilon)}$ passes. A long line of work since then [AG11, KMM12, Kap13, EKMS12, KT17, AG18, Kon18b, Tir18, ALT21, FMU22, AJJ+22] has culminated in semi-streaming algorithms with $\text{poly}(1/\varepsilon)$ passes for general graphs [FMU22] and $O(1/\varepsilon^2)$ passes for bipartite graphs [ALT21] (there are also algorithms with pass-complexity with better dependence on ε at the cost of mild dependence on n, namely, $O(\log n/\varepsilon)$ passes in [AG18, AJJ+22, Ass23] or for finding perfect matchings in $n^{3/4+o(1)}$ passes [AJJ+22]; see also [LSZ20]).

The lower bound front however has seen much less progress with only a handful of results known for single-pass algorithms [GKK12, Kap13, AKL17, Kap21] and very recently two-pass algorithms [KN21, Ass22, KN24]. But no lower bounds beyond two-pass algorithms are known for constant factor approximation of matchings in the semi-streaming model (lower bounds for computing exact maximum matchings up to almost $\Omega(\log n)$ passes are proven in [GO16];

see also [AR20, CKP+21a]). It is worth noting that in the much more restricted setting of polylog(n)-space algorithms, [AN21], building on [AKSY20], proved an $\Omega(1/\varepsilon)$-pass lower bound for estimating the matching size.

Let $\beta \in (0,1)$ is the largest parameter so that an n-vertex graph with n^{β} edge-disjoint induced matchings of size $\Theta(n)$ exist (see Section 4.2.1). This is our first result:

Theorem 2 (restated). *Any (possibly randomized) semi-streaming algorithm for $(1-\varepsilon)$-approximation of even the size of maximum matchings requires*

$$\Omega\Big(\frac{\log(1/\varepsilon)}{\log(1/\beta_{RS})}\Big)$$

passes over the stream. The lower bound holds for the entire range of $\varepsilon \in [n^{-\Theta(\beta_{RS})}, \Theta(\beta_{RS})]$.

The current known values of β_{RS} are:

$$\Omega\Big(\frac{1}{\log\log n}\Big) \underset{\text{[FLN+02]}}{\leq} \beta_{\text{RS}} \underset{\text{[FHS15]}}{\leq} 1-\Theta\Big(\frac{\log^* n}{\log n}\Big),$$

so in the case where $\beta_{\text{RS}} = \Omega(1)$, our result gives an $\Omega(\log(1/\varepsilon))$-lower bound on the number of passes. More recently, we proved the same result unconditionally for outputting a $(1-\varepsilon)$-approximate matching.

Theorem 3 (restated). *Any (possibly randomized) semi-streaming algorithm that outputs a $(1-\varepsilon)$-approximation of maximum matchings requires $\Omega(\log(1/\varepsilon))$-passes over the stream.*

Our lower bound result provides the first **pass-approximation** lower bound for (small) **constant approximation** of matchings in the semi-streaming model, a longstanding open question in the graph streaming literature.

The techniques of Theorem 2 are based on analyzing communication protocols for compressing (hidden) permutations. Prior work in this context [CKP+21a] relied on reducing such problems to Boolean domain and analyzing them via tools like XOR Lemmas and Fourier analysis on Boolean hypercube. In contrast, our technical contribution is a hardness amplification result for permutations through concatenation in place of prior XOR Lemmas. This result is proven by analyzing permutations directly via simple tools from group representation theory [HGG09] combined with detailed information-theoretic arguments. We not go further into the proof of Theorem 2, as the techniques are mostly disjoint from the rest of this thesis. We only focus on Theorem 3 in this chapter.

Dynamic Streams

In the **dynamic graph streaming** model, the edges in E are defined by a sequence of insertions and deletions in a stream $\sigma := (\sigma_1, \ldots, \sigma_N)$ of length N which is often assumed to be some poly(n). Each entry σ_i of the stream is either inserting a *new* edge (u_i, v_i) to E or deleting an *already inserted* edge from it. The goal is to make one or a few passes over the stream, use a limited memory—ideally, $O(n \cdot \text{polylog}(n))$ bits, same as **semi-streaming space**—and compute the answer to a given problem on the graph G at the end of the last pass.

- The first such algorithms for matchings were obtained in [AGM12] alongside the introduction of the dynamic graph streaming model itself. The authors in [AGM12] observed that the prior techniques of [LMSV11] (in the MapReduce/MPC model) also imply an $O(\log n)$-pass semi-streaming algorithm for 2-approximation of maximum matching in dynamic streams.[1]
- In the same work, [AGM12], building on [AG11], further improved the approximation ratio of the algorithm of [LMSV11] to $(1+\varepsilon)$-approximation for any *fixed* $\varepsilon > 0$ using $O(\log^2(n))$ passes on general graphs and $O(\log n)$ passes on bipartite graphs.
- The algorithms of [AGM12] were subsequently improved in [AG15] to an $O(\log n)$-pass algorithm for $(1+\varepsilon)$-approximation even on weighted (general) graphs. Very recently, this algorithm was simplified and slightly improved in [Ass24]. Yet another algorithm with similar guarantees for unweighted bipartite graphs was obtained in [AJJ+22].
- In addition, some generic reductions from general to bipartite [McG05, Tir18] or weighted to unweighted matchings [GKMS19, BDL21] developed over the years can be applied to the algorithms of [LMSV11] to obtain $O(\log n)$-pass algorithms for $(1+\varepsilon)$-approximation for fixed $\varepsilon > 0$.[2]
- In parallel to the line of work on multi-pass algorithms, a series of work studied single-pass algorithms for this problem [Kon15, CCHM15, AKLY16, CCE+16, AKL17, DK20, AS22]. In particular, [AKLY16] proved that any $O(1)$-approximation of matchings via single-pass algorithms requires $n^{2-o(1)}$ space and [DK20] improved this to an optimal $\Omega(n^2)$ space lower bound.

This constitutes the state-of-the-art for matchings in dynamic graph streams:

> *For semi-streaming algorithms on dynamic streams, $O(1)$-approximation to maximum matching is possible in $\approx \log n$ passes and not possible in a single pass.*

Closing this huge gap between upper and lower bounds for dynamic streaming matchings has been a longstanding open question in the graph streaming literature. This is precisely the contribution of our work: we fully settle the pass complexity of $O(1)$-approximation of maximum matching in dynamic streams by improving **both** the upper and lower bounds for this problem. This thesis only focuses on our lower bound result.

Theorem 4 (restated). *Any randomized semi-streaming algorithm for $O(1)$-approximation of maximum matching in dynamic streams with constant probability of success requires $\Omega(\log \log n)$ passes. The lower bound holds even on (unweighted) bipartite graphs.*

Let us move to the proof of our results.

[1] Technically, the algorithms of [LMSV11, AGM12], and some subsequent ones, use $n^{1+1/p}$-space in $O(p)$ passes. This translates to an $O(\frac{\log(n)}{\log\log(n)})$-pass algorithm in semi-streaming space. Yet, to keep the focus on the bigger picture, we ignore this lower-order term improvement and refer to these algorithms as $O(\log n)$ passes still.

[2] These algorithms generally have a (much) worse dependence on the parameter ε compared to the ones in the bullet point above, but for constant $\varepsilon > 0$, their guarantees are still asymptotically the same.

6.1 Insertion-only Streams

Our multi-pass lower bound for approximate maximum matching follows from the ideas of [KN24] and the optimal multi-pass lower bound for maximal independent set in Chapter 5. We reuse the preliminaries section of Maximal Independent Sets for the insertion-only streaming lower bound also (the definitions and tools from Section 5.3). The new round elimination theorem Theorem 10 of Chapter 5 will be used to prove the statement. As this result is quite short, and the techniques were already described in Chapter 3, we skip a more detailed overview and move directly to the proof.

Using the connection between communication complexity and streaming (see Proposition 5.3.2), our result is implied by a p-round communication lower bound for approximate matching. To that end, we construct a hard distribution for p-round protocols by using a graph product that *embeds* "many" $(p-1)$-round hard instances into an RS graph as in [KN24]. This combinatorial construction is the only new technical contribution of this result.

Now, we state the communication lower bound that we prove in this section.

Theorem 11. *For any $p \in \mathbb{N}$, $0 < \varepsilon \leq 1$, and sufficiently large integer $n \geq 1$, any p-round $(p+1)$-player protocol π that outputs an $(1-\varepsilon)$-approximate maximum matching with*

$$\varepsilon < \frac{1}{4} \cdot \left(\frac{1-2\delta_{\text{rs}}}{3+2\delta_{\text{rs}}}\right)^{p},$$

where δ_{rs} is the constant from Proposition 4.2.2, and has at least $(2/3)$ probability of success has communication cost

$$CC(\pi) = n^{1+\Omega\left(1/(\log\log n)^{3p}\right)}.$$

We can easily prove Theorem 3 now.

Proof of Theorem 3. Let $\mathcal{A}$ be a semi-streaming algorithm that gets an $(1-\varepsilon)$-approximate maximum matching in p-passes and $\widetilde{O}(n)$ space with probability of success at least $(2/3)$. By the connection between streaming algorithms and communication protocols Proposition 5.3.2, $\mathcal{A}$ gives rise to a protocol for approximate matching with the same guarantees.

Using Theorem 11, we know that the space required by such an algorithm is at least

$$n^{1+\Omega\left(1/(\log\log n)^{3p}\right)} = \omega(n \log^{c} n),$$

for any constant c when $p = o(\log\log n / \log\log\log n)$, which is more than the space used by $\mathcal{A}$. Therefore, the approximation ratio of such an algorithm should be:

$$\varepsilon < \frac{1}{4} \cdot \left(\frac{1-2\delta_{\text{rs}}}{3+2\delta_{\text{rs}}}\right)^{p} \implies p = \Omega(\log(1/\varepsilon)),$$

for constant δ_{rs} from Proposition 4.2.2. ∎

We prove Theorem 11 by first recursively constructing a family of distributions $\{\mathcal{G}_p\}_{p\geq 0}$ of graphs where each $\mathcal{G}_p$ is a hard distribution for p-round protocols that output a matching

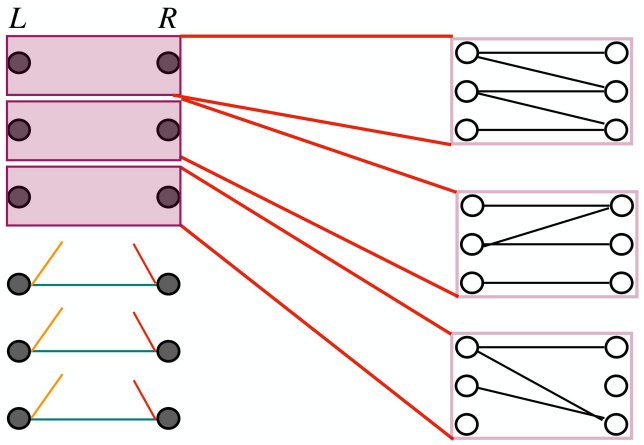


Figure 6.1: An example of the embedding product on RS graphs. The RS graph from Figure 3.7 is used and there are totally 12 many smaller graphs. We show three such graphs on the right. The shaded circles indicate vertices which are groups of three. Each edge is replaced by a specific bipartite graph from the given family.

with an approximation ratio larger than $(1 - 1/3^p)$ (see Section 6.1). Then, we show that Theorem 10 is applicable to directly prove the result.

Before describing the hard distributions, we first provide a useful embedding of a family of $r \cdot t$ many bipartite graphs within an (r, t)-RS graph. This can be seen as replacing every edge of the "outer" RS graph with one of the "inner" graphs of the family[3].

Definition 6.1.1. *Let G_{rs} be a balanced bipartite (r, t)-RS graph on $2n_1$ vertices $L_{\text{rs}} \cup R_{\text{rs}}$ for some $r, t \geq 1$. Let $\mathcal{H} := \{H_{i,j} \mid i \in [t], j \in [r]\}$ be a family of balanced bipartite **inner graphs** where all graphs in $\mathcal{H}$ are on the same $2n_2$ vertices $L_{\text{in}} \cup R_{\text{in}}$, but may have different edges.*

*We define the **embedding of $\mathcal{H}$ into** G_{rs}, denoted by $G := \text{EMBED}(\mathcal{H} \to G_{\text{rs}})$, as the following $(2 \cdot n_1 \cdot n_2)$-vertex balanced bipartite graph:*

- ***Vertices:** We have vertex sets $L_{\text{em}} := L_{\text{rs}} \times L_{\text{in}}$ and $R_{\text{em}} := R_{\text{rs}} \times R_{\text{in}}$;*
- ***Edges:** For any $i \in [t]$ and $j \in [r]$, let $e_{i,j} = (u, v)$ be the j^{th} edge of induced matching M_i^{rs} in G_{rs}. Then, for any edge $(w_L, w_R) \in H_{i,j}$, we add an edge f between vertices $(u, w_L) \in L_{\text{em}}$ and $(v, w_R) \in R_{\text{em}}$ to G. We say that the edge f is added with respect to the edge $e_{i,j}$.*

See Figure 6.1 for an illustration.

The following lemma captures the main property of these embeddings, which crucially relies on the matchings in a RS graph being induced and edge disjoint.

Lemma 6.1.2. *Let G_{rs} be a balanced bipartite (r, t)-RS graph for some $r, t \geq 1$, let*

$$\mathcal{H} := \{H_{i,j} \mid i \in [t], j \in [r]\}$$

[3]RS graphs are the counterparts of graphs with many disjoint unique path collections from Chapter 5.

be a family of balanced bipartite graphs and let $G := \text{EMBED}(\mathcal{H} \to G_{\text{rs}})$. *For any* $i \in [t]$, *the* **induced subgraph** *of* G *on vertices corresponding to* M_i^{rs}, *i.e.,* $\{(v, *) \mid v \in e \text{ for some } e \in M_i^{\text{rs}}\}$, *is a* **vertex-disjoint union** *of graphs* $H_{i,j}$ *for* $j \in [r]$ *in* $\mathcal{H}$.

Proof. For any $i \in [t]$, let $V_i := \{(v, *) \mid v \in e_{i,j} \text{ for some } e_{i,j} \in M_i^{\text{rs}}\}$ denote the vertices of G corresponding to the induced matching M_i^{rs}. Let $G_i := G[V_i]$ be the induced subgraph of G on V_i. We prove the result in two parts: (i) G_i contains the disjoint union of $H_{i,j}$ for $j \in [r]$, and (ii) it does not contain any other edges.

Part (i). Fix $j \in [r]$ and observe that all edges of $H_{i,j}$ are added to G_i by the embedding product with respect to the edge $e_{i,j} = (u, v)$ in a way that "preserves its structure". Formally, the bipartite edges of $H_{i,j}$ between vertices L_{in} and R_{in} are added, with a one-to-one correspondence, between the vertices $\{u\} \times L_{\text{in}} \subseteq L_{\text{em}}$ and $\{v\} \times R_{\text{in}} \subseteq R_{\text{em}}$ of G_i. Furthermore, since the edges of the matching M_i^{rs} are vertex-disjoint, the edges added to G with respect to the edges of M_i^{rs} form a vertex-disjoint union of $H_{i,j}$ for all $j \in [r]$ in G_i.

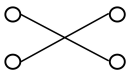
Part (ii). It remains to argue that no other edges are added to G_i by the embedding product. Towards a contradiction, assume that G_i contains an edge f that is *not* added with respect to an edge of M_i^{rs}. This implies that f is added with respect to an edge $e_{i',j'} = (u', v')$ of G_{rs} for some $i' \neq i \in [t]$ and $j' \in [r]$, and that u', v' must each be an endpoint of edges in M_i^{rs}. This, however, is a contradiction since $e_{i',j'}$ is an edge of $M_{i'}^{\text{rs}}$ that is edge disjoint from M_i^{rs}, and M_i^{rs} is an induced matching. ∎

As previously mentioned, the distributions are defined recursively as a family $\{\mathcal{G}_p\}_{p \geq 0}$ where each $\mathcal{G}_p$ is a hard distribution for p-round $(p+1)$-player protocols. We use $Q_1, Q_2, \ldots, Q_{p+1}$ to denote the players in $\mathcal{G}_p$. In addition, for any input graph G and any $a \in [p+1]$, the input to player Q_a is denoted by $G(a)$.

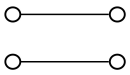
Base case for $p = 0$. The base case is defined for $p = 0$ such that protocols with 0 rounds of communication are required to solve a simple yet non-trivial matching problem (and is hence trivially impossible). Formally, the distribution is as follows and is illustrated in Figure 6.2:

The distribution $\mathcal{G}_0(n)$ for 0-round "protocols".

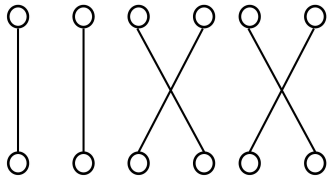
1. Let G be a balanced bipartite graph on $2n$ vertices. G is made of $r_0 = (n/2)$ vertex disjoint gadgets, each defined as follows:
   (a) Let $H = (L, R, E)$ be a gadget on vertices $L = \{u_1, u_2\}$ and $R = \{v_1, v_2\}$
   (b) With probability $1/2$, the edges (u_1, v_1) and (u_2, v_2) are included in E (a parallel gadget). Otherwise, the edges (u_1, v_2) and (u_2, v_1) are included in E (a cross gadget).
2. Player Q_1 receives $G(1) := G$ as the input (this is the only player when $p = 0$).



(a) A cross gadget.



(b) A parallel gadget.



(c) An example of a graph sampled from $\mathcal{G}_0(n)$ with $n = 6$, two cross gadgets and one parallel gadget.

Figure 6.2: The hard distribution $\mathcal{G}_0(n)$ and the gadgets.

We can immediately see that any graph sampled from $\mathcal{G}_0$ is made up only of a single perfect matching of size $2 \cdot r_0 = n$. Learning a single edge of a gadget reveals the type of gadget, i.e., whether it is a parallel gadget or cross gadget, and thus also reveals the other edge. Furthermore, to learn the type of a gadget, any 0-round "protocol", which has to commit to an answer without knowing the input, can *not* do better that random guessing.

Distributions for $p \geq 1$. The hard distribution $\mathcal{G}_p$ for p-rounds will be a $(2n)$-vertex balanced bipartite graph constructed by *embedding* "many" hard instances for $(p-1)$-rounds sampled from $\mathcal{G}_{p-1}(n_{p-1})$ into a balanced bipartite (r_p, t_p)-RS graph with $2b_p$ vertices (see Eq (6.1) and Eq (6.2) for the parameters). The distribution is as follows and is illustrated in Figure 6.3:

The distribution $\mathcal{G}_p(n)$ for p-round protocols for $p \geq 1$ and sufficiently large $n \in \mathbb{N}$.

1. Pick a fixed balanced bipartite (r_p, t_p)-RS graph G_{rs} on $2b_p$ vertices $L_{\text{rs}} \cup R_{\text{rs}}$, from Proposition 4.2.2.
2. Sample $r_p \cdot t_p$ independent instances from $\mathcal{G}_{p-1}(n_{p-1})$ denoted by $\mathcal{H} := \{H_{i,j} \mid i \in [t_p], j \in [r_p]\}$.
3. Create the graph $\widetilde{G} := \textsc{embed}(\mathcal{H} \to G_{\text{rs}})$ on vertices $L_{\text{em}} \cup R_{\text{em}}$ (as in Definition 5.4.6).
4. For $a \in [p]$, player Q_a receives the edges in $\widetilde{G}$ that correspond to subgraphs $H_{i,j}(a)$ for all $i \in [t_p]$ and $j \in [r_p]$ where $H_{i,j}(a)$ is the subgraph of $H_{i,j}$ given to the a^{th} player in $\mathcal{G}_{p-1}(n_{p-1})$.
5. Pick $i^\star$ uniformly at random from $[t_p]$ and let $M^{\text{rs}}_{i^\star}$ in G_{rs} be the *special matching*. Let $V^\star \subseteq L_{\text{rs}} \cup R_{\text{rs}}$ be the set of vertices in G_{rs} incident to $M^{\text{rs}}_{i^\star}$.
6. Player Q_{p+1} receives an arbitrary auxiliary perfect matching M^{aux} from $\{(u, *) \mid u \in (L_{\text{em}} \cup R_{\text{em}}) \setminus V^\star\}$ to new sets of vertices L_{aux} and R_{aux} (each of size

$(b_p - r_p) \cdot n_{p-1})$.

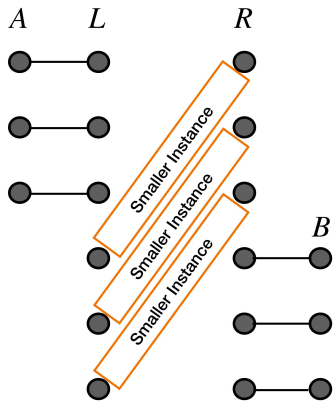


Figure 6.3: An illustration of the distribution $\mathcal{G}_p$ for $p \geq 1$. When $p = 1$, the smaller instances are as in Figure 6.2. Only the smaller instances in the special matching are shown. All edges of the outer RS graph are replaced also. The shaded circles represent groups of vertices.

The parameters n_{p-1} and b_p are chosen such that:

$$n = n_{p-1} \cdot (2b_p - r_p), \qquad b_p = (n_{p-1}) \uparrow ((\log \log n_{p-1})^3). \tag{6.1}$$

Using the family of RS graph in Proposition 4.2.2 for the construction of the distribution $\mathcal{G}_p(n)$, we get the following parameters for $p \geq 1$:

$$n = \left(\frac{3}{2} + \delta_{\text{rs}}\right) \cdot b_p \cdot n_{p-1}, \qquad r_p := \left(\frac{1}{2} - \delta_{\text{rs}}\right) \cdot b_p, \qquad t_p := b_p^{c_{\text{rs}}/\log\log b_p}. \tag{6.2}$$

Lastly we define a new parameter κ_{rs} that will be used repeatedly in the calculations.

$$\kappa_{\text{rs}} = \frac{1 - 2\delta_{\text{rs}}}{3 + 2\delta_{\text{rs}}}. \tag{6.3}$$

Notation. For any $a \in [p+1]$, let $\mathcal{G}_{p,a}(n)$ denote the distribution of edges in the input graph that are given to player Q_a. The distribution $\mathcal{G}_p(n)$ is a joint distribution of $(\mathcal{G}_{p,1}(n), \ldots, \mathcal{G}_{p,p+1}(n))$.

An instance sampled from $\mathcal{G}_p(n)$ is a graph G together with its edge partitioning (G_1, …, G_{p+1}) between the $(p+1)$ players. With a slight abuse of notation, we sometimes refer to G sampled from $\mathcal{G}_p(n)$, i.e., $G \sim \mathcal{G}_p(n)$, as the entire instance where the edge partitioning between the players is implicit. Any $G \sim \mathcal{G}_p(n)$ contains $r_p \cdot t_p$ instances $H_{1,1}, \ldots, H_{t_p,r_p}$ sampled from $\mathcal{G}_{p-1}(n_{p-1})$ that are embedded into G_{rs} to give $\widetilde{G}$. We refer to these $(p-1)$-round instances as the sub-instances of G. For any $i \in [t_p], j \in [r_p]$, we further write $\widetilde{E}_{i,j}$ to denote the edges inserted to $\widetilde{G}$ as part of the embedding of $H_{i,j}$ into G_{rs}.

Among the sub-instances of any $G \sim \mathcal{G}_p(n)$, there are r_p sub-instances that correspond

to the edges in the special matching $M^{\text{rs}}_{i^\star}$; we denote them by $\mathcal{H}^\star$ and refer to them as the **special sub-instances or subgraphs** of G.

Any $G \in \mathcal{G}_p(n)$ has r_p many special subgraphs. Each special subgraphs is an instance sampled from $\mathcal{G}_{p-1}(n_{p-1})$, which has its own corresponding r_{p-1} many special subgraphs that are instances sampled from $\mathcal{G}_{p-2}(n_{p-2})$. Continuing this gives us $r_p \cdot r_{p-1} \cdot \ldots \cdot r_1$ many special subgraphs sampled from $\mathcal{G}_0$, which is each made up of r_0 many (either parallel or cross) gadgets with 2 vertex disjoint edges each, i.e., a matching of size 2. We define these gadgets/matchings as the **ultimate gadgets**

$$\mathcal{M}^\star = \{M^\star_{j_p, j_{p-1}, \ldots j_0} : j_p \in [r_p], j_{p-1} \in [r_{p-1}], \ldots, j_0 \in [r_0]\}$$

whose union we denote as $M^\star$ and call the **ultimate induced matching** (see Property (6.1.5) below).

We need some key properties about any input graph G sampled from $\mathcal{G}_p(n)$ for $p \geq 1$.

Property 6.1.3. *For any $p \geq 0$, the distribution $\mathcal{G}_p(n)$ is a product distribution over the $(p+1)$-players.*

Proof. When $p = 0$, there is only one player, and $\mathcal{G}_0(n)$ is a product distribution trivially. Let us assume that $\mathcal{G}_{p-1}(n_{p-1})$ is a product distribution for the induction hypothesis.

In distribution $\mathcal{G}_p(n)$, we know that the input of Q_{p+1} is independent of the input of all the other players, as they receive one induced matching $i^\star$ from $[t_p]$ uniformly at random. The $r_p \cdot t_p$ instances from $\mathcal{G}_{p-1}(n_{p-1})$ are given to the other players, and they are sampled independently of each other. As distribution $\mathcal{G}_{p-1}(n_{p-1})$ is a product distribution, the input of these players is independent of each other. ∎

Property 6.1.4. *The special subgraphs of G are all vertex-disjoint. Furthermore, the subgraph of G induced by the vertices corresponding to the special subgraphs consists only of the edges of these subgraphs, i.e., the edges of $G^\star_j$ for all $j \in [r_p]$.*

Proof. Recall that, by definition, the vertices corresponding to the special subgraphs are the vertices of G that correspond to the matching $M^{\text{rs}}_{i^\star}$, i.e., $\{(u, *) \mid u \in V^\star\}$. Let $G^\star$ be the subgraph induced by these vertices.

We have by Lemma 6.1.2 that the edges of $G^\star \cap \widetilde{G}$ are exactly the edges of a vertex-disjoint union of the special subgraphs $H_{i^\star, j}$ for $j \in [r_p]$. Finally, $G^\star \cap \widetilde{G} = G^\star$ since the only edges not in $\widetilde{G}$ are the edges of the auxiliary matching M^{aux}, each of which is incident to at most a single vertex of $\{(u, *) \mid u \in V^\star\}$ and thus can not be in $G^\star$. ∎

We can extend this property to all the ultimate small gadgets also.

Property 6.1.5. *The ultimate gadgets $\mathcal{M}^\star$ in G are vertex-disjoint and their union forms an induced matching $M^\star$, namely, the ultimate induced matching, of size $(\kappa_{\text{rs}})^p \cdot n$.*

Proof. We first show by induction that the special gadgets $\mathcal{M}^\star$ are vertex-disjoint and that $M^\star$ is an induced matching of size $2 \cdot r_p \cdot r_{p-1} \cdot \ldots \cdot r_0$.

Base Case ($\mathcal{G}_0$). By construction, the only edges of any graph $G \sim \mathcal{G}_0$ are exactly the r_0 many vertex-disjoint gadgets that make up G, which are the ultimate gadgets of $\mathcal{M}^\star$. Furthermore, each gadget is made up of 2 vertex-disjoint edges and thus their union is an induced matching $M^\star$ of size $2 \cdot r_0$.

Inductive Step ($\mathcal{G}_p$). For the induction hypothesis, assume that the statement holds for graphs sampled from $\mathcal{G}_{p-1}(n_{p-1})$. By definition, any graph $G \sim \mathcal{G}_p(n)$ has special subgraphs $H_{i^\star,j} \sim \mathcal{G}_{p-1}(n_{p-1})$ for $j \in [r_p]$. By the induction hypothesis, the union of the ultimate gadgets $\mathcal{M}_j^\star$ of each $H_{i^\star,j}$ form an induced matching $M_j^\star$ of size $2 \cdot r_{p-1} \cdot \ldots \cdot r_0$. We also have that the union of the special gadgets $\mathcal{M}^\star$ in G is $M^\star = \bigcup_{j \in [r_p]} M_j^\star$.

The vertices of $M_j^\star$ and $M_{j'}^\star$ are disjoint by Property (6.1.4) for $j \neq j'$, $j, j' \in [r_p]$. As each $M_j^\star$ is an induced matching, their union $M^\star$ is an induced matching in G of size $r_p \cdot (2 \cdot r_{p-1} \cdot \ldots \cdot r_0)$.

Finally, using the parameters in Eq (6.1) and Eq (6.2), we have that

$$\begin{aligned}
|M^\star| &= 2 \cdot r_p \cdot r_{p-1} \cdot \ldots \cdot r_0 \\
&= (1/2 - \delta_{\mathrm{rs}})^p \cdot (b_p \cdot \ldots \cdot b_1) \cdot n_0 && (\text{since } 2 \cdot r_0 = n_0) \\
&= (1/2 - \delta_{\mathrm{rs}})^p (b_p \cdot \ldots \cdot b_2) \cdot (3/2 + \delta_{\mathrm{rs}})^{-1} \cdot n_1 && (\text{as } n_1 = (3/2 + \delta_{\mathrm{rs}}) \cdot n_0 \cdot b_1) \\
&= (1/2 - \delta_{\mathrm{rs}})^p (3/2 + \delta_{\mathrm{rs}})^{-p} \cdot n && (\text{repeating the above two steps } (p-1) \text{ more times}) \\
&= \left(\frac{1/2 - \delta_{\mathrm{rs}}}{3/2 + \delta_{\mathrm{rs}}} \right)^p \cdot n = (\kappa_{\mathrm{rs}})^p \cdot n,
\end{aligned}$$

by definition of κ_{rs} in Eq (6.3) as required. ∎

Property 6.1.6. *Any graph $G \sim \mathcal{G}_p(n)$ for $p \geq 0$ has a perfect matching of size n.*

Proof. The proof is again by induction.

Base Case ($\mathcal{G}_0$). Graph $G \sim \mathcal{G}_0(n)$ is made of $(n/2)$ gadgets on four vertices each. Each gadget has a perfect matching by construction, and the union of these matchings gives a perfect matching of G.

Inductive Step ($\mathcal{G}_p$). For the induction hypothesis, assume that the statement holds for graphs sampled from $\mathcal{G}_{p-1}(n_{p-1})$. By definition, any graph $G \sim \mathcal{G}_p(n)$ has special subgraphs $H_{i^\star,j} \sim \mathcal{G}_{p-1}(n_{p-1})$ for $j \in [r_p]$. Let M_j^{in} for $j \in [r_p]$ be the perfect matching of the inner instance $H_{i^\star,j}$. We pick the following edges in our matching.

$$M^{\mathrm{aux}} \cup \bigcup_{j \in [r_p]} M_j^{\mathrm{in}}$$

The vertices of M^{aux} cover all vertices of G not incident to any special subgraph $H_{i^\star,j}$ by construction. We also have that the subgraphs $H_{i^\star,j}$ are vertex-disjoint by Property (6.1.4). Thus, taking the union of these matchings is sufficient to give a perfect matching of G. ∎

Property 6.1.7. *Let $\overline{G}$ be the subgraph of G induced by the vertices which are* not *incident*

to the ultimate matching $M^\star$. Then, $\overline{G}$ has a vertex cover of size

$$(1-(\kappa_{\mathrm{rs}})^p)\cdot n.$$

Proof. For the base case of the induction on p, we see that *all* the vertices of $G \sim \mathcal{G}_0(n)$ are incident to some edge in $M^\star$. This is because $M^\star$ contains all the edges of all the $(n/2)$ gadgets of G. There is a vertex cover of size 0 trivially.

Inductive Step ($\mathcal{G}_p$). Now, suppose that the statement holds for graphs sampled from $\mathcal{G}_{p-1}(n_{p-1})$. Again, we look at the special subgraphs $H_{i^\star,j} \sim \mathcal{G}_{p-1}(n_{p-1})$ for $j \in [r_p]$. These graphs are vertex disjoint by Property (6.1.4), and they have a vertex cover V_j^{in} that covers all the edges not in $M_j^\star$ (the ultimate matching of $H_{i^\star,j}$). The vertex cover we pick contains:

$$\text{Vertices of } \widetilde{G} \text{ which matching } M_{i^\star}^{\mathrm{rs}} \text{ does not touch and } \bigcup_{j\in[r_p]} V_j^{\mathrm{in}}.$$

First, we argue that this subset of vertices covers the required edges. Any edge in M^{aux} is covered by the first part, as M^{aux} only incident to vertices not corresponding to $M_{i^\star}^{\mathrm{rs}}$. Any other edge in $\widetilde{G}$ which is not added because of $M_{i^\star}^{\mathrm{rs}}$ is also covered by the first part. We know V_j^{in} is a vertex cover of the subgraph of $H_{i^\star,j}$ with vertices not touched by $M_j^\star$ by the induction hypothesis. As $M^\star = \cup_{j\in[r_p]} M_j^\star$, we are done.

It remains to analyze the size of our vertex cover. By the induction hypothesis, the size is exactly:

$$\begin{aligned}
&r_p \cdot \left(1-(\kappa_{\mathrm{rs}})^{p-1}\right)\cdot n_{p-1} + \text{Number of vertices of } \widetilde{G} \text{ not touching } M_{i^\star}^{\mathrm{rs}} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(from each of the } r_p\text{many } H_{i^\star,j}) \\
&= n\cdot\left(1-(\kappa_{\mathrm{rs}})^{p-1}\right)\cdot\left(\frac{1-2\delta_{\mathrm{rs}}}{3+2\delta_{\mathrm{rs}}}\right) + \text{Number of vertices of } \widetilde{G} \text{ not touching } M_{i^\star}^{\mathrm{rs}} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Eq (6.1), Eq (6.2))} \\
&= n\cdot\left(1-(\kappa_{\mathrm{rs}})^{p-1}\right)\cdot\left(\frac{1-2\delta_{\mathrm{rs}}}{3+2\delta_{\mathrm{rs}}}\right) + 2(b_p-r_p)\cdot n_{p-1} \\
&= n\cdot\left(1-(\kappa_{\mathrm{rs}})^{p-1}\right)\cdot\left(\frac{1-2\delta_{\mathrm{rs}}}{3+2\delta_{\mathrm{rs}}}\right) + \left(\frac{2+4\delta_{\mathrm{rs}}}{3+2\delta_{\mathrm{rs}}}\right)\cdot n \qquad \text{(again by Eq (6.1), Eq (6.2))} \\
&= n\cdot\left(1-(\kappa_{\mathrm{rs}})^{p-1}\right)\cdot\kappa_{\mathrm{rs}} + (1-\kappa_{\mathrm{rs}})\cdot n \\
&= n\cdot\left(1-(\kappa_{\mathrm{rs}})^{p}\right),
\end{aligned}$$

which completes the proof. ∎

Property 6.1.8. *For positive $\varepsilon \le (\kappa_{\mathrm{rs}})^p$, any $(1-\varepsilon)$-approximate matching of $G \sim \mathcal{G}_p(n)$ contains at least an $(1-(\varepsilon/(\kappa_{\mathrm{rs}})^p))$ fraction of the edges of the ultimate matching $M^\star$.*

Proof. Let M be a $(1-\varepsilon)$-approximate maximum matching of G. By Property (6.1.7), G has a perfect matching of size n which implies that M has at least $(1-\varepsilon)\cdot n$ edges.

The size of the largest matching that does not use any edges of $M^\star$ is at most the vertex

cover of $\overline{G}$ from Property (6.1.7) by Konig's theorem. Therefore, the size of such a matching is at most,

$$n \cdot (1 - (\kappa_{\text{rs}})^p)\,.$$

Hence, M must contain edges of $M^\star$. In particular, we have that M contains at least,

$$\begin{aligned} &\geq (1-\varepsilon) \cdot n - (1 - (\kappa_{\text{rs}})^p) \cdot n \\ &= ((\kappa_{\text{rs}})^p - \varepsilon) \cdot n \end{aligned}$$

many edges of the ultimate matching $M^\star$. Finally, by Property (6.1.5), the number of edges in $M^\star$ is $(\kappa_{\text{rs}})^p \cdot n$. The matching M contains at least $(1 - \varepsilon \cdot (1/(\kappa_{\text{rs}})^p))$ fraction of the edges of the matching $M^\star$. ∎

These properties tell us that any protocol for β-approximate maximum matching with $\beta = (1-\varepsilon)$ with its input distributed according to $\mathcal{G}_p(n)$ necessarily learns at least $(1 - \varepsilon/(\kappa_{\text{rs}})^p)$ edges of $M^\star$. Now, suppose the goal was to use the output of the protocol and determine the edges of a uniform random gadget from $\mathcal{M}^\star$. Learning any one of the two edges of the gadget would reveal the other edge also (be it a parallel gadget or a cross gadget). Thus, any $(1-\varepsilon)$-approximation reveals at least $(1 - \varepsilon/(\kappa_{\text{rs}})^p)/2$ ultimate gadgets. We formalize this intuition by using search sequences and predicates to point to ultimate gadgets. Large matchings will be used to solve these predicates.

Let $J := (j_p, j_{p-1}, \ldots, j_0)$ where $j_i \in [r_i]$ for $0 \leq i \leq p$ be a **search sequence** that identifies a **ultimate gadget** $M^\star_J := M^\star_{j_p, j_{p-1}, \ldots, j_0} \in \mathcal{M}^\star$ of an input graph $G \sim \mathcal{G}_p(n)$. We define the **search predicate** as

$$\mathbb{P}_p(G, J) := \begin{cases} 0 & \text{if the special gadget } M^\star_J \text{ is a parallel gadget and} \\ 1 & \text{if it is a cross gadget.} \end{cases}$$

For any $(p+1)$-player protocol π on input G sampled from $\mathcal{G}_p(n)$, we say that π **solves the search predicate** $\mathbb{P}_p$ for the input graph G if, given only the transcript of messages communicated in π and a uniform random search sequence $J = (j_p, \ldots, j_0)$ (hidden from the protocol and with no further access to G), $\mathbb{P}_p(G, J)$ can be deterministically determined. It is important to note that J is *not* part of the input to the protocol and is probabilistically independent from the graph G.

Lemma 6.1.9. *For every $0 \leq \tau \leq 1$ and constant $0 < \varepsilon \leq (\kappa_{\text{rs}})^p$, any p-round $(p+1)$-player protocol π for $(1-\varepsilon)$-approximate maximum matching on input $G \sim \mathcal{G}_p(n)$ that succeeds with probability at least τ and with communication cost s can be altered and then used to solve the search predicate $\mathbb{P}_p$ for a uniform random search sequence J with probability at least*

$$(\tau/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$$

with communication cost $s + O(n \log n)$.

Proof. Let M be the $(1-\varepsilon)$-approximate maximum matching of G that is output by the protocol π and can thus be determined from its transcript. We use the output to solve $\mathbb{P}_p$ as

follows: If M contains at least one edge of the ultimate gadget $M^\star_J \in \mathcal{M}^\star$, we can correctly determine $\mathbb{P}_p(G, J)$. Otherwise, we output $\mathbb{P}_p(G, J) = 0$, which is essentially a random guess for $G \sim \mathcal{G}_p$.

When the protocol outputs a matching, we need to explicitly identify what edges from the ultimate gadgets and the ultimate induced matching have been sent in the transcript. We do this by recording the edges of M^{aux} explicitly at every level: in the first round, player Q_{p+1} additionally communicates the edges M^{aux} goven to them (M^{aux} from the definition of $\mathcal{G}_p(n)$) that identifies the r_p many special subgraphs sampled from $\mathcal{G}_{p-1}(n_{p-1})$. In the next round, Q_p outputs the M^{aux} matchings from *all* r_p-many special subgraphs sampled from $\mathcal{G}_{p-2}(n_{p-2})$. Continuing this way, we know that the edges in M which have not been revealed as a part of M^{aux} at some level, are from the ultimate induced matching.

To output these auxiliary edges explicity, one edge takes $O(\log n)$ communication and there are totally only n edges, as they form a matching. Hence, this adds at most $O(n \log n)$ factor to the communication.

Let us analyze the probability of success of this strategy. We have three cases:

(*i*) When π succeeds, which happens with probability τ, and the transcript identifies the gadget that J points to, the protocol can solve the predicate exactly. By Property (6.1.8), we know that the transcript contains at least $(1 - \varepsilon/(\kappa_{\text{rs}})^p)$-fraction of the edges of $M^\star$. The probability that it contains at least one edge from a randomly chosen J is also at least $(1 - \varepsilon/(\kappa_{\text{rs}})^p)$.

(*ii*) When π succeeds, but it does not identify the gadget that J points to, which happens with probability at most $\tau \cdot (\varepsilon/(\kappa_{\text{rs}})^p)$, we output a random guess. This succeeds with probability $(1/2)$.

(*iii*) When the algorithm fails, which happens with probability $(1 - \tau)$, we may as well assume our answer is wrong since the transcript can contain edges that are not present in the graph.

Hence, the total probability of success is at least:

$$\tau \cdot (1 - \varepsilon/(\kappa_{\text{rs}})^p) + \tau(\varepsilon/(\kappa_{\text{rs}})^p) \cdot (1/2) = (\tau/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p). \qquad \blacksquare$$

With that, to prove Theorem 11, it is enough to only consider protocols that solve the search predicate and, thus, we prove the following:

Lemma 6.1.10. *For every constant integer $p \geq 1$ any p-round protocol π that, given input $G \sim \mathcal{G}_p(n_p)$ and a uniform random search sequence, solves predicate $\mathbb{P}_p$ with probability at least*

$$\tau \cdot \rho \geq \frac{1}{2} + \frac{p}{(12)(p+1)}$$

has communication cost

$$CC(\pi) \geq \left(\frac{1}{(144 c_{\text{comp}}(p+1)^2)}\right)^p n_p^{1+1/(2\log\log n_p)^{3p}}$$

where c_{comp} is the constant from Eq (5.2) and c_{rs} is the constant from Proposition 4.2.2.

The proof of this lemma is by a direct application of Theorem 10.

Claim 6.1.11. *The problem of solving a predicate $\mathbb{P}_p$ on input graph $G \sim \mathcal{G}_p(n)$ obeys the properties (i) to (v) required to apply Theorem 10 with $\rho = (1/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$.*

Proof. Property (i) holds by the communication model we consider. This model is described in Section 5.3. Property (ii) is proven in Property (6.1.3). Property (iii) is given by the description of $\mathcal{G}_p(n)$, specifically line (2) which describes how the smaller instances are sampled.

Property (iv) is given by line (5) in the description of $\mathcal{G}_p(n)$, as the input of Q_{p+1} is fixed by index $i^\star$. Lasty, (v) is given by Lemma 6.1.9. This lemma proves that for a uniformly valid search sequence $J = (j_p, j_{r-1}, \ldots, j_0)$, the transcript determines $\mathbb{P}_p(G, J)$. For any subgraph $H_{i^\star,j}$ for $j \in [p_r]$, for any valid search sequence $J' = (j'_{p-1}, \ldots, j'_1)$ for special subgraph $H_{i^\star,j}$, we have:

$$\mathbb{P}_{p-1}(H_{i^\star,j}, (j'_{p-1}, \ldots, j'_1)) = \mathbb{P}_p(G, (j, j'_{p-1}, \ldots, j'_1)),$$

as the ultimate gadget that $(j, j_{p-1}, \ldots, j_0)$ points to is present in special subgraph $H_{i^\star,j}$ and the sequence $(j_{p-1}, \ldots, j_0)$ reveals this gadget in $H_{i^\star,j}$.

Therefore, for a uniformly random j, and a sequence $(j'_{p-1}, \ldots, j'_1)$, the transcript can be used to solve $\mathbb{P}_p$ on this sequence with probability at least $(\tau/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$, with $\rho = (1/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$. ∎

A simple corollary of Theorem 10 with $\rho = (1/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$ and Claim 6.1.11 gives the following round elimination lemma for $\mathbb{P}_p(G, J)$ for graphs $G \sim \mathcal{G}_p(n)$.

Corollary 6.1.12. *For every $p \geq 1$, suppose there exists a deterministic protocol π that solves $\mathbb{P}_p(G, J)$ for $G \sim \mathcal{G}_p(n)$ and uniformly random J with:*

- *(i) Probability of success of π in outputting a matching: τ with $0 < \tau < 1$,*
- *(ii) Constant δ_{rs} from Proposition 4.2.2,*
- *(iii) Constant ε such that π outputs a $(1 - \varepsilon)$-approximate maximum matching when it succeeds, with $\varepsilon \leq ((\kappa_{\text{rs}})^p/4)$,*
- *(iv) Parameter ρ, which is the probability with which transcript of π can be used to solve a smaller instance from $\mathcal{G}_{p-1}(n_{p-1})$, when it is successful with $\rho = (1/2) \cdot (2 - \varepsilon/(\kappa_{\text{rs}})^p)$*
- *(v) Parameter η associated with Theorem 10, $0 < \eta < 1$,*
- *(vi) Communication cost of π is at most s, for some $s \in \mathbb{R}$.*

Then, for constant c_{comp}, there exists a deterministic $(p-1)$-round p-player protocol π_{p-1} that, given input $G_{p-1} \sim \mathcal{G}_{p-1}$ and a uniform random search sequence J', solves $\mathbb{P}_{p-1}(G_{p-1}, J')$ with probability at least

$$\tau \cdot \rho - \eta - \sqrt{\frac{s}{2 \cdot r_p \cdot t_p}}$$

and communication cost

$$CC(\pi_{p-1}) \leq \frac{c_{\text{comp}}}{\eta} \cdot \left(\frac{s}{r_p} + p^2\right)$$

where $c_{\text{comp}} \geq 1$ *is an absolute constant defined in Eq (5.2) (derived from Proposition 5.3.5).*

We can use Corollary 6.1.12 to prove the lower bound in Lemma 6.1.10.

Proof of Lemma 6.1.10. We prove the lower bound for deterministic protocols, and use they easy direction of Yao's minimax principle to extend the bound to all randomized protocols.

Before proving the result by induction, we give the following useful bounds for every constant $p \geq 1$: Using the parameters in Eq (6.1), Eq (6.2) and since $b_p = n_{p-1}^{(\log\log n_{p-1})^3}$, we have that

$$\begin{aligned} n_p &:= \left(\frac{3}{2} + \delta_{\text{rs}}\right) \cdot b_p \cdot n_{p-1} \\ &= \left(\frac{3}{2} + \delta_{\text{rs}}\right) \cdot (n_{p-1}) \uparrow (\log\log n_{p-1})^3 \end{aligned}, \qquad \begin{aligned} r_p &:= \left(\frac{1}{2} - \delta_{\text{rs}}\right) \cdot b_p \\ &= \left(\frac{1}{2} - \delta_{\text{rs}}\right) \cdot (n_{p-1}) \uparrow (\log\log n_{p-1})^3 \end{aligned}$$

$$\begin{aligned} t_p &= b_p^{c_{\text{rs}}/\log\log b_p} && (6.4) \\ &\geq \left(n_{p-1}^{(\log\log n_{p-1})^3}\right)^{1/(2\cdot\log\log n_{p-1})} && (\text{since } \log\log b_p \leq 2\cdot\log\log n_{p-1}) \\ &\geq n_{p-1}^4. && (\text{for large enough } n_{p-1}) \end{aligned}$$

Let $s_p := \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_p^{1+(1/(2\log\log n_p))^{3p}}$ (as in the statement). It then follows that

$$\begin{aligned} \frac{s_p}{r_p} + p^2 &= \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_p^{1/(2\log\log n_p)^{3p}} \cdot \frac{n_p}{r_p} + p^2 && (6.5) \\ &= \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_p^{1/(2\log\log n_p)^{3p}} \cdot \left(\frac{3/2+\delta_{\text{rs}}}{1/2-\delta_{\text{rs}}}\right) \cdot n_{p-1} + p^2 \\ & && (\text{from the parameters above}) \\ &\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_p^{1/(2\log\log n_p)^{3p}} \cdot (3+8\delta_{\text{rs}}) \cdot n_{p-1} + p^2 && (\text{for } \delta_{\text{rs}} \leq 1/4) \\ &\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_p^{1/(2\log\log n_{p-1})^{3p}} \cdot (3+8\delta_{\text{rs}}) \cdot n_{p-1} + p^2 \\ & && (\text{since } \log\log n_p \geq \log\log n_{p-1}) \\ &\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot n_{p-1}^{1/(2\log\log n_{p-1})^{3(p-1)}} \cdot (3+8\delta_{\text{rs}}) \cdot n_{p-1} + p^2 \\ & && (\text{since } \log n_p \leq 2\log(n_{p-1}) \cdot (\log\log n_{p-1})) \\ &\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot 12 \cdot n_{p-1}^{1/(2\log\log n_{p-1})^{3(p-1)}} \cdot n_{p-1}. \\ & && (\text{since } p^2 = o(n_{p-1}) \text{ and } \delta_{\text{rs}} < 1) \end{aligned}$$

We also have that

$$\begin{aligned}
\frac{s_p}{r_p \cdot t_p} &\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p n_p^{(1/(2\log\log n_p))^{3p}} \cdot \frac{n_p}{r_p} \cdot \frac{1}{t_p} && (6.6)\\
&\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p n_p^{(1/(2\log\log n_p))^{3p}} \cdot \frac{n_p}{r_p \cdot n_{p-1}} \cdot \frac{1}{n_{p-1}^3} \\
&&& (\text{as } t_p \geq n_{p-1}^4 \text{ from Eq (6.4)})\\
&= \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p n_p^{(1/(2\log\log n_p))^{3p}} \cdot \left(\frac{3+2\delta_{\text{rs}}}{1-2\delta_{\text{rs}}}\right) \cdot \frac{1}{(n_{p-1})^3} \\
&&& (\text{from Eq (6.1) and Eq (6.2)})\\
&\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot (11) \cdot (n_{p-1}) \cdot \frac{1}{(n_{p-1})^3} \\
&&& (\text{as } \delta_{\text{rs}} < 1 \text{ and } \log n_p < \log(n_{p-1}) \cdot (2\log\log n_{p-1}))\\
&\leq \left(\frac{1}{(144c_{\text{comp}}(p+1)^2)}\right)^p \cdot (11) \cdot (n_{p-1})^{-2} && (6.7)\\
&< \left(\frac{1}{13p(p+1)}\right)^2. && (\text{for large enough } n_{p-1})
\end{aligned}$$

With these bounds, we are now ready to proceed with the inductive argument.

Base Case (1-round). Towards a contradiction, let π_1 be a 1-round deterministic protocol that, given input $G \sim \mathcal{G}_1(n_1)$ and a uniform random search sequence, solves $\mathbb{P}_1$ with probability at least 2/3, as in the statement, and communication cost $\text{CC}(\pi_1) < s_1$, which is strictly less than the statement. We use Corollary 6.1.12 with $\eta = 1/(12)$ to get a protocol with probability of success at least:

$$\frac{2}{3} - \frac{1}{12} - \frac{1}{14} > \frac{1}{2},$$

by our bound on $(s_1/(t_1 r_1))$ above. This protocol is deterministic, and uses 0-rounds to solve predicate $\mathbb{P}_0$ on input graph $G_0 \sim \mathcal{G}_0$ and a uniform random search sequence. Regardless of the communication cost, this is a contradiction for 0-round protocols as they cannot do better than random guessing. Therefore, π_1 must have communication cost $\text{CC}(\pi_1) \geq s_1$.

Inductive Step (p-round). Now, suppose that the statement is true for $(p-1)$-round deterministic protocols. Let π_p be a p-round deterministic protocol that, given input $G \sim \mathcal{G}_p$ and a uniform random search sequence, solves $\mathbb{P}_p$ with probability at least $\frac{1}{2} + \frac{p}{3(p+1)}$, as in the statement, and communication cost $\text{CC}(\pi_p) < s_p$. By Corollary 6.1.12 with $\eta = 1/(6p(p+1))$ and the above bounds, there exists a $(p-1)$-round deterministic protocol that, given input $G_p \sim \mathcal{G}_{p-1}(n_{p-1})$ and a uniform random search sequence, solves $\mathbb{P}_{p-1}$ with probability at least

$$\frac{1}{2} + \frac{p}{12(p+1)} - \frac{1}{12p(p+1)} - \sqrt{s_p/(2 \cdot r_p \cdot t_p)}$$

$$\geq \frac{1}{2} + \frac{p}{12(p+1)} - \frac{p}{12(p+1)} - \frac{1}{13p(p+1)} \qquad \text{(from Eq (6.6))}$$
$$\geq \frac{1}{2} + \frac{(p-1)}{12p},$$

and communication cost

$$\begin{aligned}
\mathrm{CC}(\pi_{p-1}) &< \frac{c_{\mathrm{comp}}}{\eta} \cdot (\frac{s_p}{r_p} + p^2) \\
&\leq c_{\mathrm{comp}} \cdot (12p(p+1)) \cdot \left(\frac{1}{(144c_{\mathrm{comp}}(p+1)^2)}\right)^p \cdot 12 \cdot n_{p-1}^{1+1/(2\log\log n_{p-1})^{3(p-1)}} \\
&\qquad \text{(from Eq (6.5))} \\
&\leq \left(\frac{1}{(144c_{\mathrm{comp}} \cdot p^2)}\right)^{p-1} \cdot n_{p-1}^{1+1/(2\log\log n_{p-1})^{3(p-1)}} = s_{p-1}.
\end{aligned}$$

This contradicts the induction hypothesis and, hence, π_p must have communication cost $\mathrm{CC}(\pi_p) \geq s_p$. ∎

We now explicitly complete the proof of Theorem 11.

Proof of Theorem 11. Let π be any p-round $(p+1)$-protocol for $(1-\varepsilon)$-approximate maximum matching with

$$\varepsilon < (1/4) \cdot (\kappa_{\mathrm{rs}})^p, \qquad \kappa_{\mathrm{rs}} = \frac{1-2\delta_{\mathrm{rs}}}{3+2\delta_{\mathrm{rs}}},$$

probability of success $\tau = (2/3)$ and communication cost s (as in the statement).

By Lemma 6.1.9, protocol π on input $G \sim \mathcal{G}_p(n)$ can be made into a protocol π' with communication cost $s+O(n\log n)$ that, given input G and a uniform random search sequence, solves the search predicate $\mathbb{P}_p$ with probability at least

$$\frac{\tau}{2} \cdot (2 - \varepsilon/(\kappa_{\mathrm{rs}})^p) > \frac{2}{3} \cdot \frac{1}{2} \cdot (2 - \frac{1}{4}) = \frac{7}{12} > \frac{1}{2} + \frac{p}{12(p+1)},$$

for $\tau = 2/3$ and $\varepsilon < (1/4) \cdot (\kappa_{\mathrm{rs}})^p$. Therefore, by Lemma 6.1.10, protocol π' has communication cost

$$\mathrm{CC}(\pi') = s + O(n\log n) \geq \left(\frac{1}{(144c_{\mathrm{comp}}(p+1)^2)}\right)^p \cdot n^{1+1/(2\log\log n)^{3p}} = n^{1+\Omega(1/\log\log n)^{3p}},$$

for $p = o(\log\log n/\log\log\log n)$, as c_{comp} is a constant. Finally, by rearranging, we have that $\mathrm{CC}(\pi) = s = n^{1+\Omega\left(1/(\log\log n)^{3p}\right)}$. ∎

6.2 Dynamic Streams

We move to the proof of Theorem 4. Again, we skip a detailed overview as the key points were mentioned in Chapter 3 already.

6.2.1 Preliminaries

The technical preliminaries specific to Theorem 4 is given in this section.

Notation. We use S_n to denote the set of all permutations over $[n]$ for any $n \geq 1$. For any permutation $\sigma \in S_n$, we use $\sigma(i)$ to denote the element in $[n]$ that σ maps i to for each $i \in [n]$.

For any matrix Z of dimensions $m \times n$ for integers $m, n \geq 1$, we use $Z[i, j]$ to denote the value at row i and column j for $i \in [m], j \in [n]$.

For a graph $G = (V, E)$ and any vertex $v \in V$, we use $N(v)$ to denote the neighbors of v and $\deg v := |N(v)|$ to denote its degree. For any $U \subseteq V$, we use $G[U]$ to denote the subgraph of G induced on vertices in U. We further use $\mu(G)$ to denote the maximum matching size in G.

6.2.1.1 Two-Party Communication Complexity

We work in the standard two party communication model; we provide some basic definitions here and refer the interested reader to the excellent textbooks [KN97, RY20] for more details.

There are two players Alice and Bob who receive input from $\mathcal{X}$ and $\mathcal{Y}$ respectively. The players jointly compute some function f with domain $\mathcal{X} \times \mathcal{Y}$. The players follows some protocol π to compute f. They have access to a shared tape of randomness, referred to as *public randomness*, in addition to their own *private randomness*.

Alice first sends a message to Bob, followed by a message from Bob to Alice and so on. The last player who receives a message has to output the answer. The total number of rounds is the total number of messages passed between Alice and Bob. Moreover, the message sent by any player only depends on their private input, the communicated messages so far, the public randomness, and the private randomness.

Definition 6.2.1. *For any protocol π, the **communication cost** of π, denoted by $CC(\pi)$, is defined as the worst-case (maximum) total length of messages, measured in bits, communicated by the players on any input. We assume that all **transcripts**, i.e., the set of all messages sent by any player, in π have the same worst-case length (by padding).*

Using [AMS96], again we can relate the communication cost to the space of streaming algorithms, similar to Proposition 5.3.2 for the two-party model also.

6.2.1.2 Information Cost and Message Compression

We also work with the notion of *information cost* of protocols that originated in [CSWY01] and has since found numerous applications (see, e.g., [Wei15] for an excellent survey of this topic). There are various definitions of information cost that have been considered depending on the application. The following definition due to [BBCR10] is best suited for our purpose.

Definition 6.2.2. *For any two-party protocol π whose inputs are distributed according to some distribution μ, the **(internal) information cost** is defined as:*

$$IC(\pi, \mu) := \mathbb{I}(\mathsf{X}\,; \Pi \mid \mathsf{R}, \mathsf{Y}) + \mathbb{I}(\mathsf{Y}\,; \Pi \mid \mathsf{R}, \mathsf{X}),$$

where X, Y *denote the random variable for the inputs* X, Y *sampled from* μ, Π *denotes the random variable corresponding to the communicated messages, and* R *is the public randomness.*

Since a single bit of communication can only carry one bit of information, we can upper bound information cost by the communication cost.

Proposition 6.2.3 (cf. [BBCR10])**.** *For any protocol* π *on any distribution* μ, $IC(\pi, \mu) \leq CC(\pi)$.

We also use *message compression* to reduce communication cost of limited-round protocols close to their information cost. The following is a simplified version of [JPY16, Theorem 3.4].

Proposition 6.2.4 ([JPY16, Theorem 3.4])**.** *For* $r \geq 1$, *any* $\varepsilon \in (0, 1)$, *and input distribution* μ, *any* r*-round protocol* π *can be simulated with error at most* ε *in* r*-rounds by a protocol* π' *with communication at most*

$$c_{JPY} \cdot \left(r/\varepsilon \cdot IC(\pi, \mu) + r^2/\varepsilon\right)$$

for some absolute constant $c_{JPY} \geq 1$. *The simulation by* π' *uses a source of public randomness that is independent from any randomness used in* π.

6.2.2 A Hard Distribution

We begin the proof of Theorem 4 in this section with the communication problem and the corresponding hard distribution. We restate the result here for convenience.

Theorem 4 (restated). *Any semi-streaming algorithm that given any* n*-vertex graph in a dynamic streams, outputs an* $O(1)$*-approximation to the maximum matching problem with any constant probability of success requires* $\Omega(\log \log n)$ *passes.*

There are two parts to the proof.

We prove Theorem 4 using the connection between streaming and communication models (Proposition 5.3.2). Our proof consists of two separate parts proven in the subsequent sections.

Part 1 (Information Theory). In Section 6.2.2, we define a new two-party communication problem for r-round protocols called Augmented Hidden Matrices (AHM_r, see Definition 6.2.7) and prove the following lower bound on its communication complexity in Section 6.2.2.1.

Lemma 6.2.5. *For any sufficiently large* $n_r \in \mathbb{N}$, *positive integer* $r = O(\log \log(n_r))$, *positive number* $\alpha \leq 1 - n_r^{-1/2r}$ *and constant* $c_{ADV} \geq 1$, *any* r*-round protocol that successfully solves* $\mathsf{AHM}_r(n_r, \alpha)$ *with probability of success at least*

$$\frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{ADV} \cdot (r+1)}\right)$$

must have a communication cost of at least

$$s_r = (n_r) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1}\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{JPY} \cdot c_{ADV}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}.$$

Part 2 (Combinatorics). In Section 6.2.3, we construct graph instances corresponding to the $\mathsf{AHM}_r(n_r, \alpha)$ problem. We prove the following connection between any p-pass dynamic streaming algorithm for approximate maximum matching and AHM_r where $r = 2p - 1$ in Section 6.2.3.3.

Lemma 6.2.6. *For any sufficiently large $n_r \in \mathbb{N}$, integers $p, s \geq 1$, and number $\beta \geq 1$, let $\mathcal{A}$ by any p-pass s-space dynamic streaming algorithm that computes a β-approximate maximum matching on a $(4n_r)$-vertex bipartite graphs with probability of success at least $1 - 1/\mathrm{poly}(n)$. Then, for $r = 2p - 1$ and $\alpha = 1/(4 \cdot \beta \cdot r)$, there exists an r-round protocol π for $\mathsf{AHM}_r(n_r, \alpha)$ with probability of success*

$$\mathrm{suc}(\pi) \geq \frac{1}{2} \cdot \left(1 + \frac{1}{3\beta}\right)$$

and communication cost

$$CC(\pi) \leq r \cdot s + O(r \cdot n_r \log(n_r)).$$

We now prove Theorem 4 using the above two lemmas.

Proof of Theorem 4. Without loss of generality, we can solely focus on proving the lower bound for semi-streaming algorithms that succeed with high probability instead of constant probability. This is due to the following standard reduction.

Simply run a constant probability of success streaming algorithm in parallel $O(\log n)$ times and consider all the outputs; then, spend one more pass to filter out all the ones that output an edge that does not belong to the graph (because they failed), and among the rest, output the largest matching. This way, we still obtain a semi-streaming algorithm with just one more pass (which is negligible in this context) and a high probability of success. As such, in the rest of the proof, we focus on high-probability-of-success algorithms.

For sufficiently large $n \in \mathbb{N}$, integers $p, s \geq 1$ and any constant $\beta \geq 1$, let $\mathcal{A}$ be any p-pass s-space dynamic streaming algorithm that computes a β-approximate maximum matching on any $(4n)$-vertex graph with probability of success at least $1 - 1/\mathrm{poly}(n)$.

By Lemma 6.2.6 with $r = 2p - 1$ and $\alpha = 1/(4 \cdot \beta \cdot r)$, there exists an r-round protocol π for $\mathsf{AHM}_r(n, \alpha)$ where

$$\mathrm{suc}(\pi) \geq \frac{1}{2} \cdot \left(1 + \frac{1}{3\beta}\right) \qquad \text{and} \qquad \mathrm{CC}(\pi) \leq r \cdot s + O(r^2 \cdot n \log n).$$

By Lemma 6.2.5 with $n_r = n$ and $c_{\text{ADV}} = \beta$, we have that $\text{CC}(\pi) \geq s_r$ since

$$\text{suc}(\pi) \geq \frac{1}{2} \cdot \left(1 + \frac{1}{3\beta}\right) \geq \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot \beta \cdot (r+1)}\right).$$

Then, by combining the bounds on $\text{CC}(\pi)$ and re-arranging the inequality, we obtain the following lower bound on the space of $\mathcal{A}$ using the fact that c_{JPY} and β are constants (and thus $\alpha = \Theta(1/r)$):

$$\begin{aligned} s &\geq \frac{s_r}{r} - O(n \log n) \\ &= (n) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1}\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\text{JPY}} \cdot \beta^2 \cdot 80^2)^r \cdot ((r+1)!)^3} \cdot \frac{1}{r} - O(n \log n) \\ &\geq (n) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot r^{-\Theta(r)} - O(n \log n). \end{aligned} \tag{6.8}$$

Overall, by considering $\mathcal{A}$ with $p = o(\log \log n)$ and thus $r = 2p - 1 = o(\log \log n)$, we have that the space used by the algorithm is $s \gg n \cdot \text{polylog}(n)$ using Eq (6.8). Therefore, any dynamic streaming algorithm that uses $n \cdot \text{polylog}(n)$ space, namely, a semi-streaming algorithm, to compute an $O(1)$-approximate maximum matching must use $\Omega(\log \log n)$ passes. ∎

Augmented Hidden Matrices

For any $r \geq 1$, the $\mathsf{AHM}_r(n_r, \alpha)$ problem is defined recursively using b_r^2 many instances of the $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ problem where $n_r, b_r, n_{r-1} \in \mathbb{N}$ and $\alpha \in (0, 1)$. The output of any instance is a single bit, pointed to by a *search sequence* (to be defined soon).

Definition 6.2.7 (Augmented Hidden Matrices)**.** *The Augmented Hidden Matrices problem, denoted by $\mathsf{AHM}_r(n_r, \alpha)$, is defined as follows (see Figure 6.4 for an illustration):*

- ***For* $\mathbf{r = 0}$.** *Alice does not receive any input, i.e., $A^{(0)} = \emptyset$, and Bob receives a single bit $B^{(0)} \in \{0, 1\}$. There is no communication and Alice has to output $B^{(0)}$.*
- ***For* $\mathbf{r \geq 1}$.** *Start with b_r^2 instances of the players' inputs in $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$, denoted by $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ for $i, j \in [b_r]$. Alice receives $A^{(r)} = X$ where X is a $b_r \times b_r$ matrix such that $X[i, j] = B_{i,j}^{(r-1)}$ for $i, j \in [b_r]$. Bob receives the input $B^{(r)} = (\sigma_{\text{R}}, \sigma_{\text{C}}, Y)$ where $\sigma_{\text{R}}, \sigma_{\text{C}} \in S_{b_r}$ are permutations of $[b_r]$ and Y is a $b_r \times b_r$ matrix such that, for $i, j \in [b_r]$,*

$$Y[i, j] = \begin{cases} \left(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)}\right) & \text{if } b_r \cdot \alpha < \sigma_{\text{R}}(i) \neq \sigma_{\text{C}}(j) \leq b_r, \\ A_{i,j}^{(r-1)} & \text{if } b_r \cdot \alpha < \sigma_{\text{R}}(i) = \sigma_{\text{C}}(j) \leq b_r, \\ \emptyset & \text{otherwise.} \end{cases}$$

Alice starts the communication. After all the messages are sent, the player receiving the last message is given a ***search sequence****, which is a tuple of r integers, $(k_r^\star, k_{r-1}^\star, \ldots, k_1^\star)$ with $k_i^\star \in [b_i \cdot (1 - \alpha)]$. This player must output the solution to*

AHM_{r-1} *instance* $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$ *where* $\sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) = b_r \cdot \alpha + k^\star_r$ *using the search sequence* $(k^\star_{r-1}, k^\star_{r-2} \dots k^\star_1)$.

We expound more on the defintion of AHM_r problem, and give some useful terminology next.

For $r = 0$ case, we set $n_0 = 1$, and $\mathsf{AHM}_0(1, \alpha)$ is also denoted by $\mathsf{AHM}_0(1)$. This is a base case that is trivially hard for 0-round protocols.

For each of the b_r^2 many AHM_{r-1} instances $(A^{(r-1)}, B^{(r-1)})$ used in an instance of AHM_r, the roles of Alice and Bob are swapped, i.e., Alice holds $B^{(r-1)}$ and Bob holds $A^{(r-1)}$. Out of these instances, we call the set of the $\boldsymbol{k_r} := b_r \cdot (1-\alpha)$ many AHM_{r-1} instances $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$ where $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) \leq b_r$ as the **special sub-instances** in an instance of AHM_r. Since each special sub-instance identifies another set of k_{r-1} many AHM_{r-2} special (sub-)sub-instances, recursively applying this for all special sub-instances of AHM_r identifies a set of $k_r \cdot k_{r-1} \cdot \ldots \cdot k_1$ many AHM_0 **special base instances**, where each one corresponds to a single bit $B^{(0)}$.

It is useful to point to the number of special base instances explicitly for later.

Observation 6.2.8. *In any instance of* $\mathsf{AHM}_r(n_r, \alpha)$*, there are* $k_r \cdot k_{r-1} \cdot \ldots \cdot k_1$ *many special base instances where* $k_i = b_i \cdot (1 - \alpha)$ *for each* $i \in [r]$*.*

Furthermore, we call the set of $k_r^2 - k_r$ many AHM_{r-1} instances $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$ where $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r$ as the **off-diagonal sub-instances** in an instance of AHM_r. We define some more notation to point to the sub-instances.

Notation. In any instance of AHM_r, denoted by $(A^{(r)}, B^{(r)})$, we use the following notation. We denote each of the k_r many special sub-instances as $(A^{\text{SPEC}}_k, B^{\text{SPEC}}_k) = (A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$ where $\sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) = b_r \cdot \alpha + k$ and $i, j \in [b_r]$ for $k \in [k_r]$. We collectively denote these special sub-instances as $(A^{\text{SPEC}}, B^{\text{SPEC}})$ where

$$A^{\text{SPEC}} = (A^{\text{SPEC}}_1, A^{\text{SPEC}}_2, \ldots, A^{\text{SPEC}}_{k_r}) \quad \text{and} \quad B^{\text{SPEC}} = (B^{\text{SPEC}}_1, B^{\text{SPEC}}_2, \ldots, B^{\text{SPEC}}_{k_r}).$$

We denote the collection of the $k_r^2 - k_r$ many off-diagonal sub-instances as $(A^{\text{OFF}}, B^{\text{OFF}})$ where

$$A^{\text{OFF}} = (A^{(r-1)}_{i,j} : b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r) \quad \text{and} \quad B^{\text{OFF}} = (B^{(r-1)}_{i,j} : b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r).$$

We let B^{REST} be the remaining $b_r^2 - k_r^2$ sub-instances $B^{(r-1)}_{i,j}$ where $\sigma_{\mathrm{R}}(i) \leq b_r \cdot \alpha$ or $\sigma_{\mathrm{C}}(j) \leq b_r \cdot \alpha$.

Search Sequences. After all the messages are sent, the search sequence given to the player who receives the last message is always **uniformly random** and **independent of the players' inputs**. That is, a search sequence $(k^\star_r, k^\star_{r-1}, \ldots, k^\star_1)$ is chosen where for $i \in [r]$, $k^\star_i$ is chosen uniformly at random and independently from $[b_i \cdot (1 - \alpha)]$. The solution to any instance of AHM_r given $(k^\star_r, k^\star_{r-1}, \ldots, k^\star_1)$ is the same as the solution to the uniformly chosen special sub-instance $(A^{\text{SPEC}}_{k^\star_r}, B^{\text{SPEC}}_{k^\star_r})$ on the search sequence $(k^\star_{r-1}, \ldots, k^\star_1)$. Continuing this until the last index $k^\star_1$ in the search sequence ultimately identifies a uniform random special base instance, i.e., an instance of AHM_0, whose solution is the solution to the AHM_r instance.

Our definition of the $\mathsf{AHM}_r(n_r, \alpha)$ communication problem is non-standard as the uniformly random search sequence, which is a part of the input, is given to the final player *at the very end of a protocol*. In particular, the search sequence is crucial to defining the output of the problem, but does not form part of the input held by the players throughout the protocol.

Although the search sequence defines the communication problem in a non-standard way, the easy direction of Yao's minmax theorem is still applicable. Thus, we prove Lemma 6.2.5 by considering deterministic protocols for $\mathsf{AHM}_r(n_r, \alpha)$ where the players' inputs $(A^{(r)}, B^{(r)})$ are sampled from the hard distribution (which we define shortly).

Remark 6.2.9. *The definition of* $\mathsf{AHM}_r(n_r, \alpha)$ *(Definition 6.2.7) always includes the uniform random search sequence, which is independent of the players' inputs and how they are distributed. However, when referring to the* $\mathsf{AHM}_r(n_r, \alpha)$ *problem, we do not explicitly mention the given search sequence if this is clear from context.*

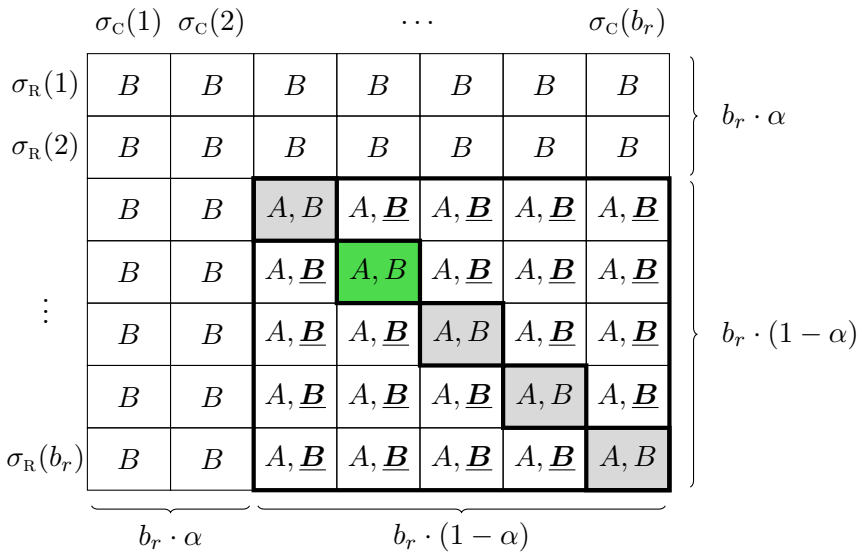


Figure 6.4: An illustration of $\mathsf{AHM}_r(n_r, \alpha)$ where $b_r = 7$, $k_r^\star = 2$, and $\alpha = 2/7$, i.e., $b_r \cdot \alpha = 2$. In this figure, the $b_r \times b_r$ matrix is presented after applying the respective permutations of rows and columns using σ_R and σ_C, which are held by Bob (alternatively, think of σ_R and σ_C as identity permutations here). In each position of the matrix, the corresponding input is either a complete instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ denoted by $(A, B) = (A^{(r-1)}, B^{(r-1)})$ or a partial instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ denoted by $B = B^{(r-1)}$ due to the asymmetry of the construction. All of the B's are held by Alice, all of the A's are held by Bob, and Bob further holds the B's corresponding to B^{OFF}, which are bolded and underlined. The shaded instances are the special sub-instances $(A^{\text{SPEC}}, B^{\text{SPEC}})$, i.e., the ones in the diagonal of the bolded box. The remaining B's belong to B^{REST}, i.e., the ones not in the bolded box. Given the search sequence $(k_r^\star, k_{r-1}^\star, \ldots, k_1^\star)$, the goal is to solve the the instance identified by $k_r^\star = 2$ (shaded in green, and denoted by $(A_2^{\text{SPEC}}, B_2^{\text{SPEC}})$), using the search sequence $(k_{r-1}^\star, \ldots, k_1^\star)$.

Next, we fix the parameters of the problem, and then define our hard distribution $\mathcal{D}_r(n_r, \alpha)$.

Parameter Choices. For all $r \geq 1$, it is sufficient for us to consider the AHM_r problem for parameters that satisfy the following conditions:

$$\begin{aligned} n_r &= n_{r-1} \cdot b_r \\ b_r &= (n_r) \uparrow \left(\frac{2^{r-1}}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^{r-1}}{2^r - 1} - 1\right) \\ n_{r-1} &= (n_r) \uparrow \left(\frac{2^{r-1} - 1}{2^r - 1}\right) / (1-\alpha) \uparrow \left(\frac{r \cdot 2^{r-1}}{2^r - 1} - 1\right). \end{aligned} \tag{6.9}$$

From now on, we only consider $\mathsf{AHM}_r(n_r, \alpha)$ with these parameter choices.

Hard Distribution $\mathcal{D}_r(n_r, \alpha)$ for $\mathsf{AHM}_r(n_r, \alpha)$:

- **For r = 0.** Sample $B^{(0)} \in \{0, 1\}$ uniformly and independently. We also use $\mathcal{D}_0(1)$ to denote this distribution.
- **For r $\geq$ 1.** Sample the b_r^2 instances $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$ independently for each $i, j \in [b_r]$ where b_r, n_{r-1} are as defined in Eq (6.9). Sample the permutations $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}} \in S_{b_r}$ uniformly and independently.

Recall that the distribution of the search sequence was already defined to be uniformly random, and independent of all other inputs.

We have the following obvious observation.

Observation 6.2.10. *Any 0-round protocol (wherein Alice outputs the answer) for solving $\mathsf{AHM}_0(1)$ succeeds with probability at most $1/2$ when the input is sampled from $\mathcal{D}_0(1)$.*

The proof of Lemma 6.2.5 relies on a round-elimination argument that shows that if there is a good deterministic protocol for $\mathsf{AHM}_r(n_r, \alpha)$ under the input distribution $\mathcal{D}_r(n_r, \alpha)$, then there must be a good deterministic protocol for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ under the input distribution $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$. We prove the following key lemma in Section 6.2.2.2.

Lemma 6.2.11. *For any $r \geq 1$, sufficiently large $n_r \in \mathbb{N}$, $\alpha, \delta \in (0, 1)$, and integer $s \geq 1$, suppose there exists a deterministic r-round protocol π_r for $\mathsf{AHM}_r(n_r, \alpha)$ with*

$$CC(\pi_r) \leq s$$

and probability of success at least δ when the input is sampled from $\mathcal{D}_r(n_r, \alpha)$. Then, for any $\varepsilon \in (0, 1)$, there exists a deterministic $(r-1)$-round protocol π_{r-1} for AHM_{r-1} with

$$CC(\pi_{r-1}) \leq c_{\text{JPY}} \cdot (r/\varepsilon) \cdot \left(\frac{s}{k_r} + r\right),$$

and probability of success

$$\mathrm{suc}(\pi_{r-1}) \geq \delta - \varepsilon - \sqrt{\frac{s}{2 \cdot b_r^2 \cdot \alpha^2}}$$

when the players' input is sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$.

With that, we have the main tools required to prove of our lower bound for $\mathsf{AHM}_r(n_r, \alpha)$.

6.2.2.1 The Lower Bound for Augmented Hidden Matrices: Proof of Lemma 6.2.5

Proof of Lemma 6.2.5. We prove this result by first assuming the existence of a deterministic r-round protocol π_r for AHM_r that is 'too good to be true' (w.r.t. the statement of the lemma) on inputs sampled from $\mathcal{D}_r(n_r, \alpha)$. Then, by iteratively applying the round-elimination argument in Lemma 6.2.11, we ultimately obtain a deterministic 0-round protocol π_0 for AHM_0 that is trivially 'too good to be true' on inputs sampled from $\mathcal{D}_0(1)$ (given Observation 6.2.10), a contradiction. Formally, we prove this by induction on r.

Base case for r = 1. Suppose that there exists a deterministic 1-round protocol π_1 for $\mathsf{AHM}_1(n_1, \alpha)$ with communication cost

$$\mathrm{CC}(\pi_1) < s_1 = \frac{n_1^2 \cdot \alpha^2}{c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2 \cdot 8}$$

and probability of success

$$\mathrm{suc}(\pi_1) \geq \frac{1}{2} \cdot \left(1 + \frac{1}{40 \cdot c_{\mathrm{ADV}}}\right)$$

when the input is sampled from $\mathcal{D}_1(n_1, \alpha)$. Then, using Lemma 6.2.11 with $\varepsilon = 1/(160 \cdot c_{\mathrm{ADV}})$, we obtain a deterministic 0-round protocol π_0 for $\mathsf{AHM}_0(1)$ which has no communication cost (since no messages are communicated) and succeeds with probability

$$\begin{aligned}
\mathrm{suc}(\pi_0) &\geq \frac{1}{2} \cdot \left(1 + \frac{1}{40 \cdot c_{\mathrm{ADV}}}\right) - \frac{1}{160 \cdot c_{\mathrm{ADV}}} - \sqrt{\frac{n_1^2 \cdot \alpha^2}{c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2 \cdot 16 \cdot b_1^2 \cdot \alpha^2}} \\
&\geq \frac{1}{2} \cdot \left(1 + \frac{1}{80 \cdot c_{\mathrm{ADV}}} - \sqrt{\frac{n_1^2}{c_{\mathrm{ADV}}^2 \cdot 160^2 \cdot b_1^2}}\right) && \text{(since } c_{\mathrm{JPY}} \geq 1\text{)} \\
&= \frac{1}{2} \cdot \left(1 + \frac{1}{160 \cdot c_{\mathrm{ADV}}}\right) && \text{(since } b_1 = n_1 \text{ by Eq (6.9))}
\end{aligned}$$

when the input is sampled from $\mathcal{D}_0(1)$. This, however, contradicts Observation 6.2.10 and thus proves the result for the base case when $r = 1$.

Inductive step for r ≥ 2. Having proven the base case, we now prove the result for any $r \geq 2$ given the result holds for $r - 1$, i.e., the inductive hypothesis. Suppose that there exists a deterministic protocol π_r for $\mathsf{AHM}_r(n_r, \alpha)$ with communication cost

$$\mathrm{CC}(\pi_r) < s_r = (n_r) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot (1 - \alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1}\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}$$

and probability of success

$$\mathrm{suc}(\pi_r) \geq \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\mathrm{ADV}} \cdot (r+1)}\right)$$

when the input is sampled from $\mathcal{D}_r(n_r, \alpha)$. Then, using Lemma 6.2.11 with $\varepsilon = 1/(40 \cdot c_{\mathrm{ADV}} \cdot (r+1)^2)$, we obtain a deterministic $(r-1)$-round protocol π_{r-1} for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ with cost

$$\mathrm{CC}(\pi_{r-1}) < c_{\mathrm{JPY}} \cdot (40 \cdot c_{\mathrm{ADV}} \cdot (r+1)^2 \cdot r) \cdot (s_r/k_r + r) \tag{6.10}$$

and probability of success

$$\mathrm{suc}(\pi_{r-1}) \geq \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\mathrm{ADV}} \cdot (r+1)}\right) - \frac{1}{40 \cdot c_{\mathrm{ADV}} \cdot (r+1)^2} - \sqrt{\frac{s_r}{2 \cdot b_r^2 \cdot \alpha^2}} \tag{6.11}$$

when the input is sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$.

We now argue that protocol π_{r-1} contradicts the result for $r-1$, namely, the inductive hypothesis. In particular, we show that $\mathrm{CC}(\pi_{r-1}) < s_{r-1}$ and

$$\mathrm{suc}(\pi_{r-1}) \geq \frac{1}{2} \cdot \left(1 + \frac{(r-1)}{20 \cdot c_{\mathrm{ADV}} \cdot r}\right).$$

The rest of the proof is a careful but rather tedious calculation of the values we obtain using the above reduction for the communication cost and probability of success of the protocol.

To bound the communication cost, we first reiterate the relation between n_r and n_{r-1} from Eq (6.9) as follows:

$$\begin{aligned}
(n_r) \uparrow \left(\frac{1}{2^r - 1}\right) &= (n_{r-1}) \uparrow \left(\frac{1}{2^{r-1} - 1}\right) \cdot (1-\alpha) \uparrow \left(\left(\frac{r \cdot 2^{r-1}}{2^r - 1} - 1\right) \cdot \frac{1}{2^{r-1} - 1}\right) \\
&= (n_{r-1}) \uparrow \left(\frac{1}{2^{r-1} - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{(r-2) \cdot 2^{r-1} + 1}{(2^r - 1) \cdot (2^{r-1} - 1)}\right).
\end{aligned} \tag{6.12}$$

Next, we use this to simplify the expression for the term s_r/k_r in Eq (6.10).

$$\begin{aligned}
\frac{s_r}{k_r} &= \underbrace{(n_r) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1}\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}}_{s_r} \cdot \underbrace{\frac{1}{b_r \cdot (1-\alpha)}}_{1/k_r} \\
&= \frac{n_r}{b_r} \cdot (n_r) \uparrow \left(\frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1} - 1\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3} \\
&= n_{r-1} \cdot (n_r) \uparrow \left(\frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1} - 1\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3} \\
&\qquad \text{(as } n_r = n_{r-1} \cdot b_r \text{ by Eq (6.9))} \\
&= (n_{r-1}) \uparrow \left(1 + \frac{1}{2^{r-1} - 1}\right) \cdot (1-\alpha) \uparrow (x) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\mathrm{JPY}} \cdot c_{\mathrm{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}. \\
&\qquad \text{(by Eq (6.12))}
\end{aligned}$$

where the exponent of $(1-\alpha)$ in the above equation is

$$\begin{aligned}
x &= \frac{(r-2)\cdot 2^{r-1}+1}{(2^r-1)\cdot(2^{r-1}-1)} + \frac{r\cdot 2^r}{2^r-1} - 1 \\
&= \frac{(r-2)\cdot(2^{r-1}-1)+1+r-2}{(2^r-1)\cdot(2^{r-1}-1)} + \frac{(r-1)\cdot 2^r+1}{2^r-1} \\
&= \frac{1}{2^r-1}\cdot\left((r-1)2^r+1+(r-2)+\frac{r-1}{2^{r-1}-1}\right) \\
&= \frac{(r-1)}{2^r-1}\cdot(2^r+1+\frac{1}{2^{r-1}-1}) \\
&= \frac{(r-1)}{2^r-1}\cdot(\frac{2^{2r-1}-2^r+2^{r-1}-1+1}{2^{r-1}-1}) \\
&= \frac{(r-1)}{2^r-1}\cdot(\frac{2^{2r-1}-2^{r-1}}{2^{r-1}-1}) \\
&= \frac{(r-1)\cdot 2^{r-1}}{2^{r-1}-1}\ .
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}
\frac{s_r}{k_r} &= (n_{r-1})\uparrow\left(1+\frac{1}{2^{r-1}-1}\right)\cdot(1-\alpha)\uparrow\left(\frac{(r-1)\cdot 2^{r-1}}{2^{r-1}-1}\right)\cdot\frac{(\alpha/(1-\alpha))^2}{(c_{\text{JPY}}\cdot c_{\text{ADV}}^2\cdot 80^2)^r\cdot((r+1)!)^3} \\
&= s_{r-1}\cdot\frac{1}{c_{\text{JPY}}\cdot c_{\text{ADV}}^2\cdot 80^2\cdot(r+1)^3}.
\end{aligned}$$

By plugging this back into Eq (6.10), we obtain

$$\begin{aligned}
\text{CC}(\pi_{r-1}) &< c_{\text{JPY}}\cdot(40\cdot c_{\text{ADV}}\cdot(r+1)^2\cdot r)\cdot\left(s_{r-1}\cdot\frac{1}{c_{\text{JPY}}\cdot c_{\text{ADV}}^2\cdot 80^2\cdot(r+1)^3}+r\right) \\
&\le \frac{1}{c_{\text{ADV}}^2\cdot 160}\cdot s_{r-1}+40\cdot c_{\text{JPY}}\cdot c_{\text{ADV}}\cdot(r+1)^4 \qquad \text{(using } r\le r+1\text{)} \\
&\le s_{r-1}\ , \qquad \text{(as } r^4 \ll s_{r-1} \text{ for } r=O(\log\log n) \text{ and } \alpha\le 1-n_r^{-1/2r}\text{)}
\end{aligned}$$

which is our desired bound on the communication cost of π_{r-1}.

To bound the probability of success, we first reiterate the following useful relation from Eq (6.9):

$$\begin{aligned}
b_r^2 &= (n_r)\uparrow\left(\frac{2\cdot 2^{r-1}}{2^r-1}\right)\cdot(1-\alpha)\uparrow\left(\frac{r\cdot 2\cdot 2^{r-1}}{2^r-1}-2\right) \\
&= (n_r)\uparrow\left(1+\frac{1}{2^r-1}\right)\cdot(1-\alpha)\uparrow\left(\frac{r\cdot 2^r}{2^r-1}\right)/(1-\alpha)^2\ . \qquad (6.13)
\end{aligned}$$

Using this, we now simplify the term $s_r/(2 \cdot b_r^2 \cdot \alpha^2)$ in Eq (6.11).

$$\begin{aligned}
\frac{s_r}{2 \cdot b_r^2 \cdot \alpha^2} &= \underbrace{(n_r) \uparrow \left(1 + \frac{1}{2^r - 1}\right) \cdot (1-\alpha) \uparrow \left(\frac{r \cdot 2^r}{2^r - 1}\right) \cdot \frac{(\alpha/(1-\alpha))^2}{(c_{\text{JPY}} \cdot c_{\text{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}}_{s_r} \cdot \frac{1}{2 \cdot b_r^2 \cdot \alpha^2} \\
&= \frac{b_r^2 \cdot \alpha^2}{(c_{\text{JPY}} \cdot c_{\text{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3} \cdot \frac{1}{2 \cdot b_r^2 \cdot \alpha^2} && \text{(by Eq (6.13))} \\
&= \frac{1}{2 \cdot (c_{\text{JPY}} \cdot c_{\text{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}\,.
\end{aligned}$$

By plugging this back into Eq (6.11), we have that

$$\begin{aligned}
\text{suc}(\pi_{r-1}) &\geq \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\text{ADV}} \cdot (r+1)}\right) - \frac{1}{40 \cdot c_{\text{ADV}} \cdot (r+1)^2} - \sqrt{\frac{1}{2 \cdot (c_{\text{JPY}} \cdot c_{\text{ADV}}^2 \cdot 80^2)^r \cdot ((r+1)!)^3}} \\
&\geq \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\text{ADV}} \cdot (r+1)}\right) - \frac{1}{40 \cdot c_{\text{ADV}} \cdot (r+1)^2} - \sqrt{\frac{1}{c_{\text{ADV}}^2 \cdot 80^2 \cdot (r+1)^4}} \\
&\qquad \text{(since } c_{\text{JPY}} \geq 1 \text{ and } 2 \cdot ((r+1)!)^3 \geq (r+1)^4 \text{ for } r \geq 1) \\
&= \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\text{ADV}} \cdot (r+1)}\right) - \frac{1}{c_{\text{ADV}} \cdot 80 \cdot (r+1)^2} \\
&= \frac{1}{2} \cdot \left(1 + \frac{r}{20 \cdot c_{\text{ADV}} \cdot (r+1)} - \frac{1}{40 \cdot c_{\text{ADV}} \cdot (r+1)^2}\right) \\
&\geq \frac{1}{2} \cdot \left(1 + \frac{r-1}{20 \cdot c_{\text{ADV}} \cdot r}\right),
\end{aligned}$$

which is our desired bound.

Overall, we have that the deterministic protocol π_{r-1} constructed from π_r using Lemma 6.2.11 contradicts the lemma for $r-1$ (inductive hypothesis) and thus the lemma must hold for deterministic r-round protocols when the input is sampled from the distribution $\mathcal{D}_r$ for every $r \geq 1$. Finally, by the easy direction of Yao's minmax principle, the same result holds for any (even randomized) r-round protocol for any instance of AHM_r (given a uniform random search sequence). ∎

6.2.2.2 The Round Elimination Argument: Proof of Lemma 6.2.11

We now get to the main part of the argument, which is the proof of Lemma 6.2.11, restated below.

Lemma 6.2.11 (restated). *For any $r \geq 1$, sufficiently large $n_r \in \mathbb{N}$, $\alpha, \delta \in (0,1)$, and integer $s \geq 1$, suppose there exists a deterministic r-round protocol π_r for $\mathsf{AHM}_r(n_r, \alpha)$ with*

$$CC(\pi_r) \leq s$$

and probability of success at least δ when the input is sampled from $\mathcal{D}_r(n_r, \alpha)$. Then, for any

$\varepsilon \in (0,1)$, *there exists a deterministic* $(r-1)$*-round protocol* π_{r-1} *for* AHM_{r-1} *with*

$$CC(\pi_{r-1}) \leq c_{\text{JPY}} \cdot (r/\varepsilon) \cdot \left(\frac{s}{k_r} + r\right),$$

and probability of success

$$\text{suc}(\pi_{r-1}) \geq \delta - \varepsilon - \sqrt{\frac{s}{2 \cdot b_r^2 \cdot \alpha^2}}$$

when the players' input is sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$.

Starting with a deterministic r-round protocol π_r for $\mathsf{AHM}_r(n_r, \alpha)$ where $\text{CC}(\pi_r) \leq s$ and $\text{suc}(\pi_r) \geq \delta$ under $\mathcal{D}_r$ as in the statement of Lemma 6.2.11, we construct a deterministic $(r-1)$-round protocol π_{r-1} for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ that proves the lemma in three steps:

(*i*) **(Input Embedding)** Construct a randomized r-round protocol $\pi_r^{(1)}$ under $\mathcal{D}_{r-1}$ with success probability the same as π_r under $\mathcal{D}_r$, i.e., $\text{suc}(\pi_r^{(1)}) \geq \delta$, and internal information cost less than the communication cost of π_r by a k_r multiplicative factor, i.e.,

$$\text{IC}(\pi_r^{(1)}, \mathcal{D}_{r-1}) \leq \frac{s}{k_r}.$$

(*ii*) **(Message Compression)** Construct a randomized r-round protocol $\pi_r^{(2)}$ under $\mathcal{D}_{r-1}$ with success probability less than that of $\pi_r^{(1)}$ under $\mathcal{D}_{r-1}$ by at most an additive ε factor, i.e., $\text{suc}(\pi_r^{(2)}) \geq \delta - \varepsilon$, and with communication cost similar to the internal information cost of $\pi_r^{(1)}$ under $\mathcal{D}_{r-1}$, i.e.,

$$\text{CC}(\pi_r^{(2)}) \leq c_{\text{JPY}} \cdot \frac{r}{\varepsilon} \cdot \left(\text{IC}(\pi_r^{(1)}, \mathcal{D}_{r-1}) + r\right) \leq c_{\text{JPY}} \cdot \frac{r}{\varepsilon} \cdot \left(\frac{s}{k_r} + r\right).$$

(*iii*) **(Guessing the First Message)** Construct a deterministic $(r-1)$-round protocol π_{r-1} under $\mathcal{D}_{r-1}$ with success probability less than that of $\pi_r^{(2)}$ under $\mathcal{D}_{r-1}$ by at most an additive $\simeq \sqrt{s/(b_r^2 \cdot \alpha^2)}$ factor, i.e, $\text{suc}(\pi_{r-1}) \geq \delta - \varepsilon - \sqrt{s/(2 \cdot b_r^2 \cdot \alpha^2)}$, and with communication cost no more than that of $\pi_r^{(2)}$, i.e.,

$$\text{CC}(\pi_{r-1}) \leq \text{CC}(\pi_r^{(2)}) \leq c_{\text{JPY}} \cdot \frac{r}{\varepsilon} \cdot \left(\frac{s}{k_r} + r\right).$$

Step (*i*): A Low Information Cost Protocol via Input Embedding

We construct a low information cost r-round protocol $\pi_r^{(1)}$ for AHM_{r-1} by considering an input $(A^{(r-1)}, B^{(r-1)}) \sim \mathcal{D}_{r-1}(n_{r-1}, \alpha)$ and embedding it in a simulated input instance $(A^{(r)}, B^{(r)}) \sim \mathcal{D}_r(n_r, \alpha)$ for AHM_r such that its output can be used to solve $(A^{(r-1)}, B^{(r-1)})$. Then, the players simulate a run of the protocol π_r in r rounds using the simulated input.

To simplify the exposition, we disambiguate the players in the construction of $\pi_r^{(1)}$ (and π_{r-1} later in step (*iii*)). We use Alice and Bob to denote the players in the input instance of AHM_{r-1} where Alice holds $A^\star = A^{(r-1)}$ and Bob holds $B^\star = B^{(r-1)}$. On the other hand,

we use Q_X and Q_Y to refer to the players in the simulated input instance of AHM_r where player Q_X holds $A^{(r)} = X$ and player Q_Y holds $B^{(r)} = (\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}, Y)$.

An r-round protocol $\pi_r^{(1)}$ for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ on input $(A^\star, B^\star) \sim \mathcal{D}_{r-1}(n_{r-1}, \alpha)$ where $(k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_1)$ is the given uniform random search sequence:

(a) Using public randomness, independently sample a uniform random search index $k^\star \in [k_r]$. Then, jointly sample the following from $\mathcal{D}_r(n_r, \alpha)$:

- The uniform random permutations $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}$ of $[b_r]$;
- Bob's part of all the $k_r^2 - k_r$ many off-diagonal sub-instances

$$B^{\mathrm{OFF}} = (B^{(r-1)}_{i,j} : b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r);$$

- Alice's part of $k^\star - 1$ many special sub-instances $A^{\mathrm{SPEC}}_{<k^\star} = (A^{\mathrm{SPEC}}_k : 1 \leq k < k^\star)$;
- Bob's part of $k_r - k^\star$ many special sub-instances $B^{\mathrm{SPEC}}_{>k^\star} = (B^{\mathrm{SPEC}}_k : k^\star < k \leq k_r)$.

Let $i^\star, j^\star \in [b_r]$ be such that $\sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) = b_r \cdot \alpha + k^\star$, which both players can compute since $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}, k^\star$ are sampled using public randomness.

(b) Bob takes on the role of player Q_X and sets $X[i^\star, j^\star] = B^\star$. Bob privately samples X from $\mathcal{D}_r(n_r, \alpha)$ conditioned on all the random variables from step (a) and $X[i^\star, j^\star]$, i.e., the remainder of Q_X's input $A^{(r)}$.

(c) Alice takes on the role of player Q_Y and sets $Y[i^\star, j^\star] = A^\star$. Alice privately samples Y from $\mathcal{D}_r(n_r, \alpha)$ conditioned on all the variables from step (a) and $Y[i^\star, j^\star]$, i.e., the remainder of player Q_Y's input $B^{(r)}$.

(d) The players Q_X (= Bob) and Q_Y (= Alice) simulate a run of protocol π_r in r rounds using their respective inputs $A^{(r)} = X$ and $B^{(r)} = (\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}, Y)$ and starting with player Q_X.

(e) At the end of the protocol, the player that receives the final message gets the uniform random search sequence $(k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$ and then returns the answer of π_r on the search sequence $(k^\star, k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$.

Observation 6.2.12. *In protocol $\pi_r^{(1)}$, $(A^{(r)}, B^{(r)})$ in the simulation of π_r is sampled from $\mathcal{D}_r(n_r, \alpha)$ and $(k^\star, k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$ is a uniform random search sequence.*

Proof. The input to the simulation of π_r is $(A^{(r)}, B^{(r)})$ where $A^{(r)} = X$ and $B^{(r)} = (\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}, Y)$. All variables of the input are jointly sampled from $\mathcal{D}_r(n_r, \alpha)$ except for Alice and Bob's embedded instance $(A^\star, B^\star) \sim \mathcal{D}_{r-1}(n_{r-1}, \alpha)$ where $X[i^\star, j^\star] = B^\star$ and $Y[i^\star, j^\star] = A^\star$. Since the instance of AHM_{r-1} corresponding to $(X[i^\star, j^\star], Y[i^\star, j^\star])$ is independent of all other variables in the distribution $\mathcal{D}_r(n_r, \alpha)$, all other variables are jointly distributed according to $\mathcal{D}_r(n_r, \alpha)$. It remains to show that $(X[i^\star, j^\star], Y[i^\star, j^\star])$ distributed according to

$\mathcal{D}_{r-1}(n_{r-1}, \alpha)$, which is required in $\mathcal{D}_r(n_r, \alpha)$. This is immediate from the embedding as $(X[i^\star, j^\star], Y[i^\star, j^\star]) = (A^\star, B^\star) \sim \mathcal{D}_{r-1}(n_{r-1}, \alpha)$

Each search index in $(k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$ is uniformly sampled. Then, since $k^\star \in [k_r]$ is also uniformly sampled, we have $(k^\star, k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$ is a random search sequence as required. ∎

With this observation, we have that Alice and Bob have successfully simulated π_r for $\mathsf{AHM}_r(n_r, \alpha)$ on the correct distribution, as the search sequence $(k^\star, k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_0)$ is uniformly random and the players' inputs are distributed according to $\mathcal{D}_r(n_r, \alpha)$. Since the messages communicated by the players are exactly the messages of the simulated protocol π_r, the communication cost of $\pi^{(1)}_r$ is the same as π_r. Although the communication cost is large, we now show that its internal information cost (about input $(A^\star, B^\star)$) is smaller by a multiplicative k_r factor.

Notation. We use $\mathsf{A}^\star, \mathsf{B}^\star$ to denote the random variables corresponding to the input instance of AHM_{r-1} given to Alice and Bob, which are distributed according to $\mathcal{D}_{r-1}$. We use $\mathsf{\Pi}$ to denote the random variable corresponding to the set of messages sent by both Alice and Bob. For $i \in [r]$, Π_i denotes the message sent in round i and $\mathsf{\Pi}_i$ denotes the random variable corresponding to Π_i. We use $\mathsf{A}^{\text{OFF}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{A}^{\text{SPEC}}, \mathsf{B}^{\text{SPEC}}$ to denote the random variables corresponding to $A^{\text{OFF}}, B^{\text{OFF}}$, $k^\star$, $A^{\text{SPEC}} = (A^{\text{SPEC}}_1, A^{\text{SPEC}}_2, \ldots, A^{\text{SPEC}}_{k_r})$, $B^{\text{SPEC}} = (B^{\text{SPEC}}_1, \ldots, B^{\text{SPEC}}_{k_r})$, respectively.

The following claim proves that information cost of $\pi^{(1)}_r$ about its input sampled from $\mathcal{D}_{r-1}$ is $1/k_r$ times smaller than the information cost of π_r about its own input sampled from $\mathcal{D}_r$.

Claim 6.2.13. *In protocol $\pi^{(1)}_r$, we have that*

$$IC(\pi^{(1)}_r, \mathcal{D}_{r-1}(n_{r-1}, \alpha)) \le \frac{1}{k_r} \cdot IC(\pi_r, \mathcal{D}_r(n, \alpha)).$$

Proof. By the definition of internal information cost (Definition 6.2.2),

$$\begin{aligned} \mathrm{IC}(\pi^{(1)}_r, \mathcal{D}_{r-1}(n_{r-1}, \alpha)) = \mathbb{I}(\mathsf{A}^\star; \mathsf{\Pi} \mid \underbrace{\boldsymbol{\sigma}_{\mathrm{R}}, \boldsymbol{\sigma}_{\mathrm{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}}_{\text{public randomness of } \pi^{(1)}_r}, \mathsf{B}^\star) \\ + \mathbb{I}(\mathsf{B}^\star; \mathsf{\Pi} \mid \underbrace{\boldsymbol{\sigma}_{\mathrm{R}}, \boldsymbol{\sigma}_{\mathrm{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}}_{\text{public randomness } \pi^{(1)}_r}, \mathsf{A}^\star). \end{aligned} \tag{6.14}$$

Each mutual information term corresponds to the amount of information communicated by each player about their input, and we bound them separately.

For Alice's mutual information term in Eq (6.14), we have that

$$\begin{aligned} &\mathbb{I}(\mathsf{A}^\star; \mathsf{\Pi} \mid \boldsymbol{\sigma}_{\mathrm{R}}, \boldsymbol{\sigma}_{\mathrm{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^\star) \\ &\quad = \frac{1}{k_r} \cdot \sum_{k \in [k_r]} \mathbb{I}(\mathsf{A}^\star; \mathsf{\Pi} \mid \boldsymbol{\sigma}_{\mathrm{R}}, \boldsymbol{\sigma}_{\mathrm{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star = k, \mathsf{B}^{\text{SPEC}}_{>k}, \mathsf{A}^{\text{SPEC}}_{<k}, \mathsf{B}^\star) \end{aligned}$$

(by the definition of conditional mutual information and the uniform distribution of $\mathsf{k}^\star$)

$$= \frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{A}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star = k, \mathsf{B}_{\geq k}^{\text{SPEC}}, \mathsf{A}_{<k}^{\text{SPEC}})$$

(as $\mathsf{A}^\star = \mathsf{A}_k^{\text{SPEC}}, \mathsf{B}^\star = \mathsf{B}_k^{\text{SPEC}}$ in protocol $\pi_r^{(1)}$ when $\mathsf{k}^\star = k$)

$$= \frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{A}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}_{\geq k}^{\text{SPEC}}, \mathsf{A}_{<k}^{\text{SPEC}});$$

the last part holds because the joint distribution of $(\mathsf{A}_{\leq k}^{\text{SPEC}}, \mathsf{B}_{\geq k}^{\text{SPEC}}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \Pi)$ is independent of the value of $\mathsf{k}^\star$. In particular, if we sample any input $(A^{(r)}, B^{(r)})$ from $\mathcal{D}_r(n, \alpha)$ (which does not include $k^\star = k_r^\star$), the values of $\boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}_{\leq k}^{\text{SPEC}}, \mathsf{B}_{\geq k}^{\text{SPEC}}$ are fixed and the value of Π is also fixed as π_r is deterministic. Hence, the distribution of $\mathsf{k}^\star$ remains uniform over $[k_r]$ irrespective of the instance $(A^{(r)}, B^{(r)})$. Continuing the bounding of Alice's mutual information term, we have that

$$\frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{A}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}_{\geq k}^{\text{SPEC}}, \mathsf{A}_{<k}^{\text{SPEC}})$$

$$\leq \frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{A}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}, \mathsf{A}_{<k}^{\text{SPEC}})$$

(by Proposition 4.3.2 as $\mathsf{B}_{<k}^{\text{SPEC}} \perp \mathsf{A}_k^{\text{SPEC}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}_{<k}^{\text{SPEC}}$ by the definition of distribution $\mathcal{D}_r$)

$$= \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{A}^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}})$$

(by the chain rule of mutual information, Fact 4.3.1-(7))

$$\leq \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{A}^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{REST}})$$

(by Proposition 4.3.2, as $\mathsf{A}^{\text{SPEC}} \perp \mathsf{B}^{\text{REST}} \mid \mathsf{B}^{\text{SPEC}}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}$ by the definition of distribution $\mathcal{D}_r$)

$$= \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{A}^{\text{SPEC}}, \mathsf{B}^{\text{OFF}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{REST}}) \qquad \text{(as } \mathsf{B}^{\text{OFF}} \text{ is fixed)}$$

$$\leq \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{A}^{\text{SPEC}}, \mathsf{A}^{\text{OFF}}, \mathsf{B}^{\text{OFF}}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}; \Pi \mid \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{REST}})$$

(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))

$$= \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{B}^{(r)}; \Pi \mid \mathsf{A}^{(r)}).$$

(recall that input of Q_Y (resp. Q_X) simulated by Alice (resp. Bob) in $\pi_r^{(1)}$ is $B^{(r)}$ (resp. $A^{(r)}$))

For Bob's mutual information term in Eq (6.14), following a similar argument, we have

$$\mathbb{I}(\mathsf{B}^\star; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}_{>\mathsf{k}^\star}^{\text{SPEC}}, \mathsf{A}_{<\mathsf{k}^\star}^{\text{SPEC}}, \mathsf{A}^\star)$$

$$= \frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{B}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star = k, \mathsf{B}_{>k}^{\text{SPEC}}, \mathsf{A}_{\leq k}^{\text{SPEC}})$$

(by definition of conditional mutual information and $\mathsf{A}^\star, \mathsf{B}^\star$ and the uniform distribution of $\mathsf{k}^\star$)

$$= \frac{1}{k_r} \cdot \sum_{k\in[k_r]} \mathbb{I}(\mathsf{B}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}_{>k}^{\text{SPEC}}, \mathsf{A}_{\leq k}^{\text{SPEC}})$$

(as in the previous case for Alice, for any $k \in [k_r]$, $(\mathsf{k}^\star = k) \perp (\mathsf{B}_{\geq k}^{\text{SPEC}}, \mathsf{A}_{\leq k}^{\text{SPEC}}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}})$)

$$\leq \frac{1}{k_r} \cdot \sum_{k \in [k_r]} \mathbb{I}(\mathsf{B}_k^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}_{>k}^{\text{SPEC}}, \mathsf{A}^{\text{SPEC}})$$

(by Proposition 4.3.2, as $\mathsf{A}_{>k}^{\text{SPEC}} \perp \mathsf{B}_k^{\text{SPEC}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}_{>k}^{\text{SPEC}}$ by the definition of distribution $\mathcal{D}_r$)

$$= \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{B}^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}})$$

(by chain rule of mutual information, Fact 4.3.1-(7))

$$\leq \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{B}^{\text{SPEC}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{A}^{\text{OFF}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}})$$

(by Proposition 4.3.2, as $\mathsf{A}^{\text{OFF}} \perp \mathsf{B}^{\text{SPEC}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}$ by the definition of distribution $\mathcal{D}_r$)

$$\leq \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{REST}}; \Pi \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{A}^{\text{OFF}}, \mathsf{B}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}})$$

(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))

$$= \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{A}^{(r)}; \Pi \mid \mathsf{B}^{(r)}).$$

(recall that input of Q_X (resp. Q_Y) simulated by Bob (resp. Alice) in $\pi_r^{(1)}$ is $A^{(r)}$ (resp. $B^{(r)}$))

With both these bounds, we can upper bound the LHS of Eq (6.14) and obtain

$$\begin{aligned} \text{IC}(\pi_r^{(1)}, \mathcal{D}_{r-1}) &\leq \frac{1}{k_r} \cdot \left(\mathbb{I}(\mathsf{B}^{(r)}; \Pi \mid \mathsf{A}^{(r)}) + \mathbb{I}(\mathsf{A}^{(r)}; \Pi \mid \mathsf{B}^{(r)}) \right) \\ &= \frac{1}{k_r} \cdot \text{IC}(\pi_r, \mathcal{D}_r), \end{aligned}$$

where the last step follows since the messages communicated in $\pi_r^{(1)}$ are exactly the messages communicated in the simulation of π_r and since $(\mathsf{A}^{(r)}, \mathsf{B}^{(r)}) \sim \mathcal{D}_r$ in $\pi_r^{(1)}$ by Observation 6.2.12. ∎

We now obtain the following main lemma of this step of the argument:

Lemma 6.2.14. *Protocol $\pi_r^{(1)}$ is an r-round protocol for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ where Bob speaks first with communication cost at most s, probability of success at least δ, and internal information cost at most s/k_r when the input is sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$.*

Proof. In protocol $\pi_r^{(1)}$, the instance $(A^{(r)}, B^{(r)})$ used to simulate π_r is constructed without any communication, so the total number of bits communicated in $\pi_r^{(1)}$ is at most the total number of bits communicated in π_r, which is $\text{CC}(\pi_r) \leq s$ as in the statement of Lemma 6.2.11. Bob takes on the role of Q_X in the simulation and thus Bob speaks first.

By construction, protocol $\pi_r^{(1)}$ outputs the same answer as π_r since the solution to $(A^{(r)}, B^{(r)}) \sim \mathcal{D}_r$ on search sequence $(k^\star, k_{r-1}^\star, k_{r-2}^\star, \ldots, k_0^\star)$ is the same as the solution to the input $(A_{k^\star}^{\text{SPEC}}, B_{k^\star}^{\text{SPEC}}) = (A^\star, B^\star) \sim \mathcal{D}_{r-1}$ on the search sequence $(k_{r-1}^\star, k_{r-2}^\star, \ldots, k_0^\star)$. Therefore, $\pi_r^{(1)}$ succeeds with probability at least δ. Finally, the bound on the internal information cost of $\pi_r^{(1)}$ follows from Claim 6.2.13 and the fact that the communication cost of π_r, namely, s, is an upper bound of its internal information cost. ∎

Step (*ii*): A Low Communication Cost Protocol via Message Compression

In this step, we compress the communication cost of $\pi_r^{(1)}$ using standard message compression techniques. As a direct corollary of Proposition 6.2.4, we get the following:

Corollary 6.2.15. *For any $0 < \varepsilon < 1$, there exists an r-round protocol $\pi_r^{(2)}$ for $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ where Bob speaks first with*

$$CC(\pi_r^{(2)}) \leq c_{\text{JPY}} \cdot (r/\varepsilon) \cdot \left(\frac{s}{k_r} + r\right).$$

and probability of success at least $\delta - \varepsilon$ when the input is sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$.

Proof. We use Proposition 6.2.4 on protocol $\pi_r^{(1)}$ to get protocol $\pi_r^{(2)}$ such that the simulation fails with probability at most ε. Then, we have that

$$\text{CC}(\pi_r^{(2)}) \leq c_{\text{JPY}} \cdot \left(r/\varepsilon \cdot \text{IC}(\pi_r^{(1)}, \mathcal{D}_{r-1}) + r^2/\varepsilon\right) \leq c_{\text{JPY}} \cdot (r/\varepsilon) \cdot \left(\frac{s}{k_r} + r\right). \quad \text{(by Lemma 6.2.14)}$$

The simulation of $\pi_r^{(1)}$ using compressed messages fails with probability at most ε, so $\pi_r^{(2)}$ succeeds with probability at least $\delta - \varepsilon$, as desired. ∎

Since this step only compresses the messages sent in $\pi_r^{(1)}$, protocol $\pi_r^{(2)}$ is constructed in the same way as $\pi_r^{(1)}$ with an additional message compression step that uses an independent source of public randomness. In particular, $\pi_r^{(2)}$ is a simulation of π_r on an instance of AHM_r sampled from $\mathcal{D}_r$, but with communication cost smaller by a $\simeq k_r$ multiplicative factor and with a slightly smaller probability of success.

Step (*iii*): An $(r-1)$-Round Protocol via Guessing the First Message

In this final and the most important step, we take the r-round protocol $\pi_r^{(2)}$ for AHM_{r-1} where Bob communicates first and we eliminate the first message by making the players *guess* it using public randomness, thus beginning the protocol from the second round. Therefore, we obtain an $(r-1)$-round protocol π_{r-1} for AHM_{r-1} where Alice communicates first as required. This alters the joint distribution of the input and thus affects the guarantees of the protocol (as in Corollary 6.2.15), but we will show that the effect is small.

An $(r-1)$-round protocol π_{r-1} for AHM_{r-1} on inputs $(A^{(r-1)}, B^{(r-1)}) \sim \mathcal{D}_{r-1}(n_{r-1}, \alpha)$ where $(k^\star_{r-1}, k^\star_{r-2}, \ldots, k^\star_1)$ is the given uniform random search sequence:

(a) Using public randomness, jointly sample the first message Π_1 and the search index $k^\star$ from the protocol $\pi_r^{(2)}$. Then, jointly sample the following from $\mathcal{D}_r(n_r, \alpha)$ conditioned on $\Pi_1, k^\star$:

- The uniform random permutations $\sigma_{\text{R}}, \sigma_{\text{C}}$ of $[b_r]$;

- Bob's part of all the $k_r^2 - k_r$ many off-diagonal sub-instances

$$B^{\text{OFF}} = (B_{i,j}^{(r-1)} : b_r \cdot \alpha < \sigma_{\text{R}}(i) \neq \sigma_{\text{C}}(j) \leq b_r);$$

- Alice's part of the $k^\star - 1$ many special sub-instances $A_{<k^\star}^{\text{SPEC}} = (A_k^{\text{SPEC}} : 1 \leq k < k^\star)$;
- Bob's part of the $k_r - k^\star$ many special sub-instances $B_{>k^\star}^{\text{SPEC}} = (B_k^{\text{SPEC}} : k^\star < k \leq k_r)$.

Let $i^\star, j^\star \in [b_r]$ be such that $\sigma_{\text{R}}(i) = \sigma_{\text{C}}(j) = b_r \cdot \alpha + k^\star$, which both players can compute since $\sigma_{\text{R}}, \sigma_{\text{C}}, k^\star$ are sampled using public randomness.

(b) Bob takes on the role of player Q_X and sets $X[i^\star, j^\star] = B^\star$. Bob privately samples X from $\mathcal{D}_r(n_r, \alpha)$ conditioned on all the random variables from step (a) and $X[i^\star, j^\star]$, i.e., the remainder of Q_X's input $A^{(r)}$.

(c) Alice takes on the role of player Q_Y and sets $Y[i^\star, j^\star] = A^\star$. Alice privately samples Y from $\mathcal{D}_r(n_r, \alpha)$ conditioned on all the variables from step (a) and $Y[i^\star, j^\star]$, i.e., the remainder of player Q_Y's input $B^{(r)}$.

(d) The players Q_X (= Bob) and Q_Y (= Alice) simulate protocol $\pi_r^{(2)}$ using their respective inputs $A^{(r)} = X, B^{(r)} = (\sigma_{\text{R}}, \sigma_{\text{C}}, Y)$ and the search sequence $(k^\star, k_{r-1}^\star, k_{r-2}^\star, \ldots, k_1^\star)$. They use Π_1 as a guess of the first message sent from Q_X to Q_Y and thus only simulate $r-1$ rounds of $\pi_r^{(2)}$ starting with the second round, i.e., player Q_Y communicates first.

Observe that, similar to the construction of $\pi_r^{(2)}$ (which is identical to $\pi_r^{(1)}$, except for the message compression), protocol π_{r-1} embeds the input for AHM_{r-1} sampled from $\mathcal{D}_{r-1}$ into a simulated instance $(A^{(r)}, B^{(r)})$ of AHM_r. Then, the players simulate a run of $\pi_r^{(2)}$ using $(A^{(r)}, B^{(r)})$ where, to remove the first round of communication, the first message is guessed using public randomness. However, due to the random guessing of the first message, the simulated instance is no longer distributed according to $\mathcal{D}_r(n_r, \alpha)$ (recall Observation 6.2.12), i.e., the guarantees of $\pi_r^{(2)}$ do not hold for π_{r-1}. Despite this, we show that the simulated instance and first message are statistically close to being distributed as they are in $\pi_r^{(2)}$ and thus similar guarantees hold.

Let $\mathcal{D}^{\text{FAKE}}$ denote the joint distribution of $(\mathsf{A}^{(r)}, \mathsf{B}^{(r)}, \Pi_1, \mathsf{k}^\star)$ as it is in π_{r-1} and let $\mathcal{D}^{\text{REAL}}$ denote the joint distribution of $(\mathsf{A}^{(r)}, \mathsf{B}^{(r)}, \Pi_1, \mathsf{k}^\star)$ as it is in $\pi_r^{(2)}$. Let

$$\mathsf{R}_a := (\Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}_{>\mathsf{k}^\star}^{\text{SPEC}}, \mathsf{A}_{<\mathsf{k}^\star}^{\text{SPEC}})$$

denote the public randomness used in step (a) of π_{r-1}. We show in Claim 6.2.16 that the distributions $\mathcal{D}^{\text{FAKE}}$ and $\mathcal{D}^{\text{REAL}}$ only differ in the way that the input $(\mathsf{A}^\star, \mathsf{B}^\star) = (\mathsf{A}_{\mathsf{k}^\star}^{\text{SPEC}}, \mathsf{B}_{\mathsf{k}^\star}^{\text{SPEC}})$ is sampled.

Claim 6.2.16. *We have that*

$$\mathcal{D}^{\text{FAKE}} = \text{dist}(\mathsf{R}_a) \times \quad \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) \quad \times \text{dist}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}) \times \text{dist}(\mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}),$$
$$\mathcal{D}^{\text{REAL}} = \text{dist}(\mathsf{R}_a) \times \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a) \times \text{dist}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}) \times \text{dist}(\mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}).$$

Proof. We first consider $(\mathsf{A}^{(r)}, \mathsf{B}^{(r)}, \Pi_1, \mathsf{k}^\star)$ directly as it is sampled in π_{r-1}. The random variable R_a in π_{r-1} is jointly sampled in step (*a*). Then, Alice and Bob embed $(\mathsf{A}^\star, \mathsf{B}^\star) \sim \mathcal{D}_{r-1}$ as $\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star} = \mathsf{A}^\star$ and $\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} = \mathsf{B}^\star$, i.e., $(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) \sim \mathcal{D}_{r-1}$, which is *not* conditioned on R_a. Finally, Bob and Alice privately sample the remainder of X and Y (and thus the remainder of $A^{(r)}$ and $B^{(r)}$), respectively, which is only conditioned on R_a and their respective embedded inputs. This defines $\mathcal{D}^{\text{FAKE}}$.

We then obtain $\mathcal{D}^{\text{REAL}}$ from $\mathcal{D}^{\text{FAKE}}$ by including all previously sampled random variables in the conditioning of the subsequently sampled random variables. That is, we obtain

$$\begin{aligned}\mathcal{D}^{\text{REAL}} &= \text{dist}(\mathsf{R}_a) \\ &\quad \times \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a) \\ &\quad \times \text{dist}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) \\ &\quad \times \text{dist}(\mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{\geq\mathsf{k}^\star}).\end{aligned}$$

It is easy to verify that this is exactly the distribution of $(\mathsf{A}^{(r)}, \mathsf{B}^{(r)}, \Pi_1, \mathsf{k}^\star)$ as it is in $\pi^{(2)}_r$.

Finally, we simplify the third and fourth terms in $\mathcal{D}^{\text{REAL}}$ to show that they are identical to the corresponding terms in $\mathcal{D}^{\text{FAKE}}$, which follows directly from the claims that

$$(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}) \perp \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star} \quad \text{and} \quad \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{REST}} \perp \mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{\geq\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star},$$

respectively. To prove these claims, it is sufficient to bound the corresponding mutual information terms as follows:

$$\begin{aligned}&\mathbb{I}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}) \\ &\leq \mathbb{I}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\leq\mathsf{k}^\star}, \mathsf{B}^{\text{REST}} \mid \mathsf{R}_a, \mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}) \\ &\qquad \text{(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))} \\ &= \mathbb{I}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\leq\mathsf{k}^\star}, \mathsf{B}^{\text{REST}} \mid \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{\leq\mathsf{k}^\star}) \quad \text{(by the definition of } \mathsf{R}_a) \\ &\leq \mathbb{I}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\leq\mathsf{k}^\star}, \mathsf{B}^{\text{REST}}, \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{\leq\mathsf{k}^\star}) \\ &\qquad \text{(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))} \\ &= \mathbb{I}(\mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{>\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\leq\mathsf{k}^\star}, \mathsf{B}^{\text{REST}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{\leq\mathsf{k}^\star}) \\ &\qquad\qquad \text{(as } \Pi_1 \text{ is fixed by } \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}) \\ &= 0, \qquad\qquad \text{(by Fact 4.3.1-(2) and construction of distribution } \mathcal{D}_r)\end{aligned}$$

and

$$\begin{aligned}&\mathbb{I}(\mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{REST}}; \mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{\geq\mathsf{k}^\star} \mid \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) \\ &\leq \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{REST}}, \Pi_1; \mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{\geq\mathsf{k}^\star} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{\geq\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{\leq\mathsf{k}^\star}) \\ &\qquad \text{(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))}\end{aligned}$$

$$= \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{REST}}; \mathsf{A}^{\text{OFF}}, \mathsf{A}^{\text{SPEC}}_{\geq \mathsf{k}^\star} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{\leq \mathsf{k}^\star})$$
(as Π_1 is fixed by $\mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}$)
$$= 0.$$ (by Fact 4.3.1-(2) and construction of distribution $\mathcal{D}_r$)

This completes the proof. ∎

We note here that the simulated instance $(A^{(r)}, B^{(r)})$ in $\mathcal{D}^{\text{REAL}}$ is sampled according to $\mathcal{D}_r(n_r, \alpha)$ (recall Observation 6.2.12). However, this is not true in $\mathcal{D}^{\text{FAKE}}$ since $(\mathsf{A}^\star, \mathsf{B}^\star) = (\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star})$ is sampled according to $= \mathcal{D}_{r-1}$ instead of $\mathcal{D}_r \mid \mathsf{R}_a$. Thus, to bound the total variation distance between $\mathcal{D}^{\text{REAL}}$ and $\mathcal{D}^{\text{FAKE}}$, we show that the information revealed by the random variable R_a about the input $(\mathsf{A}^\star, \mathsf{B}^\star)$ in $\pi^{(2)}_r$, i.e., under distribution $\mathcal{D}^{\text{REAL}}$, is not too large. We need an independence claim (Claim 6.2.17) and then a crucial bound on the information carried by the first message (Claim 6.2.18).

Claim 6.2.17.
$$\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star} \perp \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}.$$

Proof. Again, we bound the mutual information between the terms. Let R_{comp} represent the independent source of public randomness used in the message compression in step *(ii)*. Then,

$$\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
$$\leq \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(moving Π_1 by chain rule in Fact 4.3.1-(7), and non-negativity of mutual information Fact 4.3.1-(2))
$$\leq \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{REST}}, \mathsf{R}_{\text{comp}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(by data processing inequality in Fact 4.3.1-(8), Π_1 is fixed by $\mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{OFF}}, \mathsf{R}_{\text{comp}}$)
$$= \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}; \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{REST}} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(since R_{comp} is an entirely independent source of randomness)
$$\leq 0.$$ (by Fact 4.3.1-(2) and construction of $\mathcal{D}_r$) ∎

The following claim is the heart of the entire proof.

Claim 6.2.18. *For any $k \in [k_r]$,*
$$\mathbb{I}(\mathsf{B}^{\text{SPEC}}_k; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}_{<k}, \mathsf{B}^{\text{SPEC}}_{>k}) \leq \frac{s}{b_r^2 \cdot \alpha^2}.$$

Proof. Let $\rho_{\text{R}}, \rho_{\text{C}}$ be the inverse of $\sigma_{\text{R}}, \sigma_{\text{C}}$, and let $\boldsymbol{\rho}_{\text{R}}, \boldsymbol{\rho}_{\text{C}}$ denote the random variables corresponding to $\rho_{\text{R}}, \rho_{\text{C}}$ respectively. For ease of exposition, we represent any permutation ρ on $[b_r]$ as an ordered set $S := \{\rho(\ell) : \ell \in [b_r]\}$ with the ordering that $\rho(\ell) < \rho(\ell')$ for $\ell < \ell'$. Using this representation, we allow for partial definitions of permutations, i.e., ordered subsets $T \subset S$. In particular, we require the following partial definitions of ρ_{R} and ρ_{C}:

- Let T_{ROW} be an ordered set
$$T_{\text{ROW}} = \{\rho_{\text{R}}(\ell) \mid b_r \cdot \alpha < \ell \leq b_r, \ell \neq b_r \cdot \alpha + k\}$$

of size $b_r \cdot (1-\alpha) - 1$, which is the set of rows that corresponds to B^{OFF} except for the row corresponding to $\mathsf{B}_k^{\text{SPEC}}$. Let T_{ROW} denote its random variable.

- Let T_{COL} be an ordered set

$$T_{\text{COL}} = \{\rho_{\text{C}}(\ell) \mid b_r \cdot \alpha < \ell \leq b_r, \ell \neq b_r \cdot \alpha + k\}$$

of size $b_r \cdot (1-\alpha) - 1$, which is the set of columns that corresponds to B^{OFF} except for the column corresponding to $\mathsf{B}_k^{\text{SPEC}}$. Let T_{COL} denote its random variable.

We give an illustration of these definitions in Figure 6.5.

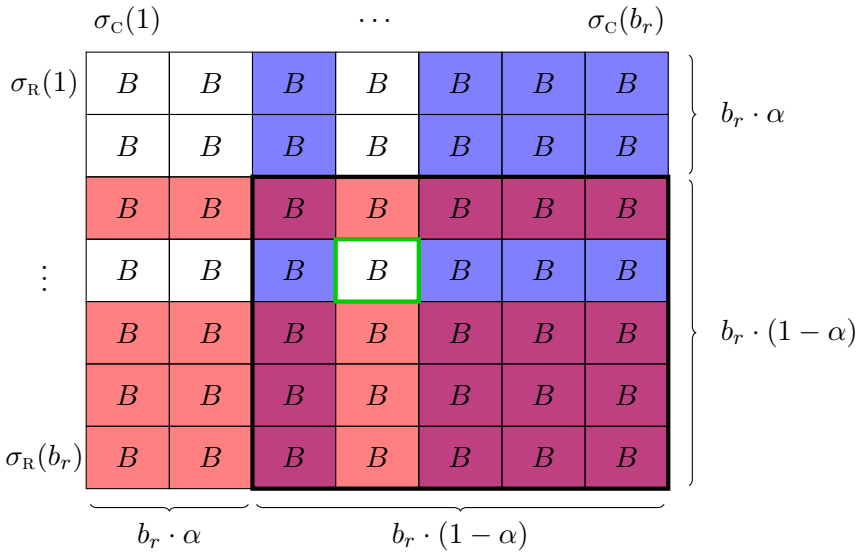


Figure 6.5: An illustration of the partial definitions T_{ROW} and T_{COL} for an instance of $\mathsf{AHM}_r(n_r, \alpha)$ where $b_r = 7$, $\alpha = 2/7$, and $k = 2$. In this figure, the $b_r \times b_r$ matrix is after applying the respective permutations of rows and columns using σ_{R} and σ_{C} held by Bob. In each position of the matrix, the corresponding input is the $B's$ from each instance of $\mathsf{AHM}_{r-1}(n_{r-1}, \alpha)$ that is held by Alice. T_{ROW} and T_{COL} are partial definitions of the permutations ρ_{R} and ρ_{C}, i.e., the inverses of σ_{R} and σ_{C}, respectively. T_{ROW} corresponds to fixing the rows filled red and T_{COL} corresponds to fixing the columns filled blue, which implies fixing positions filled purple. The unfilled (white) positions remain unrestricted by $T_{\text{ROW}}, T_{\text{COL}}$, i.e., the green outlined special position B_k^{SPEC} could still (uniformly) be in any of the unfilled positions when conditioned on $T_{\text{ROW}}, T_{\text{COL}}$ (instead of $\sigma_{\text{R}}, \sigma_{\text{C}}$).

This now allows us to write the indices $i, j \in [b_r]$ of the instances corresponding to B^{SPEC} and B^{OFF} using $\rho_{\text{R}}, \rho_{\text{C}}, T_{\text{ROW}}, T_{\text{COL}}$:

(i) For any $\ell \in [k_r]$, $\mathsf{B}_\ell^{\text{SPEC}}$ is the same as the random variable $\mathsf{B}_{\rho_{\text{R}}(b_r \cdot \alpha + \ell), \rho_{\text{C}}(b_r \cdot \alpha + \ell)}$ where we omit the superscript $r-1$ as it is clear from context;

(ii) $\mathsf{B}^{\text{OFF}}, \mathsf{B}_{<k}^{\text{SPEC}}, \mathsf{B}_{>k}^{\text{SPEC}}$ is the union of combinatorial rectangles formed by

$$(\mathsf{T}_{\text{ROW}} \times \mathsf{T}_{\text{COL}}), \quad (\mathsf{T}_{\text{ROW}} \times \{\boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)\}), \quad (\{\rho_{\text{R}}(b_r \cdot \alpha + k)\} \times \mathsf{T}_{\text{COL}}).$$

Finally, let Z be defined as,

$$\mathsf{Z} := (\mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}_{<k}, \mathsf{B}^{\text{SPEC}}_{>k}).$$

With these at hand, we can start bounding the mutual information term in the claim. Firstly, it is easy to see that,

$$\mathbb{I}(\mathsf{B}^{\text{SPEC}}_k ; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}_{<k}, \mathsf{B}^{\text{SPEC}}_{>k}) = \mathbb{I}(\mathsf{B}^{\text{SPEC}}_k ; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{Z}),$$

by definition of Z.

We need the following intermediate claim for our proof.

Claim 6.2.19.

$$\text{dist}(\mathsf{B}^{\text{SPEC}}_k, \Pi_1, \mathsf{Z} \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}) = \text{dist}(\mathsf{B}^{\text{SPEC}}_k, \Pi_1, \mathsf{Z} \mid \mathsf{T}_{\text{ROW}}, \mathsf{T}_{\text{COL}}, \boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k), \boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)).$$

Proof. Let $\boldsymbol{\rho}_{\text{R}}([b_r \cdot \alpha])$ and $\boldsymbol{\rho}_{\text{C}}([b_r \cdot \alpha])$ denote the random variables $\boldsymbol{\rho}_{\text{R}}(i)$ and $\boldsymbol{\rho}_{\text{C}}(i)$ for all $i \in [b_r \cdot \alpha]$ respectively. We know that random variable $\boldsymbol{\sigma}_{\text{R}}$ is fixed by $\boldsymbol{\rho}_{\text{R}}$, as they are inverses of each other. Random variable $\boldsymbol{\rho}_{\text{R}}$ is in turn fixed by all the values each element of $[b_r]$ is mapped to, i.e., $\boldsymbol{\rho}_{\text{R}}(i)$ for $i \in [b_r]$.

Hence, it is sufficient to argue that,

$$(\mathsf{B}^{\text{SPEC}}_k, \Pi_1, \mathsf{Z}) \perp \boldsymbol{\rho}_{\text{R}}([b_r \cdot \alpha]), \boldsymbol{\rho}_{\text{C}}([b_r \cdot \alpha]) \mid \mathsf{T}_{\text{ROW}}, \mathsf{T}_{\text{COL}}, \boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k), \boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k),$$

to prove the claim. We know that $\mathsf{B}^{\text{SPEC}}_k, \mathsf{Z}$ is the random variable corresponding to the instances in $A^{(r)}$ inside the combinatorial rectangle

$$(\mathsf{T}_{\text{ROW}} \cup \{\boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k)\}) \times (\mathsf{T}_{\text{COL}} \cup \{\boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)\}).$$

The entirety of this rectangle is fixed by the conditioning.

Proving the claim amounts to proving that the joint distribution of $B^{(r-1)}_{i,j}$ for a fixed set of positions of $A^{(r)}$ (the ones inside the rectangle fixed by conditioning), and the transcript Π_1 is independent of the *permutation* of the rest of the positions of $A^{(r)}$. This is true because Q_X, who sends Π_1, does not have access to either of the permutations $\boldsymbol{\rho}_{\text{R}}$ or $\boldsymbol{\rho}_{\text{C}}$ (held by Q_Y). ∎ Claim 6.2.19

We can continue to bound the mutual information now.

$$\begin{aligned}
&\mathbb{I}(\mathsf{B}^{\text{SPEC}}_k ; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{Z}) \\
&= \mathbb{I}(\mathsf{B}^{\text{SPEC}}_k ; \Pi_1 \mid \mathsf{Z}, \mathsf{T}_{\text{ROW}}, \mathsf{T}_{\text{COL}}, \boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k), \boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)) && \text{(by Claim 6.2.19)} \\
&= \mathop{\mathbb{E}}_{\substack{(T_{\text{ROW}}, T_{\text{COL}}) \\ \sim (\mathsf{T}_{\text{ROW}}, \mathsf{T}_{\text{COL}})}} \mathbb{I}(\mathsf{B}^{\text{SPEC}}_k ; \Pi_1 \mid \mathsf{Z}, \mathsf{T}_{\text{ROW}} = T_{\text{ROW}}, \mathsf{T}_{\text{COL}} = T_{\text{COL}}, \boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k), \boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)) \\
& && \text{(by definition of conditional mutual information)} \\
&\underset{(1)}{\leq} \frac{1}{b_r^2 \cdot \alpha^2} \mathop{\mathbb{E}}_{\substack{(T_{\text{ROW}}, T_{\text{COL}}) \\ \sim (\mathsf{T}_{\text{ROW}}, \mathsf{T}_{\text{COL}})}} \sum_{\substack{i \in [b_r] \setminus \mathsf{T}_{\text{ROW}}, \\ j \in [b_r] \setminus \mathsf{T}_{\text{COL}}}} \mathbb{I}(\mathsf{B}_{i,j} ; \Pi_1 \mid \mathsf{Z}, T_{\text{ROW}}, T_{\text{COL}}, \boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k) = i, \boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k) = j)
\end{aligned}$$

$$\underset{(2)}{=} \frac{1}{b_r^2 \cdot \alpha^2} \underset{\substack{(T_{\text{ROW}},T_{\text{COL}}) \\ \sim(\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}})}}{\mathbb{E}} \sum_{i\in[b_r]\setminus\mathsf{T}_{\text{ROW}},j\in[b_r]\setminus\mathsf{T}_{\text{COL}}} \mathbb{I}(\mathsf{B}_{i,j}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}}=T_{\text{ROW}},\mathsf{T}_{\text{COL}}=T_{\text{COL}})$$

$$\underset{(3)}{=} \frac{1}{b_r^2 \cdot \alpha^2} \sum_{i\in[b_r]\setminus\mathsf{T}_{\text{ROW}},j\in[b_r]\setminus\mathsf{T}_{\text{COL}}} \mathbb{I}(\mathsf{B}_{i,j}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}})$$

where (1) holds by the definition of conditional mutual information since, conditioned on any $\mathsf{T}_{\text{ROW}} = T_{\text{ROW}}$ and $\mathsf{T}_{\text{COL}} = T_{\text{COL}}$, the values of $\boldsymbol{\rho}_{\text{R}}(b_r \cdot \alpha + k)$ and $\boldsymbol{\rho}_{\text{C}}(b_r \cdot \alpha + k)$ are uniformly chosen from $[b_r]\setminus T_{\text{ROW}}$ and $[b_r]\setminus T_{\text{COL}}$ (each of size $b_r \cdot \alpha + 1$), respectively; (2) holds because, conditioned on any choice of $\mathsf{T}_{\text{ROW}} = T_{\text{ROW}}$ and $\mathsf{T}_{\text{COL}} = T_{\text{COL}}$, the event $\boldsymbol{\rho}_{\text{C}}(b_r\cdot\alpha+k) = j, \boldsymbol{\rho}_{\text{R}}(b_r\cdot\alpha+k) = i$ is independent of $(\mathsf{B}_{i,j}, \Pi_1, \mathsf{Z})$; and (3) holds by the linearity of expectation and the definition of conditional mutual information.

We re-index both the sets $[b_r]\setminus\mathsf{T}_{\text{ROW}}$ and $[b_r]\setminus\mathsf{T}_{\text{COL}}$ as $[b_r\cdot\alpha+1]$ for ease of exposition just for this final part of the proof. We know that $\mathsf{B}_{i,j}$ for any $i,j\in[b_r]$ are sampled independently of each other in distribution $\mathcal{D}_r$. Hence, using Proposition 4.3.2, we have that

$$\begin{aligned}
&\frac{1}{b_r^2 \cdot \alpha^2} \sum_{i\in[b_r]\setminus\mathsf{T}_{\text{ROW}},j\in[b_r]\setminus\mathsf{T}_{\text{COL}}} \mathbb{I}(\mathsf{B}_{i,j}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}}) \\
&\leq \frac{1}{b_r^2 \cdot \alpha^2} \sum_{i\in[b_r\cdot\alpha+1],j\in[b_r\cdot\alpha+1]} \mathbb{I}(\mathsf{B}_{i,j}\,;\Pi_1 \mid \mathsf{B}_{i,>j},\mathsf{B}_{>i},\mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}}) \\
&= \frac{1}{b_r^2 \cdot \alpha^2} \cdot \mathbb{I}(\mathsf{B}_{1,1},\ldots,\mathsf{B}_{1,b_r\cdot\alpha+1},\mathsf{B}_{2,1},\ldots,\mathsf{B}_{b_r\cdot\alpha+1,b_r\cdot\alpha+1}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}}) \\
&\qquad\qquad \text{(by chain rule of mutual information, Fact 4.3.1-(7))} \\
&\leq \frac{1}{b_r^2 \cdot \alpha^2} \cdot \mathbb{I}(\mathsf{B}^{(r)}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}}) \\
&\text{(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))} \\
&\leq \frac{s}{b_r^2 \cdot \alpha^2}. \qquad \text{(as } \mathbb{I}(\mathsf{B}^{(r)}\,;\Pi_1 \mid \mathsf{Z},\mathsf{T}_{\text{ROW}},\mathsf{T}_{\text{COL}}) \leq \mathbb{H}(\Pi_1) \leq s \text{ by Fact 4.3.1-(1))}
\end{aligned}$$

This proves the claim. ∎

We now use these claims to show that the information revealed by random variable R_a about the input $(\mathsf{A}^\star,\mathsf{B}^\star)$ is not too large in $\mathcal{D}^{\text{REAL}}$.

Claim 6.2.20.

$$\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star},\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}\,;\mathsf{R}_a) \leq \frac{s}{b_r^2\cdot\alpha^2}.$$

Proof. First, by chain rule of mutual information (Fact 4.3.1-(7)), we have that

$$\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star},\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}\,;\mathsf{R}_a) = \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}\,;\mathsf{R}_a) + \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}\,;\mathsf{R}_a \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}).$$

We can prove that the second term is zero as follows: Let R_{comp} represent the independent source of public randomness used in the message compression in step *(ii)*. Then,

$\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}\,;\mathsf{R}_a \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star})$

$$= \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) \qquad \text{(by expanding } \mathsf{R}_a\text{)}$$

$$= \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}) + \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1 \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(by chain rule in Fact 4.3.1-(7))

$$= 0 + \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1 \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}).$$
(by Fact 4.3.1-(2) and construction of $\mathcal{D}_r$)

$$\leq \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{R}_{\text{comp}} \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}).$$
(by data processing inequality Fact 4.3.1-(8) as Π_1 is fixed by $\mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{REST}}, \mathsf{R}_{\text{comp}}$)

$$= \mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{B}^{\text{REST}}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star} \mid \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}).$$
(since R_{comp} is an entirely independent source of randomness)

$$= 0.$$
(by Fact 4.3.1-(2) and construction of $\mathcal{D}_r$)

Therefore, an upper bound on $\mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{R}_a)$ is an upper bound on $\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{R}_a)$. We have,

$$\mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{R}_a)$$

$$\leq \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{R}_a, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$

(by the chain rule (Fact 4.3.1-(7)) and non-negativity of mutual information (Fact 4.3.1-(2)))

$$= \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}) \qquad \text{(by expanding } \mathsf{R}_a\text{)}$$

$$= \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star}) + \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(by chain rule in Fact 4.3.1-(7))

$$= 0 + \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(by Fact 4.3.1-(2) and construction of $\mathcal{D}_r$)

$$\leq \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star, \mathsf{B}^{\text{SPEC}}_{>\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{<\mathsf{k}^\star})$$
(removing $\mathsf{A}^{\text{SPEC}}_{<\mathsf{k}^\star}$ in the conditioning by Proposition 4.3.3 and Claim 6.2.17)

$$= \frac{1}{k_r} \cdot \sum_{k \in [k_r]} \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{k}; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{k}^\star = k, \mathsf{B}^{\text{SPEC}}_{>k}, \mathsf{B}^{\text{SPEC}}_{<k})$$
(by the definition of conditional mutual information)

$$= \frac{1}{k_r} \cdot \sum_{k \in [k_r]} \mathbb{I}(\mathsf{B}^{\text{SPEC}}_{k}; \Pi_1 \mid \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}}, \mathsf{B}^{\text{OFF}}, \mathsf{B}^{\text{SPEC}}_{>k}, \mathsf{B}^{\text{SPEC}}_{<k})$$
(as event $\mathsf{k}^\star = k$ is independent of $(\mathsf{B}^{\text{SPEC}}, \mathsf{B}^{\text{OFF}}, \Pi_1, \boldsymbol{\sigma}_{\text{R}}, \boldsymbol{\sigma}_{\text{C}})$ (see proof of Claim 6.2.13))

$$\leq \frac{s}{b_r^2 \cdot \alpha^2},$$

by applying Claim 6.2.18 to bound each term, completing the proof. ∎

We are now ready to show that $\mathcal{D}^{\text{FAKE}}$ and $\mathcal{D}^{\text{REAL}}$ are statistically close.

Lemma 6.2.21.

$$\|\mathcal{D}^{\text{REAL}} - \mathcal{D}^{\text{FAKE}}\|_{\text{tvd}} \leq \sqrt{\frac{s}{2 \cdot b_r^2 \cdot \alpha^2}}.$$

Proof. Firstly, we use weak chain rule over total variation distance (Fact 4.3.8) using the

definitions of distributions $\mathcal{D}^{\text{FAKE}}$ and $\mathcal{D}^{\text{REAL}}$ established in Claim 6.2.16 to obtain

$$\begin{aligned}
\|\mathcal{D}^{\text{REAL}} - \mathcal{D}^{\text{FAKE}}\|_{\text{tvd}} &\leq \mathop{\mathbb{E}}_{R\sim \mathsf{R}_a} \|\text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a = R) - \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star})\|_{\text{tvd}} \\
&\leq \mathop{\mathbb{E}}_{R\sim \mathsf{R}_a} \sqrt{\frac{1}{2} \cdot \mathbb{D}(\text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a = R) \;||\; \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}))} \\
&\qquad \text{(by Pinsker's inequality in Fact 4.3.10)} \\
&\leq \frac{1}{\sqrt{2}} \cdot \sqrt{\mathop{\mathbb{E}}_{R\sim \mathsf{R}_a} \mathbb{D}(\text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star} \mid \mathsf{R}_a = R) \;||\; \text{dist}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}))} \\
&\qquad \text{(by Jensen's inequality as the function } \sqrt{\cdot} \text{ is concave)} \\
&= \frac{1}{\sqrt{2}} \cdot \sqrt{\mathbb{I}(\mathsf{A}^{\text{SPEC}}_{\mathsf{k}^\star}, \mathsf{B}^{\text{SPEC}}_{\mathsf{k}^\star}; \mathsf{R}_a)} \\
&\qquad \text{(by the relation between mutual information and KL-divergence in Fact 4.3.4)} \\
&\leq \sqrt{\frac{s}{2 \cdot b_r^2 \cdot \alpha^2}}, \qquad \text{(by Claim 6.2.20)}
\end{aligned}$$

proving the lemma. ∎

Finally, we complete the proof of the main round-elimination argument.

Proof of Lemma 6.2.11. By construction, we know that the total communication of protocol π_{r-1} is upper bounded by the communication of protocol $\pi_r^{(2)}$ (since the messages of the former are a subset of the ones for the latter). Using Corollary 6.2.15, we get that the communication of π_{r-1} is at most

$$\text{CC}(\pi_{r-1}) \leq c_{\text{JPY}} \cdot (r/\varepsilon) \cdot \left(\frac{s}{k_r} + r\right).$$

Furthermore, by construction, protocol π_{r-1} is an $(r-1)$-round protocol where Alice communicates first.

We know, again from Corollary 6.2.15, that the probability of success of $\pi_r^{(2)}$ is at least $\delta - \varepsilon$ when the simulated input $(A^{(r)}, B^{(r)})$ is sampled from $\mathcal{D}_r$, i.e., the random variables follow distribution $\mathcal{D}^{\text{REAL}}$. However, in π_{r-1}, due to the fact that we sample the first round message at random, the simulated input is only statistically close to being sampled from $\mathcal{D}_r$, i.e., the random variables follow distribution $\mathcal{D}^{\text{FAKE}}$. Therefore, the probability of success of π_{r-1}, using Fact 4.3.5, is bounded as follows:

$$\begin{aligned}
\text{suc}(\pi_{r-1}) &\geq \Pr\left(\pi_r^{(2)} \text{ succeeds on input } \mathcal{D}^{\text{REAL}}\right) - \|\mathcal{D}^{\text{REAL}} - \mathcal{D}^{\text{FAKE}}\|_{\text{tvd}} \\
&\geq \delta - \varepsilon - \sqrt{\frac{s}{2 \cdot b_r^2 \cdot \alpha^2}}.
\end{aligned}$$

Lastly, to get a deterministic protocol out of π_{r-1}, we use an averaging argument and fix the random bits of π_{r-1} so that we have the same performance guarantee over the input distribution $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$. This gives us the desired $(r-1)$-round deterministic protocol for $\mathsf{AHM}_{r-1}(n_r, \alpha)$ when the players' inputs are sampled from $\mathcal{D}_{r-1}(n_{r-1}, \alpha)$, and the search

sequence is uniformly random. ∎

6.2.3 Reduction to Bipartite Matching in Dynamic Streams

In this section, we prove the connection between maximum bipartite matching in the dynamic streaming model and the Augmented Hidden Matrices (AHM_r) problem, which is presented as Lemma 6.2.6. We repeat this lemma for the convenience of the reader here.

Lemma 6.2.6 (restated). *For any sufficiently large $n_r \in \mathbb{N}$, integers $p, s \geq 1$, and number $\beta \geq 1$, let $\mathcal{A}$ by any p-pass s-space dynamic streaming algorithm that computes a β-approximate maximum matching on a $(4n_r)$-vertex bipartite graphs with probability of success at least $1 - 1/\text{poly}(n)$. Then, for $r = 2p - 1$ and $\alpha = 1/(4 \cdot \beta \cdot r)$, there exists an r-round protocol π for $\mathsf{AHM}_r(n_r, \alpha)$ with probability of success*

$$\text{suc}(\pi) \geq \frac{1}{2} \cdot \left(1 + \frac{1}{3\beta}\right)$$

and communication cost

$$CC(\pi) \leq r \cdot s + O(r \cdot n_r \log(n_r)).$$

The key to proving Lemma 6.2.6 is to construct a bipartite graph $G = (L \cup R, E_{\text{ins}} \setminus E_{\text{del}})$, which is defined by edge insertions E_{ins} and deletions E_{del}, from any input instance for AHM_r with the following property. An $O(1)$-approximate maximum bipartite matching in G can be used to solve AHM_r on the search sequence with a non-trivial advantage over randomly guessing the answer.

6.2.3.1 The Bipartite Graph Construction using AHM_r

Let us first recall the important aspects of AHM_r. It is defined recursively using b_r^2 many AHM_{r-1} sub-instances, which are denoted $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ for $i, j \in [b_r]$ (see Figure 6.4 in Section 6.2.2). An instance of AHM_r identifies k_r many special sub-instances $(A^{\text{SPEC}}, B^{\text{SPEC}})$. Each of these has its own k_{r-1} many special (sub-)sub-instances and, continuing this, ultimately corresponds to the $k_r \cdot k_{r-1} \cdot \ldots \cdot k_1$ many special base instances, where each one corresponds to a single bit as they are instances of AHM_0. Then, given the uniform random search sequence, which is provided at the end of a protocol, solving AHM_r requires solving a uniform random special base instance.

We begin by showing how the base instances (or bits) are represented in our graph. We will use the following basic bipartite graph construction on four vertices, called a **bit graph**, to encode bits in our construction of G.

Definition 6.2.22 (Bit Graph)**.** *Given a bit $x \in \{0, 1\}$ and a graph $G = (L \cup R, E)$ with $L = \{\ell_1, \ell_2\}$ and $R = \{r_1, r_2\}$, graph G is said to be the bit graph of x if*

- *$E = \{(\ell^1, r^1), (\ell^2, r^2)\}$ when $x = 0$, and $E = \{(\ell^1, r^2), (\ell^2, r^1)\}$ when $x = 1$.*

The bit x can be identified by either of the edges in E. See Figure 6.6 for an illustration.



(a) When the bit $x = 0$.

(b) When the bit $x = 1$.

Figure 6.6: An illustration of the bit graph on vertices corresponding to a bit $x \in \{0, 1\}$.

With that, given an instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$, we give the following recursive procedure that separately encodes $A^{(r)}$ and $B^{(r)}$ as the edges of a bipartite graph on the vertex sets L and R, where,

$$L = \{\ell_t^1, \ell_t^2 \mid t \in [n_r]\} \qquad \text{and} \qquad R = \{r_t^1, r_t^2 \mid t \in [n_r]\}.$$

For ease of notation, we let

$$L[i] = \left\{\ell_t^1, \ell_t^2 \mid (i-1) \cdot n_{r-1} < t \leq i \cdot n_{r-1}\right\} \subseteq L$$

for $i \in [b_r]$, and we let $R[j] \subseteq R$ for $j \in [b_r]$ be defined similarly. The edges we add corresponding to the sub-instance $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ will be only across vertices in $L[i]$ and $R[j]$.

We also use $\mathbf{1}(t)$ and $\mathbf{0}(t)$ to denote the $t \times t$ matrix of all ones and all zeroes, respectively. The construction of the graph is as follows.

EDGES(r, Z, L, R) where $Z \in \{A^{(r)}, B^{(r)}, \mathbf{1}(n_r), \mathbf{0}(n_r)\}$ with $|L| = |R| = 2n_r$:

- **For $\mathbf{r = 0}$.**
  - When $Z \in \{0, 1\}$, return the edges of the bit graph corresponding to Z.
  - When $Z = \emptyset$, return the empty set of edges.
- **For $\mathbf{r \geq 1}$.**
  - When $Z \in \{\mathbf{1}(n_r), \mathbf{0}(n_r)\}$, return the edges

    $$\bigcup_{i,j \in [b_r]} \text{EDGES}(r-1, Z', L[i], R[j])$$

    where Z' is the $n_{r-1} \times n_{r-1}$ matrix with all its entries the same as the entries in Z.
  - When $Z = A^{(r)}$, return the edges

    $$\bigcup_{i,j \in [b_r]} \text{EDGES}(r-1, B_{i,j}^{(r-1)}, L[i], R[j]).$$

– When $Z = B^{(r)}$, return the edges[a]

$$\Bigg(\bigcup_{\substack{i,j\in[b_r] \\ b_r\cdot\alpha<\sigma_{\mathrm{R}}(i)\neq\sigma_{\mathrm{C}}(j)\le b_r}} \textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j]) \Bigg)$$
$$\cup \Bigg(\bigcup_{\substack{i,j\in[b_r] \\ b_r\cdot\alpha<\sigma_{\mathrm{R}}(i)=\sigma_{\mathrm{C}}(j)\le b_r}} \textsc{edges}(r-1, A^{(r-1)}_{i,j}, L[i], R[j]) \Bigg).$$

If r is even, in addition to the above, return the edges

$$\bigcup_{\substack{i,j\in[b_r] \\ \sigma_{\mathrm{R}}(i)\le b_r\cdot\alpha \text{ or } \sigma_{\mathrm{C}}(j)\le b_r\cdot\alpha}} \Big(\textsc{edges}(0, \mathbf{1}(n_{r-1}), L[i], R[j]) \cup \textsc{edges}(0, \mathbf{0}(n_{r-1}), L[i], R[j])\Big).$$

See Figure 6.7 for an illustration.

[a] This procedure ignores $A^{\mathrm{OFF}} = (A^{(r-1)}_{i,j} : b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \le b_r)$.

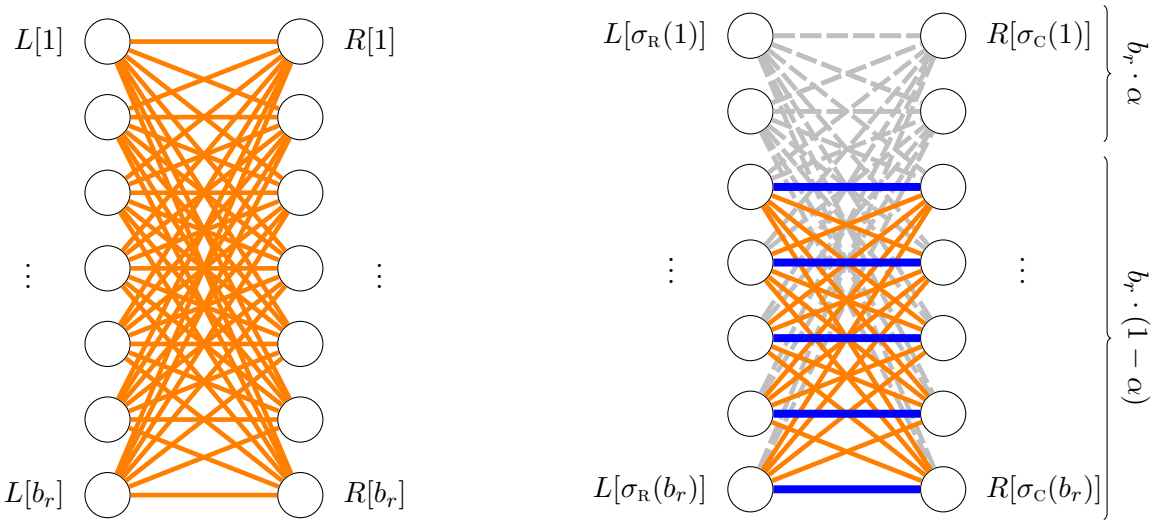
Note that this procedure does not yet define the graph G. However, it allows Alice and Bob to construct edges (insertions or deletions) from their respective inputs, which we now show can be done without any communication.

Claim 6.2.23. *For any $r \ge 0$, given an instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$, Alice and Bob can compute $\textsc{edges}(r, A^{(r)}, L, R)$ and $\textsc{edges}(r, B^{(r)}, L, R)$, respectively, without any communication.*

Proof. The proof of the claim is by induction on r where the base case when $r = 0$ is trivial, i.e., both $\textsc{edges}(0, A^{(0)}, L, R)$ and $\textsc{edges}(0, B^{(0)}, L, R)$ can be constructed without communication.

For any $r \ge 1$, the player holding $A^{(r)}$ has access to all $B^{(r-1)}_{i,j}$ for $i, j \in [b_r]$. Thus, the player can construct the required set of edges $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$ for all $i, j \in [b_r]$ without any communication by the induction hypothesis. The player holding $B^{(r)}$ has access to permutations $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}$ and the following:

- $B^{(r-1)}_{i,j}$ for $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \le b_r$ and thus the player can construct construct the required
$$\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$$
without any communication by the induction hypothesis; and
- $A^{(r-1)}_{i,j}$ when $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) \le b_r$ and thus the player can construct
$$\textsc{edges}(r-1, A^{(r-1)}_{i,j}, L[i], R[j])$$
without any communication, again, by the induction hypothesis.



(a) An illustration of Alice's edges corresponding to $A^{(r)}$. An orange edge between $L[i]$ and $R[j]$ for each $i, j \in [b_r]$ represents the set of edges corresponding to the input $B^{(r-1)}_{i,j}$ held by Alice.

(b) An illustration of Bob's edges corresponding to $B^{(r)}$. The vertex groups are represented after the permutations $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}$ held by Bob. Each orange edge represents the edges corresponding to each off-diagonal sub-instances B^{OFF} held by Bob. Each blue edge represents the edges corresponding to the special sub-instances A^{SPEC} held by Bob. Each dashed gray edge represents the set of no edges when r is odd and the set of all possible edges when r is even.

Figure 6.7: Illustrations of Alice's edges $\textsc{edges}(r, A^{(r)}, L, R)$ (Figure 6.7a) and Bob's edges $\textsc{edges}(r, B^{(r)}, L, R)$ (Figure 6.7b). In both illustrations, the nodes on the left and right corresponds to vertex groups $L[i]$ and $R[j]$ for each $i, j \in [b_r]$, respectively. An edge between vertex groups $L[i]$ and $R[j]$ represents the set of edges corresponding to the instance $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$.

Additionally, when r is even, this player constructs

$$\textsc{edges}(r-1, \mathbf{1}(n_{r-1}), L[i], R[j]) \quad \text{and} \quad \textsc{edges}(r-1, \mathbf{0}(n_{r-1}), L[i], R[j])$$

when $\sigma_{\mathrm{R}}(i) \le b_r \cdot \alpha$ or $\sigma_{\mathrm{C}}(j) \le b_r \cdot \alpha$, which is trivial and does not need any communication since it only requires the knowledge of $\sigma_{\mathrm{R}}, \sigma_{\mathrm{C}}$. ∎

Using the above recursive definition, we obtain the main procedure for constructing a bipartite graph G from any instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$.

One player will be solely responsible for adding edges to G, and the other player only deletes edges from G. The particular roles of Alice and Bob depends on the parity of r.

$\textsc{graph}(A^{(r)}, B^{(r)})$ for instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$:

- When r is **odd**, Alice inserts the edges $E_{\mathrm{ins}} = \textsc{edges}(r, A^{(r)}, L, R)$ and Bob deletes

the edges $E_{\text{del}} = \textsc{edges}(r, B^{(r)}, L, R)$;

- When r is **even**, Bob inserts the edges $E_{\text{ins}} = \textsc{edges}(r, B^{(r)}, L, R)$ and Alice deletes the edges $E_{\text{del}} = \textsc{edges}(r, A^{(r)}, L, R)$.

Return graph $G = (L \cup R, E_{\text{ins}} \backslash E_{\text{del}})$. See Figure 6.8 for an illustration of G.

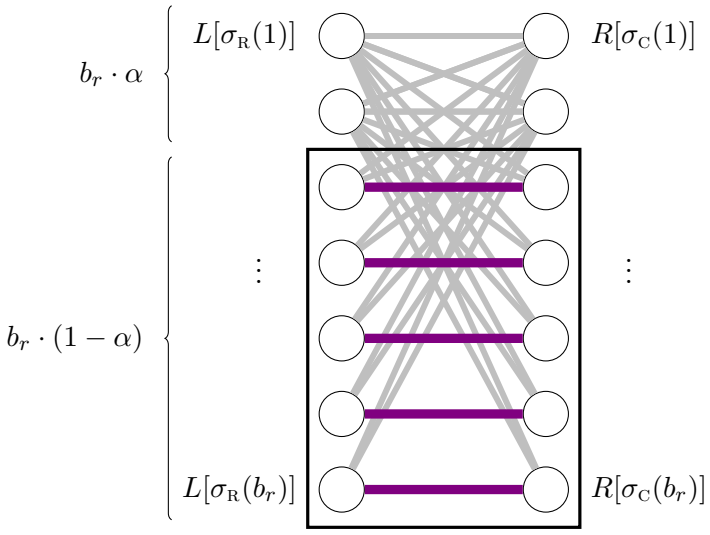


Figure 6.8: An illustration of a graph $G = \textsc{graph}(A^{(r)}, B^{(r)})$ constructed from an instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$. The nodes on the left and right corresponds to vertex groups $L[i]$ and $R[j]$ for each $i, j \in [b_r]$ and are ordered after applying the permutations σ_{R} and σ_{C}, respectively. An edge between vertex groups $L[i]$ and $R[j]$ represents the set of edges corresponding to the instance $(A^{(r-1)}_{i,j}, B^{(r-1)}_{i,j})$, where a lack of edge implies that all edges inserted have also been deleted, i.e., the off-diagonal sub-instances $(A^{\text{OFF}}, B^{\text{OFF}})$. Each purple edge represents the edges that correspond to each special sub-instance in $(A^{\text{SPEC}}, B^{\text{SPEC}})$. The remaining thick gray edges represent the edges corresponding to sub-instances in B^{REST}. The box represents the subgraph of G induced by the vertices $L^{\text{SPEC}} \cup R^{\text{SPEC}}$. See Figure 6.4 for a comparison to AHM_r.

Now, let us show that the instances we create are indeed valid graphs.

Claim 6.2.24. *For any $r \geq 0$, the graph $G = (L \cup R, E_{\text{ins}} \backslash E_{\text{del}}) = \textsc{graph}(A^{(r)}, B^{(r)})$ is a valid graph, namely, $E_{\text{del}} \subseteq E_{\text{ins}}$.*

Proof. By construction, the player that inserts the edges and the player that deletes the edges depends on the parity of r. When r is odd, $E_{\text{ins}} = \textsc{edges}(r, A^{(r)}, L, R)$ and $E_{\text{del}} = \textsc{edges}(r, B^{(r)}, L, R)$, and when r is even, the roles are reversed. Hence, we prove the claim by induction on r where, when r is odd,

$$\textsc{edges}(r, B^{(r)}, L, R) \subset \textsc{edges}(r, A^{(r)}, L, R),$$

, and when r is even, the vice-versa holds.

When $r = 0$ (even case), there are no edge deletions since $\textsc{edges}(0, A^{(0)}, L, R) = \emptyset$, so the claim vacuously holds. When $r \geq 1$, we consider the cases where r is odd and even separately.

Odd r. We consider the following groupings of the edges separately:

- When $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r$, the edge set $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$ held by Bob is directly contained in the edges $\textsc{edges}(r, A^{(r)}, L, R)$ held by Alice.
- When $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) \leq b_r$, Bob holds the edge set $\textsc{edges}(r-1, A^{(r-1)}_{i,j}, L[i], R[j])$ and Alice holds the edge set $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$. Immediately by the induction hypothesis, since $r-1$ is even, we have that $\textsc{edges}(r-1, A^{(r-1)}_{i,j}, L[i], R[j]) \subseteq \textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$.

Even r. Here, we have three different cases:

- When $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) \neq \sigma_{\mathrm{C}}(j) \leq b_r$, the edges $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$ held by Alice are directly contained in the edges $\textsc{edges}(r, B^{(r)}, L, R)$ by Bob.
- When $b_r \cdot \alpha < \sigma_{\mathrm{R}}(i) = \sigma_{\mathrm{C}}(j) \leq b_r$, by the induction hypothesis as $r-1$ is odd, we have that $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j]) \subseteq \textsc{edges}(r-1, A^{(r-1)}_{i,j}, L[i], R[j])$ and are thus contained in $\textsc{edges}(r, B^{(r)}, L, R)$.
- When $\sigma_{\mathrm{R}}(i) \leq b_r \cdot \alpha$ or $\sigma_{\mathrm{C}}(j) \leq b_r \cdot \alpha$, the edges $\textsc{edges}(r-1, B^{(r-1)}_{i,j}, L[i], R[j])$ are trivially a subset of $\textsc{edges}(r-1, \mathbf{1}(n_{r-1}), L[i], R[j]) \cup \textsc{edges}(r-1, \mathbf{0}(n_{r-1}), L[i], R[j])$, which are all included in $\textsc{edges}(r, B^{(r)}, L, R)$ since r is even.

Overall, we have that $E_{\mathrm{del}} \subseteq E_{\mathrm{ins}}$ for any choice of $r \geq 0$ and thus $G = (L \cup R, E_{\mathrm{ins}} \backslash E_{\mathrm{del}}) = \textsc{graph}(A^{(r)}, B^{(r)})$ is a valid graph. ∎

With that, we have that the players can take an instance $(A^{(r)}, B^{(r)})$ of AHM_r and construct a valid graph $G = (L \cup R, E_{\mathrm{ins}} \backslash E_{\mathrm{del}})$ without any communication. In the next section, we will show some structural properties of G that relate it to $(A^{(r)}, B^{(r)})$.

6.2.3.2 Properties of the Constructed Graph

We need some notation before we begin proving the desirable properties of our construction. We define the following sets of special vertices: For $i, j \in [b_r]$,

$$L^{\mathrm{SPEC}}_k := L[i] \text{ when } \sigma_{\mathrm{R}}(i) = b_r \cdot \alpha + k \qquad \text{and} \qquad R^{\mathrm{SPEC}}_k := R[j] \text{ when } \sigma_{\mathrm{C}}(j) = b_r \cdot \alpha + k$$

for all $k \in [k_r]$, and

$$L^{\mathrm{SPEC}} := \bigcup_{k \in [k_r]} L^{\mathrm{SPEC}}_k \qquad \text{and} \qquad R^{\mathrm{SPEC}} := \bigcup_{k \in [k_r]} R^{\mathrm{SPEC}}_k.$$

See Figure 6.8 for an illustration of the special vertices. In the following claim, we show that the special vertices do, in fact, correspond to the special sub-instances $(A^{\text{SPEC}}, B^{\text{SPEC}})$ of $(A^{(r)}, B^{(r)})$.

Claim 6.2.25. *For $r \geq 1$, the subgraph of $G = \text{GRAPH}(A^{(r)}, B^{(r)})$ induced by the vertices $L^{\text{SPEC}} \cup R^{\text{SPEC}}$ is exactly the vertex-disjoint union of graphs $\text{GRAPH}(A_k^{\text{SPEC}}, B_k^{\text{SPEC}})$ for all $k \in [k_r]$.*

Proof. We only prove the claim when r is odd, as the case when r is even follows similarly.

Observe first that the only possible edges between L^{SPEC} and R^{SPEC} are the edges between $L[i]$ and $R[j]$ corresponding to $(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ for any $i, j \in [b_r]$ with $b_r \cdot \alpha < \sigma_{\text{R}}(i), \sigma_{\text{C}}(j) \leq b_r$. Considering only these edges, we split our analysis into two cases:

- When $b_r \cdot \alpha < \sigma_{\text{R}}(i) \neq \sigma_{\text{C}}(j) \leq b_r$, Alice adds the edges corresponding to $B_{i,j}^{(r-1)}$, which is exactly the set of edges deleted by Bob. Hence, none of these edges are present in G.
- When $\sigma_{\text{R}}(i) = \sigma_{\text{C}}(j) = b_r \cdot \alpha + k$ for $k \in [k_r]$, Alice adds the edges corresponding to $B_{i,j}^{(r-1)}$, whereas Bob deletes the edges corresponding to $A_{i,j}^{(r-1)}$. As $r-1$ is even, these edge insertions and deletions exactly correspond to $\text{GRAPH}(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$. It then follows that the subgraph of G induced by $L[i] \cup R[j] = L_k^{\text{SPEC}} \cup R_k^{\text{SPEC}}$ is an exact copy of $\text{GRAPH}(A_k^{\text{SPEC}}, B_k^{\text{SPEC}})$.

Finally, for any $k_1, k_2 \in [k_r]$, $L_{k_1}^{\text{SPEC}} \cup R_{k_1}^{\text{SPEC}}$ is disjoint from $L_{k_2}^{\text{SPEC}} \cup R_{k_2}^{\text{SPEC}}$. Hence, the subgraph induced by $L^{\text{SPEC}} \cup R^{\text{SPEC}}$ is a vertex-disjoint union of the graphs $\text{GRAPH}(A_k^{\text{SPEC}}, B_k^{\text{SPEC}})$ for $k \in [k_r]$, proving the claim. ∎

To highlight another key structural property in our construction of G, recall from Observation 6.2.8 that any instance $(A^{(r)}, B^{(r)})$ of AHM_r defines $k_r \cdot k_{r-1} \cdot \ldots \cdot k_1$ many special base instances. We identify these special base instances and their corresponding sets of special base vertices in L and R using the tuples in $[k_r] \times [k_{r-1}] \times \ldots \times [k_1]$. Formally, we have the following notation:

Notation. Any tuple $s = (s_r, s_{r-1}, \ldots, s_1)$ identifies the special sub-instance $(A_{s_r}^{\text{SPEC}}, B_{s_r}^{\text{SPEC}})$ of $(A^{(r)}, B^{(r)})$, then the special sub-instance $(A_{s_{r-1}}^{\text{SPEC}}, B_{s_{r-1}}^{\text{SPEC}})$ inside $(A_{s_r}^{\text{SPEC}}, B_{s_r}^{\text{SPEC}})$, and so on until finally pointing to a single special base instance $(A_{s_1}^{\text{SPEC}}, B_{s_1}^{\text{SPEC}})$. We denote this special base instance as (A_s^*, B_s^*). Furthermore, we use (A^*, B^*) to denote the collection of special base instances (A_s^*, B_s^*) for all $s = (s_r, s_{r-1}, \ldots, s_1) \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]$.

Any tuple $s = (s_r, s_{r-1}, \ldots, s_1)$ also identifies the sets of vertices $L_{s_r}^{\text{SPEC}} \subseteq L$ and $R_{s_r}^{\text{SPEC}} \subseteq R$, then the sets $L_{s_{r-1}}^{\text{SPEC}} \subseteq L_{s_r}^{\text{SPEC}}$ and $R_{s_{r-1}}^{\text{SPEC}} \subseteq R_{s_r}^{\text{SPEC}}$, and so on until identifying the sets of special base vertices $L_{s_1}^{\text{SPEC}} \subseteq L_{s_2}^{\text{SPEC}}$ in L and $R_{s_1}^{\text{SPEC}} \subseteq R_{s_2}^{\text{SPEC}}$ in R. We denote these two sets of special base vertices as L_s^* and R_s^*. We further define

$$L^* := \bigcup_{s \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]} L_s^* \qquad \text{and} \qquad R^* := \bigcup_{s \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]} R_s^*.$$

With this notation, we can now show that the special base vertices $L^* \cup R^*$ correspond to the special base instances (A^*, B^*) using Claim 6.2.25.

Corollary 6.2.26. *For $r \geq 1$, the subgraph of $G = \textsc{graph}(A^{(r)}, B^{(r)})$ induced by the vertices $L^* \cup R^*$ is exactly the vertex-disjoint union of the bit graphs $\textsc{graph}(A^*_s, B^*_s)$ for all $s \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]$.*

Proof. We prove this by induction on r.

Base case for $\mathbf{r = 1}$. For every $s_1 \in [k_1]$, we have the following directly from the above definitions:

$$L^*_s = L^{\text{SPEC}}_{s_1}, R^*_s = R^{\text{SPEC}}_{s_1}, A^*_s = A^{\text{SPEC}}_{s_1}, B^*_s = B^{\text{SPEC}}_{s_1}.$$

Therefore, $L^* = L^{\text{SPEC}}$ and $R^* = R^{\text{SPEC}}$. Then, by Claim 6.2.25, we have that the subgraph induced by the vertices L^* and R^* is a vertex-disjoint union of graphs

$$\textsc{graph}(A^{\text{SPEC}}_{s_1}, B^{\text{SPEC}}_{s_1}) = \textsc{graph}(A^*_{s_1}, B^*_{s_1})$$

for all $s_1 \in [k_1]$. Furthermore, for each $s_1 \in [k_1]$, $\textsc{graph}(A^*_{s_1}, B^*_{s_1})$ is a bit graph since it corresponds to the special base instance $(A^*_{s_1}, B^*_{s_1})$, which is an instance of AHM_0.

Inductive step for $\mathbf{r \geq 2}$. Let $H^{\text{BASE}} := G[L^* \cup R^*]$ be the subgraph in the statement. It will also be useful for us to consider the subgraph $H^{\text{SPEC}} := G[L^{\text{SPEC}} \cup R^{\text{SPEC}}]$. Since $L^* \subseteq L^{\text{SPEC}}$ and $R^* \subseteq R^{\text{SPEC}}$, we have that,

$$H^{\text{BASE}} = G[L^* \cup R^*] = H^{\text{SPEC}}[L^* \cup R^*].$$

Now, by Claim 6.2.25, H^{SPEC} is a vertex-disjoint union of subgraphs $H^{\text{SPEC}}_{s_r} := \textsc{graph}(A^{\text{SPEC}}_{s_r}, B^{\text{SPEC}}_{s_r})$ for $s_r \in [k_r]$. Therefore, we only need to show that, for each $s_r \in [k_r]$, the subgraph $H^{\text{SPEC}}_{s_r}$, when restricted to the vertices in $L^* \cup R^*$ is exactly a vertex-disjoint union of the bit graphs $\textsc{graph}(A^*_s, B^*_s)$ for all $s \in \{s_r\} \times [k_{r-1}] \times \ldots \times [k_1]$.

Since the only vertices of $L^* \cup R^*$ that are in $H^{\text{SPEC}}_{s_r}$ are exactly its special base vertices, what we want immediately follows from the induction hypothesis applied to $H^{\text{SPEC}}_{s_r}$ for each $s_r \in [k_r]$. ∎

We can also show that these graphs have perfect matchings.

Claim 6.2.27. *For $r \geq 0$ and any $(A^{(r)}, B^{(r)})$, the graph $G = \textsc{graph}(A^{(r)}, B^{(r)})$ contains a perfect matching, i.e., of size $2n_r$.*

Proof. We prove this claim by induction on r where the base case when $r = 0$ is trivial, i.e., we have that $\textsc{graph}(A^{(0)}, B^{(0)})$ is a single bit graph, which has a perfect matching by Definition 6.2.22.

When $r \geq 1$, let $H = G[L^{\text{SPEC}} \cup R^{\text{SPEC}}]$ be its subgraph induced on the special vertices. By Claim 6.2.25 and the induction hypothesis, we immediately have that H contains a perfect matching. Hence, it remains to argue that there is a perfect matching in the subgraph

$$H' = G[L \backslash L^{\text{SPEC}} \cup R \backslash R^{\text{SPEC}}].$$

By definition of the special vertices, $L \backslash L^{\text{SPEC}}$ corresponds to $L[i]$ where $\sigma_{\text{R}}(i) \leq b_r \cdot \alpha$ and, similarly, $R \backslash R^{\text{SPEC}}$ corresponds to $R[j]$ where $\sigma_{\text{C}}(j) \leq b_r \cdot \alpha$ for $i, j \in [b_r]$. Then, to show that there is a perfect matching in H', it is sufficient to argue that

$$G[L[i] \cup R[j]] \text{ for each } i, j \in [b_r] \text{ where } \sigma_{\text{R}}(i) = \sigma_{\text{C}}(j) \leq b_r \cdot \alpha$$

has a perfect matching since these are vertex disjoint graphs that cover all of $L \backslash L^{\text{SPEC}}$ and $R \backslash R^{\text{SPEC}}$. We consider the case when r is odd and even separately.

Odd r. By construction, Alice adds edges corresponding to $B_{i,j}^{(r-1)}$. In order to obtain $\textsc{graph}(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$, which has a perfect matching by the induction hypothesis, Bob would need to delete the edges corresponding to $A_{i,j}^{(r-1)}$ since $r-1$ is even. Instead, Bob deletes no edges so $G[L[i] \cup R[j]]$ is a superset of $\textsc{graph}(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$, which implies that it also has a perfect matching.

Even r. By construction, Alice deletes the edges corresponding to $B_{i,j}^{(r-1)}$. In order to obtain $\textsc{graph}(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$, which has a perfect matching by the induction hypothesis, Bob would need to add the edges corresponding to $A_{i,j}^{(r-1)}$ since $r-1$ is odd. Instead, Bob adds a superset of these edges since he adds all possible edges

$$(\textsc{edges}(r-1, \mathbf{1}(n_{r-1}), L[i], R[j]) \cup \textsc{edges}(r-1, \mathbf{0}(n_{r-1}), L[i], R[j])).$$

Therefore, $\textsc{graph}(A_{i,j}^{(r-1)}, B_{i,j}^{(r-1)})$ is a subgraph of $G[L[i] \cup R[j]]$ and has a perfect matching. ∎

Finally, using these key structural properties, we show that any graph G has a perfect matching and that any β-approximate maximum matching in G includes many edges that correspond to the bit graphs of the special base instances.

Lemma 6.2.28. *For $r \geq 1$, any β-approximate maximum matching M in $G = \textsc{graph}(A^{(r)}, B^{(r)})$ identifies the bits of at least*

$$n_r \cdot (1/\beta - 2\alpha r)$$

many special base instances in $(A^{(r)}, B^{(r)})$.

Proof. By Corollary 6.2.26, we know that the edges in $H = G[L^* \cup R^*]$ are able to identify the bits of the special base instance. Therefore, we first obtain a bound on the number of edges in M that must have both endpoints in $L^* \cup R^*$, i.e., the number of its edges in H.

Observe that the number of edges in M that are not in H have at least one endpoint *not* in $L^* \cup R^*$, i.e., there are at most $|(L \cup R) \backslash (L^* \cup R^*)|$ many such edges. Therefore, M has at least

$$|M| - |(L \cup R) \backslash (L^* \cup R^*)|$$

edges in H. Then, by definition of the special base vertices $L^* \cup R^*$ and the choice of parameters in Eq (6.9) and Observation 6.2.8, we have that

$$|L^*| = |R^*| = 2 \cdot k_r \cdot k_{r-1} \cdot \ldots \cdot k_1 = 2n_r \cdot (1-\alpha)^r$$

and

$$|(L \cup R) \backslash (L^* \cup R^*)| = 4n_r - 4n_r \cdot (1-\alpha)^r = 4n_r(1-(1-\alpha)^r).$$

Now, by Lemma 6.2.27 and since M is a β-approximation, we have that M is of size at least $2n_r/\beta$, which gives us the following bound:

$$\begin{aligned} |M| - |(L \cup R) \backslash (L^* \cup R^*)| &\geq 2n_r/\beta - 4n_r(1-(1-\alpha)^r) \\ &= 4n_r(1/2\beta - 1 + (1-\alpha)^r). \end{aligned}$$

Finally, by Corollary 6.2.26, the only edges in H are the edges of the vertex-disjoint union of the bit graphs $\textsc{graph}(A^*_s, B^*_s)$ for $s \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]$. By Definition 6.2.22, each bit graph is a matching of exactly two edges, each of which can be used to identify the underlying bit of the special base instance. Therefore, in the worst case, every two edges of M in H identifies a single bit of a special base instance and thus at least

$$\begin{aligned} 2n_r(1/2\beta - 1 + (1-\alpha)^r) &\geq 2n_r(1/2\beta - 1 + 1 - \alpha \cdot r) \qquad \text{(by Bernoulli's inequality)} \\ &= n_r \cdot (1/\beta - 2\alpha \cdot r) \end{aligned}$$

many such bits are identified by M. ∎

6.2.3.3 A Protocol for $\mathsf{AHM}_r(n_r, \alpha)$ using a Dynamic Streaming Algorithm

Using the above graph construction and its key properties, we can now construct a protocol π for any instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$ that simulates a run of any p-pass s-space dynamic streaming algorithm $\mathcal{A}$ for β-approximate maximum matching on the bipartite graph $G = \textsc{graph}(A^{(r)}, B^{(r)})$. We then use this protocol to prove Lemma 6.2.6.

A protocol π for $\mathsf{AHM}_r(n_r, \alpha)$ on any input $(A^{(r)}, B^{(r)})$ using the p-pass s-space dynamic streaming algorithm $\mathcal{A}$ where $r = 2p-1$:

(a) Alice computes the edge insertions $E_A = \textsc{edges}(r, A^{(r)}, L, R)$ and then constructs an arbitrary ordering σ_A of the edges E_A. Similarly, Bob computes the edge deletions $E_B = \textsc{edges}(r, B^{(r)}, L, R)$ and then constructs an arbitrary ordering σ_B of the edges E_B.

(b) The players simulate each pass of $\mathcal{A}$ on the stream of insertions σ_A followed by deletions σ_B in the usual manner: To simulate a single pass, Alice runs $\mathcal{A}$ on the insertions in σ_A then sends the memory state to Bob, who continues $\mathcal{A}$ on the deletions in σ_B and sends the memory state back to Alice. In only the final pass, Bob computes the output matching M of $\mathcal{A}$ instead of sending its memory state.

(c) In parallel to the simulation, Alice and Bob also exchange messages to identify the labels of all the special base instances in (A^*, B^*), each of which is identified by a tuple $(s_r, s_{r-1}, \ldots, s_1) \in [k_r] \times [k_{r-1}] \times \ldots \times [k_1]$ as follows.

- In the first round, Alice does not send anything. In the second round, Bob

sends the two input permutations $\sigma_{\text{R}}, \sigma_{\text{C}}$ of $[b_r]$ along with the message to Alice, using which Alice can identify k_r many special sub-instances of AHM_r.

- For any round $2 < t \leq r$, the player who receives the message of round $t-1$ identifies the $k_r \cdot k_{r-1} \cdot \ldots \cdot k_{r-t+3}$ many special instances of AHM_{r-t+2} (sub-instances of AHM_{r-t+3}). In round t, this player sends $2 \cdot k_r \cdot k_{r-1} \cdot \ldots \cdot k_{r-t+3}$ many permutations of $[b_{r-t+2}]$, corresponding to the two input permutations of each of these instances of AHM_{r-t+2}.
- After receiving message of round r, Bob is able to identify the $k_r \cdot k_{r-1} \cdot \ldots \cdot k_2$ many special instances of AHM_1. Bob already knows the two input permutations of $[b_1]$ associated with all these special instances of AHM_1, and thus knows the labels of all the special base instances.

(d) At the end of the protocol, the final player receives a uniform random search sequence $s^* = (k_r^\star, k_{r-1}^\star, \ldots, k_1^\star)$. Then, using the information obtained in the previous step, the player identifies the special base instance $(A^*_{s^*}, B^*_{s^*})$ and corresponding special base vertices $L^*_{s^*} = \{\ell_1, \ell_2\}$ and $R^*_{s^*} = \{r_1, r_2\}$. The player returns x_{s^*} as the output of the protocol, which is determined as follows:

- If M contains the edge (ℓ_1, r_1) or (ℓ_2, r_2), $x_{s^*} = 0$;
- If M contains the edge (ℓ_1, r_2) or (ℓ_2, r_1), $x_{s^*} = 1$;
- Otherwise, x_{s^*} is a uniform random bit.

We now use this constructed protocol to prove the desired connection between maximum bipartite matching in the dynamic streaming model and the AHM_r problem, thus proving Lemma 6.2.6.

Proof of Lemma 6.2.6. To prove this lemma, we show that π is an r-round protocol with

$$\text{CC}(\pi) \leq r \cdot s + O(r^2 \cdot n_r \log n_r) \quad \text{and} \quad \text{suc}(\pi) \geq 1/2 + 1/6\beta$$

where, as in the statement of the lemma, $r = 2p - 1$, $\alpha = 1/(4\beta r)$, and $\beta \geq 1$ is the approximation guarantee of the maximum matching returned by $\mathcal{A}$.

First, we argue the number of rounds of communication required by each step separately:

- By Claim 6.2.23, Alice and Bob can compute their edges E_A and E_B without any communication.
- In the simulation of $\mathcal{A}$, each pass of the algorithm is simulated using two rounds of communication, one message from Alice and one message from Bob, except for the final pass, which only requires one round, i.e., one message from Alice. Since $\mathcal{A}$ has p passes, simulating it requires $r = 2p - 1$ rounds of communication.
- Identifying the labels of the special base instances (which is only required at the end to output the solution) also requires r rounds of communication since, at the end of each

round $t \in [r]$, $k_r \cdot k_{r-1} \cdot \ldots \cdot k_{r-t+1}$ many special instances of AHM_{r-t} are identifiable by the player that receives the message.

- Returning the output of protocol π is solely computed from the output of $\mathcal{A}$ and the identified labels, which can be done without any further communication.

Since the simulation of $\mathcal{A}$ and identifying the labels is done in parallel, protocol π is an r-round protocol.

Next, we argue the communication cost of π by considering separately the only two steps that require communication. To simulate $\mathcal{A}$, the players only exchange the memory state of $\mathcal{A}$. Since this requires at most s bits for each of the r messages, the overall communication required in the simulation is at most $r \cdot s$ bits.

Next, we give an upper bound on the communication required to communicate the labels. In round t for $2 \leq t \leq r$, the player sends $2 \cdot k_r \cdot k_{r-1} \cdot \ldots \cdot k_{r-t+3}$ many permutations of $[b_{r-t+2}]$. Using Eq (6.9), we can say, in total,

$$\begin{aligned} 2 \cdot k_r \cdot k_{r-1} \cdot \ldots \cdot k_{r-t+3} \cdot b_{r-t+2} \cdot \log(b_{r-t+2}) &\leq 2 \cdot b_r \cdot b_{r-1} \cdot \ldots \cdot b_{r-t+3} \cdot b_{r-t+2} \cdot \log(b_{r-t+2}) \\ &= O(n_r \cdot \log(n_r)) \end{aligned}$$

bits are sent. In r rounds, the total number of bits to find the labels of all special instances is $O(r \cdot n_r \cdot \log(n_r))$. Therefore, $\mathrm{CC}(\pi) \leq r \cdot s + O(r \cdot n_r \log n_r)$.

Finally, we argue the probability of success of π. The goal for solving the instance $(A^{(r)}, B^{(r)})$ of $\mathsf{AHM}_r(n_r, \alpha)$ is to output the bit of the special base instance $(A^*_{s^*}, B^*_{s^*})$ where $s^* = (k^\star_r, k^\star_{r-1}, \ldots, k^\star_1)$ is the uniform random search sequence given to the final player, i.e., Bob since $r = 2p - 1$ is odd, at the end of the protocol. In protocol π, the predictor x_{s^*} of the solution is obtained from the matching M returned by $\mathcal{A}$ or is a uniform random guess.

When $\mathcal{A}$ succeeds, which occurs with $1 - 1/\mathrm{poly}(n)$ probability, it is guaranteed to output a β-approximate maximum matching M in the bipartite graph $\textsc{graph}(A^{(r)}, B^{(r)})$. By Lemma 6.2.28, the edges of M identify the bits of at least $n_r \cdot (1/\beta - 2\alpha r)$ many special base instances, which are $n_r \cdot (1-\alpha)^r$ many in total by Observation 6.2.8. Since s^* uniformly selects one of the special base instances to correspond to the solution bit, M identifies it with probability at least

$$\frac{1/\beta - 2\alpha r}{(1-\alpha)^r} \geq 1/\beta - 2\alpha r = 1/\beta - 1/2\beta = 1/2\beta$$

since $\alpha = 1/(4\beta r) \in (0, 1)$.

Thus, we have,

- When $\mathcal{A}$ succeeds and M correctly identifies the solution bit, the predictor x_{s^*} returned by π is correct (with probability 1);
- When $\mathcal{A}$ succeeds but M does not identify the solution bit, the predictor x_{s^*} returned by π is a random guess and thus is correct with probability 1/2;

- When $\mathcal{A}$ fails, there is no guarantee on the predictor bit, and we might as well assume the answer is wrong (with probability 1);

As such, we have that the protocol succeeds with probability at least

$$(1 - 1/\text{poly}(n)) \cdot \left(\frac{1}{2\beta} \cdot 1 + \left(1 - \frac{1}{2\beta}\right) \cdot \frac{1}{2}\right) + 1/\text{poly}(n) \cdot 0 \geq \frac{1}{2} \cdot \left(1 + \frac{1}{3\beta}\right),$$

since β is a constant. This concludes the proof. ∎

Remark 6.2.29. *In the construction of* $\textsc{graph}(A^{(r)}, B^{(r)})$*, when* r *is even, Alice deletes the edges before Bob adds them: for the corresponding streaming problem, this corresponds to deleting edges that have not been inserted (although the final stream still ensures that any edge that is deleted will be inserted); this is* <u>not</u> *consistent with the definition of dynamic graph streams. Nevertheless, in our construction, we* only *use the* AHM_r *lower bound when* $r = 2p - 1$ *and thus is odd. Here, Alice adds edges to the graph, and Bob deletes the edges after all the edges are added. As a result, in the corresponding streaming problem, no edge is deleted before it is inserted, thus adhering to the restriction of the dynamic graph streams.*

7 Distributed Triangle Detection

7.1 Introduction

In the distributed triangle detection problem, we have an n-vertex graph $G = (V, E)$ representing a distributed network with one player for each vertex of the graph who sees the edges incident on the vertex. Players communicate in synchronous rounds by sending $O(\log n)$ bit messages to each of their neighbors per round. The goal is to detect if G contains a triangle as a subgraph in a small number of rounds. Detecting in this problem means that if G contains no triangles, *all* players should output *No*; and, if G contains a triangle, at least *some* player should output *Yes*.

This model of distributed communication is referred to as the CONGEST model [Pel00] to contrast it with the LOCAL model [Lin92] that ignores bandwidth limitations and focuses solely on locality, imposed by communicating only over the edges of the graph. Triangle detection can be trivially solved in a single round of LOCAL by each vertex collecting the neighborhood of its own neighbors. But, this approach incurs a significant communication bottleneck and misrepresents the true complexity of this problem in distributed networks. As a result, triangle detection (among other subgraphs) has been studied extensively in the CONGEST model in recent years, leading to the following state of affairs (see also Section 7.1.1 for more on the related work):

- From the upper bound side, the first algorithm for distributed triangle detection is due to [IG17] and achieves $\widetilde{O}(n^{2/3})$ rounds. This was subsequently improved in [CPZ19] to $\widetilde{O}(n^{1/2})$ rounds using expander decompositions, and further refined in [CS19, CPSZ21] to $\widetilde{O}(n^{1/3})$ rounds. See also [HPZZ21] for an algorithm with guarantees based on maximum degree of the graph. These algorithms are randomized and their guarantees hold with high probability. Subsequent work [CLV22], building on [CS20], obtained deterministic $n^{1/3+o(1)}$ round algorithms for this problem. Plugging in the more recent deterministic expander routing algorithm of [CHS24] in the framework of [CLV22], leads to an $\widetilde{O}(n^{1/3})$ round deterministic algorithm.

- On the lower bound side, not much is known about this problem. Basically, the only lower bounds for this problem are due to [ACKL20] and [FGKO18] who, respectively, showed that deterministic or randomized algorithms cannot solve this problem in a *single* round.

We refer the reader to the excellent survey of [Cen22] for a detailed overview of the literature.

The lack of progress on lower bounds can be attributed to two factors: (1) the main lower bound technique in this model, namely reductions from two-player communication complexity, is provably incapable of establishing any meaningful lower bounds for triangle detection [ACKL20, FGKO18, Cen22] (see also Section 2.3.2); and, (2) there is a provable barrier for establishing strong lower bounds: [EFF+22] shows that a lower bound of n^{δ} rounds

for any constant $\delta > 0$ implies circuit complexity lower bounds that are entirely out of reach of existing techniques (see also Chapter B).

Given the above challenges, proving any lower bounds for distributed triangle detection has been considered a challenging task [ACKL20, FGKO18, Cen22], and prior work has only ruled out one-round algorithms. In this work, we make further progress on this tantalizing open question.

Theorem 5 (restated). *Any algorithm for distributed triangle detection in CONGEST on n-vertex graphs requires $\Omega(\log \log n)$ rounds to succeed with high constant probability.*

Theorem 5 provides a partial answer to the question of determining the round complexity of distributed triangle detection—from the lower bound side—raised repeatedly in the literature, e.g., in [ACKL20, Open Question 1] and [Cen22, Open Problem 2.2], among others.

To put our main result in perspective, we should highlight other lower bounds for distributed triangle detection in related models. The lack of lower bounds in CONGEST motivated [ACKL20] to consider a more limited model of communication wherein each player sends a single bit on each of its edges per round (simulating a round of CONGEST requires $\Omega(\log n)$ rounds in this model). In this model, [ACKL20] proved an $\Omega(\log^* n)$ lower bound for *deterministic* algorithms and [FGKO18] extended the lower bound to $\Omega(\log n)$ rounds. Similarly, [DKO14] proved that any deterministic algorithm that broadcasts the *same* $O(\log n)$-bit message to all its neighbors in each round, needs $n^{1-o(1)}$ rounds[1]. To our knowledge, for randomized algorithms, even in these computationally weaker models, no lower bounds better than a single round were known.

Technical perspective. As mentioned earlier, a key bottleneck in proving lower bounds for distributed triangle detection is the inadequacy of standard applications of communication complexity. This is reminiscent of another (loosely) related area of research: *multi-pass graph streaming lower bounds*, wherein direct applications of communication complexity are often limited (see, [ACK19a] or [Ass23] for a discussion of this topic). However, recent years have witnessed a flurry of breakthroughs in proving lower bounds in this model, e.g., in [GO16, AR20, CKP+21a, AKNS24a] (see [Ass23] for a quick summary) using more direct arguments tailored to the model and problems at hand.

Our work is inspired by these successes and our conceptual contribution is to draw a connection between some of these techniques and the distributed triangle detection problem. At a technical level however, this requires bypassing the inherent difference between the point-to-point communication aspect of CONGEST model versus the broadcast nature of all these lower bounds in graph streaming and related models (e.g. [ANRW15, AKZ22]). Our main technical contribution is thus bridging this gap, which we hope can be of independent interest for proving other distributed lower bounds in scenarios which are not amenable to standard communication complexity arguments.

Our techniques. We shall go over our techniques in detail in the streamlined overview of our approach in Section 7.3. For now, we only mention the high-level bits of our techniques.

[1] The lower bound of [DKO14] also holds in the more general model of *broadcast* CLIQUE wherein each player can send the same message to all vertices of the graph, not only its neighbors.

Our proof is based on *round elimination*, specifically the types introduced in [ANRW15][2], and generalized more recently in [AKZ22] and [AKNS24a, ABK+25] to prove lower bounds in *broadcast* CLIQUE and graph streaming models, respectively[3]. Prior work employ this technique by creating "large" r-round hard instances via combining many $(r-1)$-round hard "small" instances together, in a way that solving the large instance, requires solving one or a few of small instances as well. The construction ensures that the identities of these few small instances are *hidden* from the players in the first round. The analysis then shows that the first-round messages can be "eliminated", leaving the protocol with solving a hard (small) $(r-1)$-instance in $(r-1)$ rounds, which, inductively, is impossible. As we will argue later, direct applications of this approach inherently fail for us given CONGEST allows for a very large communication overall, and the complexity only comes from the limited and "inconsistent" view of each individual player. We thus show how to implement and analyze round elimination at a "per player" level instead of "per (small) instance" level.

It is worth mentioning that graph streaming applications of these techniques in [AKNS24a, ABK+25] rely on complex combinatorial constructions to facilitate the lower bound arguments. In contrast, our hard input distributions are quite elementary from a combinatorial point of view and all the challenge is in the information-theoretic analysis of these distributions.

Barriers for improvement. As stated earlier, the barrier identified in [EFF+22] suggests that, modulo breaking notoriously difficult circuit complexity barriers, the best lower bound we can hope for distributed triangle detection is some $n^{o(1)}$ rounds. This is precisely what our Theorem 5 achieves, although at the "lower end" of the $n^{o(1)}$ range. A natural question now is whether we can improve our lower bound further to "higher ends" of $n^{o(1)}$; we provide some context for this question here.

A slight optimization of the parameters in [EFF+22] gives us that even an $n^{\Omega(1/\log\log n)}$ lower bound for distributed triangle detection breaks circuit complexity barriers; we present this proof in Chapter B for completeness. Moreover, the proof of [EFF+22] relies on expander decompositions and routing schemes in CONGEST model. Expander decompositions now run in polylog(n) rounds [CMGS25] and it is *plausible* that the current algorithms for expander routings can be improved dramatically to also run in polylog(n) rounds. If such a fast algorithms for routing exists, even an $\Omega(\log^c(n))$-round lower bound for some large constant $c > 1$ for triangle detection could lead to breaking circuit complexity barriers (see Remark B.4.3). In light of all these, we believe that realistically, the best lower bound one can hope to prove for our problem is an $\Omega(\log n)$ bound. We leave closing the gap of $\Omega(\log\log n)$ in our Theorem 5 to this "realistic target" of $\Omega(\log n)$ as an interesting open question.

Finally, we remark that our techniques, adapted from [AKNS24a, ABK+25] (and originated from [ANRW15]), are heavily tailored to proving $\Omega(\log\log n)$ lower bounds – in fact, both prior lower bounds of [AKNS24a, ABK+25] are *optimal* for their respective problems,

[2]Their work addresses the welfare maximization problem in mechanism design and bipartite matching in a broadcast communication model between (active) players on one side and (passive) items on the other side.

[3]We note that while conceptually related, this technique is entirely different from the round elimination in LOCAL lower bounds; see, e.g., [Suo20] for more details on the LOCAL round elimination technique.

and as such cannot be extended beyond these bounds. We believe proving a lower bound of $\Omega(\log n)$ for triangle detection in CONGEST likely requires new techniques.

7.1.1 Related Work

Listing vs detection. A more general version of our problem is triangle *listing* where the goal is that every triangle in the network is output by at least one player. All algorithms mentioned earlier for distributed triangle detection [IG17, CPSZ21, CS20, CLV22, CHS24], with the exception of [IG17], also solve the listing problem within the same round complexity. As such, this problem can also be solved in $\widetilde{O}(n^{1/3})$ rounds of CONGEST [CS19, CHS24]. In addition, [IG17, PRS18] proved that this $\widetilde{\Omega}(n^{1/3})$ rounds is nearly-optimal for the listing problem. Their proof is based on a clever but relatively simple information-theoretic argument on random graphs which shows some vertex "learns" $\Omega(n^{4/3})$ bits about the graph (outside its neighborhood) based solely on the answer; since each vertex can only receive $O(n \log n)$ bits per round, this implies an $\Omega(n^{1/3}/\log n)$ round lower bound. This technique is entirely disjoint from our work for triangle detection.

CONGEST vs CLIQUE models. The lower bounds of [IG17, PRS18] hold in the more general (Congested) CLIQUE model [LPPP05] wherein *every* pair of players communicates $O(\log n)$ bits per round. Interestingly, better CLIQUE algorithms are known for triangle *detection*: [CKK+15] designed an $O(n^{1-2/\omega})$ rounds algorithm for triangle detection where ω is the matrix multiplication exponent which currently stands at $\omega \sim 2.371339$ [ADV+25]. Unlike the lower bounds of [IG17, PRS18] for triangle listing, proving *any* super-constant round lower bounds for *any* decision problem in CLIQUE—so, in particular, triangle detection—is a highly challenging task, as it implies circuit lower bounds that are beyond the reach of existing techniques [DKO14].

CONGEST algorithms for triangle listing or detection in [CPSZ21, CS20, CLV22, CHS24] all work by using expander decompositions and routings, to decompose the graph into well-connected components and simulate triangle *listing* algorithms in the CLIQUE model on these components. The circuit complexity barrier of [EFF+22] for proving triangle detection lower bound in CONGEST follows a similar pattern and relies on earlier work of [DKO14] in the CLIQUE model.

Other subgraphs. There is also a large body of work on distributed subgraph detection beyond triangles. One example is the 4-clique listing CONGEST algorithm of [EFF+22] that runs in $n^{3/4+o(1)}$ rounds, which was subsequently improved to $\widetilde{O}(n^{1-2/p})$-round algorithms by [CCGL21, CLV22, CHS24] for all p-cliques for $p \geq 3$. In addition, [CK18] proved a nearly-optimal $\widetilde{\Omega}(n^{1/2})$ lower on the round complexity of detecting 4-cliques in CONGEST. Unlike for triangles, the lower bound for 4-clique can be proven using the standard two-party communication complexity approach. We refer the interested reader to the excellent survey of [Cen22] for a thorough review of the literature on distributed subgraph listing and detection, as well as more details on the uniqueness of *triangles* among all other subgraphs when it comes to proving lower bounds.

7.2 Preliminaries

7.2.1 Notation

At certain places with consecutive lengthy equations, we may color some parts of the text to highlight the differences between nearby equations; these highlights are at no place necessary for parsing the equations and can always be ignored entirely.

Tripartite graphs. We work with tripartite graphs $G = (V, E)$ with vertex set partitioned into $V = A \sqcup B \sqcup C$, and $|A| = |B| = |C| = n$ and no edges with both end points in A or B or C. We use $A = \{a_1, a_2, \ldots, a_n\}$ to denote the elements of set A, and similarly for B, C. For any $i \in [n]$, we use $\{< a_i\}$ to refer to the set $\{a_1, a_2, \ldots, a_{i-1}\}$. This is defined analogously for $\{< b_i\}$ and $\{< c_i\}$.

An important note on the notation: to avoid the repetition and clutter in the notation, we use x, y, z to iterate over vertices of the graph without referring to each particular part of A, B, C. When we use $x \in \{a, b, c\}$, we use X to refer to the set among $\{A, B, C\}$ that x belongs to and vice-versa, and use Y and Z for the other two sets (e.g., if $x = a$, then $X = A$ and $Y \neq Z \in \{B, C\}$).

Another example is to write x_i for $x \in \{a, b, c\}$, to refer to a_i or b_i or c_i, correspondingly. That is, when, say, $x = a$, any other mention of x_i or x_i^* should be interpreted as a_i or a_i^*, respectively.

Vectors. Given a vector L of ℓ elements, we use $L[i]$ for $i \in [\ell]$ to refer to the i^{th} element of L. We use $L[S]$ for any $S \subseteq [\ell]$ to refer to all the elements at positions from S in vector L.

7.2.2 Model of Communication

For technical reasons, we work with a stronger model than CONGEST, wherein vertices are allowed to communicate to some select other vertices in certain rounds even without having an edge to them. Given we are proving a lower bound, this can only strengthen our result. Throughout the paper, we use $G = (V, E)$ to denote the input graph and use the terms 'players' and 'vertices' interchangeably.

Channels. We consider communication as happening over **channels** between pairs of vertices. When we say a channel exists between $u, v \in V$ in some particular round, vertices u and v can send messages to each other in this round. Edge (u, v) may not exist in the input graph, and the channels are a superset of the edges. The sets of channels available may vary between the rounds.

Let r be the total number of rounds in the protocol. We use $\mathcal{C}_j \subseteq V \times V$ to denote the set of channels available at round $(r - j + 1)$ for $j \in [r]$. The channels and the graph satisfy:

$$\mathcal{C}_r \supseteq \mathcal{C}_{r-1} \supseteq \ldots \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_0 := E,$$

where recall that E is the set of edges in the input graph (for notational convenience, it is easier to define the channels in this reverse order, e.g., have $\mathcal{C}_r$ be the channels in round 1 instead of r).

Type of vertex pairs. To any pair of vertices $u, v \in V$, we assign a **type**, denoted by $type(u, v)$, which is an integer in $[r+1] \cup \{0\}$ to determine until when the channel is available for communication:

- If $(u, v) \in E$, then $type(u, v) = 0$;
- If $(u, v) \in \mathcal{C}_j \setminus \mathcal{C}_{j-1}$ for some $j \in [r]$, then $type(u, v) = j$;
- If $(u, v) \notin \mathcal{C}_r$, then $type(u, v) = r + 1$ and (u, v) is not a channel for any round in the protocol.

The **channel-degree** of any vertex u will denote the total number of other vertices v such that $type(u, v) < r + 1$, i.e., the total number of vertices v such that $(u, v) \in \mathcal{C}_r$.

Sources of randomness. Any protocol has three distinct sources of randomness:

- Public randomness: this is a global source of random bits visible to all vertices.
- Pair randomness: for each pair $u, v \in V$, there is a tape of random bits visible only to the two vertices u and v. This is independent of whether (u, v) form any channel in the graph.
- Private randomness: each vertex $u \in V$ has a private tape of random bits only visible to itself.

Input of players. We work with tripartite graphs with a *known* partition of vertices into three parts A, B, C, each of size n (players also know n). The vertices in each layer are identified with elements from $V = A \sqcup B \sqcup C$; we refer to this element as the **identity** of the vertex. Each vertex $x \in X$ is given two vectors of length n, one for each of layers; in the vector for a layer Y, for each $y \in Y$, the type of pair (x, y) which is an integer in $[r+1] \cup \{0\}$ is specified. Note that this uniquely identifies the input graph and all the communication channels.

Communication in a protocol. Communication happens in rounds. In round $i \in [r]$, each vertex $u \in V$ simultaneously sends messages to all $v \in V$ such that $(u, v) \in \mathcal{C}_{r+1-i}$ i.e., in the first round the vertices can send messages over all pairs $(u, v) \in \mathcal{C}_r$, in the second round, over $\mathcal{C}_{r-1}$, and so on (as $\mathcal{C}_0 = E$, the players can always communicate over the edges of G).

Output in a protocol: When no triangle exists in the input graph G, we want *all* vertices to declare that no triangle exists at the end of the last round. When a triangle exists, we want *at least one* vertex to declare that a triangle exists (the vertex may not be part of any triangle).

Protocol parameters. There are three relevant parameters associated with a protocol π:

- $\mathsf{round}(\pi)$: the total number of rounds used by protocol π.

- $\mathsf{bw}(\pi)$: the bandwidth of the protocol, or the maximum of the number of bits sent by any vertex over any channel, taken over all rounds (in CONGEST, we have $\mathsf{bw}(\pi) = O(\log n)$).
- $\mathsf{suc}(\pi)$: the minimum success probability of the protocol π over all input instances where the probability is taken over all the sources of randomness.

For deterministic protocols π, we also have the following parameter for any distribution μ of inputs:

- $\mathsf{suc}(\pi, \mu)$: probability of success of π on average over inputs drawn from the distribution μ.

Our model is **more powerful than the CONGEST** model, as any protocol π in CONGEST can be implemented in our model by only communicating messages over channels $\mathcal{C}_0 = E$ (and ignoring other channels). It is also worth pointing out—even though this will not be related to our paper—that this model *in the limit* can even capture the CLIQUE model by allowing a channel between *all* pairs of vertices throughout all the rounds (i.e., $\mathcal{C}_r = \ldots = \mathcal{C}_1 = \binom{V}{2}$ and $\mathcal{C}_0 = E$ as before). Although for our lower bounds, we certainly do not allow all channels to exist.

7.3 Technical Overview

Our proof of Theorem 5 is quite technical and involves various information-theoretic maneuvers that can be unintuitive or daunting to parse. Thus, we use this section to unpack our main ideas and give a streamlined overview of our approach. We emphasize that this section oversimplifies many details and the discussions will be informal for the sake of intuition. The rest of the paper is written in an independent way so the reader can skip this part and directly jump to technical arguments.

We start with a recap of the round elimination arguments that we have seen so far, and others that have a similar flavor.

7.3.1 Background: Round Elimination Arguments in Prior Work

We now switch to discussing the type of round elimination ideas that have proven quite successful in establishing distributed lower bounds in certain broadcast models [ANRW15, AKZ22] and more recently graph streaming lower bounds [AKNS24a, ABK+25][4]. While there are substantial differences between applications of these ideas across these works, at a sufficiently high level (and for the purpose of comparison with our own work), we can characterize all of them as follows.

We inductively create a family of distributions $\{\mathcal{G}_r\}_{r \geq 0}$ where $\mathcal{G}_r$ generates hard inputs for r-round algorithms (with the base case $\mathcal{G}_0$ being simply some non-trivial distribution as 0-round algorithms are typically very easy to analyze directly). For any $r \geq 1$, the distribution $\mathcal{G}_r$ is constructed by sampling a "large" number of $(r-1)$-round independent

[4]We shall note that origins of these ideas also directly traces back to the two-party communication complexity model and the "round elimination lemma" of [MNSW95] in that model.

instances $I_1, I_2, \ldots, I_{k_r}$ from $\mathcal{G}_{r-1}$. These instances are then "packed" together carefully to ensure that for some randomly chosen $j^\star \in [k_r]$, (1) the answer to the instance I sampled from $\mathcal{G}_r$ is determined by the answer of the inner instance $I_{j^\star}$ from $\mathcal{G}_{r-1}$, and yet, (2) in the first round of the protocol, the players are oblivious to the value of $j^\star$. We note that this way, often the inputs and number of players in r-round instances in $\mathcal{G}_r$ are polynomially larger than those in $(r-1)$-round instances.

Suppose we have an r-round protocol π_r for solving instances $I \sim \mathcal{G}_r$ with success probability $\mathsf{suc}(\pi_r)$. We construct an $(r-1)$-round protocol π_{r-1} for solving instances $I' \sim \mathcal{G}_{r-1}$ using π_r:

1. The players of π_{r-1} use public randomness to sample an index $j^\star \in [k_r]$ as well as the first round of messages $M^{(1)}$ of π_r from its distributions. They assume the inner $(r-1)$-round instance $I_{j^\star}$ of the outer r-round instance I is their input I'.
2. They then use a combination of private and public randomness to sample the remainder of I conditioned on $M^{(1)}$ and $I_{j^\star} = I'$ to create a complete r-round instance[a].
3. Finally, the players in π_{r-1} already have access to the first-round message $M^{(1)}$ and thus only need to run π_r from its second round onwards—while simulating remaining parts of I outside $I_{j^\star} = I'$—and output the same answer as π_r.

[a] This step is highly non-trivial and specialized as each player of I' only knows its own input in I' and not the entire I' and so it is not clear (nor always possible) for the players to sample I conditioned on $M^{(1)}$ and $I_{j^\star} = I'$.

To argue the correctness of the protocol π_{r-1}, we only need to show that the distribution of inputs and messages to π_r induced by π_{r-1} is within $o(1)$ total variation distance of the correct distribution; the rest follows by construction, since the answer to $I \sim \mathcal{G}_r$ is the same as $I_{j^\star}$ which is chosen to be I', namely, the instance π_{r-1} wants to solve in the first place. Specifically,

- The *right* distribution of all random variables in π_r, for a fixed $j^\star$, can be expressed as:

$$\mathsf{M}^{(1)} \times \left(\mathsf{I}_{j^\star} \mid \mathsf{M}^{(1)}\right) \times \left(\mathsf{I}_{-j^\star} \mid \mathsf{M}^{(1)}, \mathsf{I}_{j^\star}\right),$$

  where $\mathsf{I}_{-j^\star}$ denotes the random variable for all $(r-1)$-round sub-instances except for $I_{j^\star}$.

- The distribution sampled from in the protocol π_{r-1} on the other hand is:

$$\underbrace{\mathsf{M}^{(1)}}_{\text{publicly}} \times \underbrace{\mathsf{I}_{j^\star}}_{\text{input}} \times \underbrace{\left(\mathsf{I}_{-j^\star} \mid \mathsf{M}^{(1)}, \mathsf{I}_{j^\star}\right)}_{\text{mix of publicly and privately}},$$

  namely, here, $M^{(1)}$ and $I_{j^\star}$ are sampled independently of each other.

Assuming the third step of sampling can be done—which, to emphasize again, contains the

bulk of efforts in many of these works —the distance between the two distributions is only a function of their second arguments. This distance, for a random choice of $j^\star \in [k_r]$, can be upper bounded as

$$\mathop{\mathbb{E}}_{j^\star \in [k_r]} \|\mathsf{I}_{j^\star} - (\mathsf{I}_{j^\star} \mid \mathsf{M}^{(1)})\|_{\mathrm{tvd}}^2 \leq \mathop{\mathbb{E}}_{j^\star \in [k_r]} \mathbb{I}(\mathsf{I}_{j^\star}; \mathsf{M}^{(1)}) \leq \frac{1}{k_r} \cdot \mathbb{I}(\mathsf{I}_1, \ldots, \mathsf{I}_{k_r}; \mathsf{M}^{(1)}) = o(1);$$

here, the first inequality is standard (see Fact 4.3.10 and Fact 4.3.4) and the main inequality is the second one: it holds roughly because the players in π_r (but of course not π_{r-1}) are *oblivious* to the choice of $j^\star$ in their first round and thus the information revealed by their first-round messages is "spread" over all k_r sub-instances. The final equality also holds by ensuring that k_r is much larger than the length of sampled messages $M^{(1)}$ (and applying Fact 4.3.1-(1)).

All in all, this means protocol π_{r-1} has success probability $\mathsf{suc}(\pi_{r-1}, \mathcal{G}_{r-1}) \geq \mathsf{suc}(\pi_r, \mathcal{G}_r) - o(1)$ and a similar communication bandwidth as π_r, but with one fewer round. Continuing like this then leaves us with a 0-round protocol for $\mathcal{G}_0$ with a non-trivial probability of success, a contradiction.

Before moving on, we should caution the reader that while the above discussion captures the common theme of arguments across the aforementioned work, none of those work follow this recipe directly. For instance, [ANRW15] samples the message $M^{(1)}$ to be only the ones originating from players in $I_{j^\star}$ (to ensure its size is small with respect to k_r); [AKZ22] does not even sample the remainder of I and follows the simulation through sampling relevant messages from $I_{-j^\star}$ to $I_{j^\star}$; and finally, for [AKNS24a, ABK+25], the number of players is a lot fewer (in fact, just two players in [ABK+25]) and instead the players are forced to solve many albeit a small fraction of inner sub-instances at the same time. This discussion also only focused on the information-theoretic aspects of the lower bounds and entirely neglected the combinatorial parts of how one can "pack" so many $(r-1)$-round independent instances inside a single r-round instance. However, we hope our description provides a big picture of these prior arguments.

7.3.2 Our Approach

We now start presenting the main ideas behind our own work.

7.3.2.1 A Hard Input Distribution

We generate a recursive family of hard input distributions $\{\mathcal{G}_r\}_{r \geq 0}$ where $\mathcal{G}_r$ samples tripartite graphs with n_r vertices in each layer for r-round protocols. For reasons that will become clear soon, unlike in Section 7.3.1, we cannot specify our r-round instances as a combination of many $(r-1)$-round instances. Instead, they are obtained by sampling a single $(r-1)$-round instance plus a very large individualized "noise" for each vertex in the inner $(r-1)$-round instance to hide its inner edges among the outer graph.

We use parameters n_r and d_r in the description of the distribution and they are defined

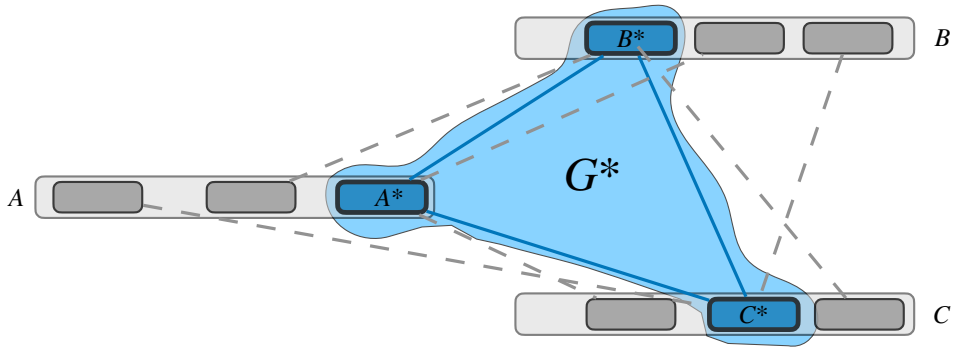


Figure 7.1: An illustration of our hard input distribution. The solid (blue) edges show the $(r-1)$-round hard instance inside an r-round hard instance and the dashed (gray) edges show the extra unique channels sampled for each vertex of the inner $(r-1)$-round instance.

recursively as follows[5]:

$$d_r = \text{poly}(n_{r-1}) \qquad \text{and} \qquad n_r = \text{poly}(d_r). \tag{7.1}$$

The base case $\mathcal{G}_0$ has $n_0 = 1$ and simply consists of three vertices a, b, c with an edge between each pair appearing with probability $1/2$ (and thus a triangle with probability $1/8$). The distribution for r-rounds for $r > 0$ is as follows.

Distribution 1. A hard distribution $\mathcal{G}_r$ of graphs $G = (A \sqcup B \sqcup C, E)$ for r-round protocols:

1. Sample three sets of vertices $A^* \subseteq A, B^* \subseteq B$, and $C^* \subseteq C$ each of size n_{r-1} independently.
2. Sample an $(r-1)$-round instance G^* over (A^*, B^*, C^*) and let the induced subgraph of G on these vertices be G^*. The vertices in different layers have a channel in $\mathcal{C}_r$ to each other, whose type is determined by the channel-type in G^* (not having a channel in G^* translates to a type r in G).
3. For any x in some layer X^*, any layer $Y \neq X \in \{A, B, C\}$, and any type $t \in [r] \cup \{0\}$ of channel, sample enough vertices for v uniformly from $Y \setminus Y^*$ such that x has d_r type-t channels in total to Y. We require all sampled vertices to be unique across all vertices of G^*.

See Figure 7.1 for an illustration.

It is easy to see that a graph G sampled from $\mathcal{G}_r$ for $r \geq 1$ has a triangle iff its inner graph G^* has one. Thus, the task of players in solving an r-round instance G reduces to that of solving the inner $(r-1)$-round instance G^*. Moreover, while a player in G can determine whether it belongs to G^* or not (its channel-degree is only 1 in the latter case), it cannot

[5]We specify these parameters explicitly in the actual proof, but here keep them at this level to focus on their connections as opposed to their actual values.

determine this information about its neighbors on its own. Intuitively, this suggests that the messages between two players $x, y \in G^*$ cannot reveal almost any information about G^* as either of these vertices have $> d_r$ channels, but only $n_{r-1} \ll d_r$ of them are in G^*, and their identities are unknown to the sender. Hence, one expects the first round messages to be "wasted" until the players find their channels in G^*, and then have to solve a hard $(r-1)$-round instance in the remaining $(r-1)$ rounds.

As expected, turning this intuition into an actual proof is quite challenging. For instance, our distribution $\mathcal{G}_1$ for one-round protocols is almost identical to the ones for the one-round lower bound of [FGKO18] (modulo the notion of channels) and based on the same principle as the deterministic one-round lower bound of [ACKL20]. But, even for one-round protocols, quite a lot of technical work has been done in [FGKO18, ACKL20] to establish the lower bound. As such, all our effort in this paper is dedicated to formalizing this basic intuition. It is also worth pointing out that while many one-round lower bounds can be cast as round elimination arguments to zero-round instances, this is *not* the case for the arguments in [FGKO18, ACKL20], and thus from a technical point of view, our arguments already deviate from these prior approaches even for one-round protocols.

7.3.2.2 First Attempt on Round Elimination: Public Sampling of Messages

Following prior work in Section 7.3.1, let us consider creating an $(r-1)$-round protocol π_{r-1} for $G_{r-1} \sim \mathcal{G}_{r-1}$ from an r-round protocol π_r for $G_r \sim \mathcal{G}_r$:

1. The players of π_{r-1} use public randomness to sample vertices (A^*, B^*, C^*) and map $[n_{r-1}]$, namely, vertices of their input G_{r-1} in each layer, to these sets accordingly. They then define the induced subgraph G^* in G_r to be G_{r-1} after this mapping of vertices.

2. The players of π_{r-1} sample the first-round messages $M^{(1)}$ of π_r using public randomness conditioned on (A^*, B^*, C^*) but independent of the types of channels in G^*.
   ...

We will already run into a serious problem!

For this approach to have any chance of succeeding, we need to be able to say that the joint distribution of $(M^{(1)}, G^*) \mid (A^*, B^*, C^*)$ in the protocol π_{r-1} is close to being a product distribution, so that sampling them independently in π_{r-1} does not change their distribution too much. But certainly distribution of $M^{(1)}$ highly correlates with that of G^*: for any channel in G^*, the message of $M^{(1)}$ can easily reveal the type of this channel with very small communication (and if we consider the original CONGEST model, then simply having a message between a pair of vertices reveals the existence of the edge between them as well!). Thus, the protocol above is doomed to fail.

This issue stems from the inherent difference of point-to-point communication versus broadcast ones targeted in Section 7.3.1. Basically, sampling the message $M^{(1)}$ publicly is effectively the same as revealing the first round messages to everyone (in the second round);

but unlike a broadcast model, for us, this is way too much information. Fortunately, to simulate a CONGEST protocol π_r, each vertex only needs to know the messages communicated to and from it and not all messages. Hence, the **first lesson** for our round elimination protocol is that sampling first-round messages should "respect" the point-to-point communication pattern as well.

7.3.2.3 Second Attempt on Round Elimination: Pair Sampling of Messages

Equipped with our previous lesson, we make a second attempt at creating the protocol π_{r-1}. In addition to public randomness, we have equipped the vertices with pair randomness, to facilitate the sampling of first round messages. See Section 7.2 for the different sources of randomness given to the vertices.

1. The players of π_{r-1} use public randomness to sample vertices (A^*, B^*, C^*) and define G^* of G_r based on their input G_{r-1}, as before.
2. Then, every pair of vertices in $x, y \in G^*$, use pair randomness to sample the messages to and from each other, namely, $M^{(1)}_{x\to y}$ and $M^{(1)}_{y\to x}$, conditioned on (A^*, B^*, C^*) and $type(x, y)$ but independent of the rest of G^*.
   ...

Again, we stop the description of the protocol here already.

Let us examine the two distributions involved here:

- The right distribution of involved variables is:

$$(\mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*) \times (\mathsf{G}^* \mid \mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*) \times \bigtimes_{(x,y)\in \mathsf{X}^*\neq \mathsf{Y}^*} \left(\mathsf{M}^{(1)}_{x,y} \mid \mathsf{M}^{(1)}_{<(x,y)}, \mathsf{G}^*, \mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*\right).$$

  where $\mathsf{M}^{(1)}_{x,y} = (\mathsf{M}^{(1)}_{x\to y}, \mathsf{M}^{(1)}_{y\to x})$ denotes both messages communicated over a pair (x, y), and, under some arbitrary ordering between all pairs of vertices $(x, y) \in G^*$, variable $\mathsf{M}^{(1)}_{<(x,y)}$ collects these messages for all pairs before (x, y).

- On the other hand, the distribution of variables in π_{r-1} are:

$$\underbrace{(\mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*)}_{\text{public randomness}} \times \underbrace{(\mathsf{G}^* \mid \mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*)}_{\text{input plus renaming}} \times \bigtimes_{(x,y)\in \mathsf{X}^*\neq \mathsf{Y}^*} \underbrace{\left(\mathsf{M}^{(1)}_{x,y} \mid \mathsf{G}^*_{x,y}, \mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*\right)}_{\text{pair randomness}},$$

  where $\mathsf{G}^*_{x,y}$ is the random variable for $type(x, y)$ (conditioned on vertices (A^*, B^*, C^*)).

Bounding the difference between these distributions thus boils down to bounding the following for every pair $(x, y) \in G^*$ (we use $G^*_{-(x,y)}$ to denote the other types in G apart from (x, y) and we say $\mathsf{N}^* = (\mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*)$ to avoid the clutter):

$$\| \left(\mathsf{M}^{(1)}_{x,y} \mid \mathsf{M}^{(1)}_{<(x,y)}, \mathsf{G}^*, \mathsf{N}^*\right) - \left(\mathsf{M}^{(1)}_{x,y} \mid \mathsf{G}^*_{x,y}, \mathsf{N}^*\right) \|^2_{\text{tvd}} \le \mathbb{I}(\mathsf{M}^{(1)}_{x,y}; \mathsf{G}^*_{-(x,y)}, \mathsf{M}^{(1)}_{<(x,y)} \mid \mathsf{G}^*_{x,y}, \mathsf{N}^*).$$

(by Fact 4.3.10 and Fact 4.3.4)

Using the chain rule of mutual information (Fact 4.3.1-(7)), we can further write the RHS above as

$$\mathbb{I}(\mathsf{M}^{(1)}_{x,y}\,;\mathsf{G}^*_{-(x,y)},\mathsf{M}^{(1)}_{<(x,y)}\mid\mathsf{G}^*_{x,y},\mathsf{N}^*)=\mathbb{I}(\mathsf{M}^{(1)}_{x,y}\,;\mathsf{G}^*_{-(x,y)}\mid\mathsf{G}^*_{x,y},\mathsf{N}^*)+\mathbb{I}(\mathsf{M}^{(1)}_{x,y}\,;\mathsf{M}^{(1)}_{<(x,y)}\mid\mathsf{G}^*,\mathsf{N}^*). \tag{7.2}$$

Let us now consider each term in the RHS separately.

First term in RHS of Eq (7.2). This term measures the information revealed by the messages between x, y about *other* pairs inside G^* (conditioned on $type(x, y)$). Our intuition is that this information should be quite low because vertices x and y cannot distinguish between their channels in or out of G^*, and thus should not be able to tell each other about the type of their other edges inside the graph G^*.

At this point, it is tempting to follow a similar strategy as in Section 7.3.1 and re-write the distribution $\mathcal{G}_r$ so that it is a combination of, say, k_r many $(r-1)$-round instances. We can then say that given x participates in k_r many instances and only one of them is special (namely, is G^*), x will not be able to reveal much information about this instance in its message. The problem is again the unicast versus broadcast: if the partitioning of the input into separate instances are known to the players, even if they do not know which one is special, we still have a problem: vertex x can reveal a lot of information to its neighbor y about the *same* instance they are both part of.

To bypass this, we instead use an *inconsistent* way of creating these instances. Suppose in the distribution $\mathcal{G}_r$, for the vertex $x \in G^*$ we sample k_r *partial* $(r-1)$-round sub-instances $K_1, \ldots, K_{k_r}$ on $2n_{r-1} - 1$ vertices such that each one is sampled from a distribution that combined with the edge (x, y) can form the input of x in a $\mathcal{G}_{r-1}$ instance. See Figure 7.2 for an illustration.

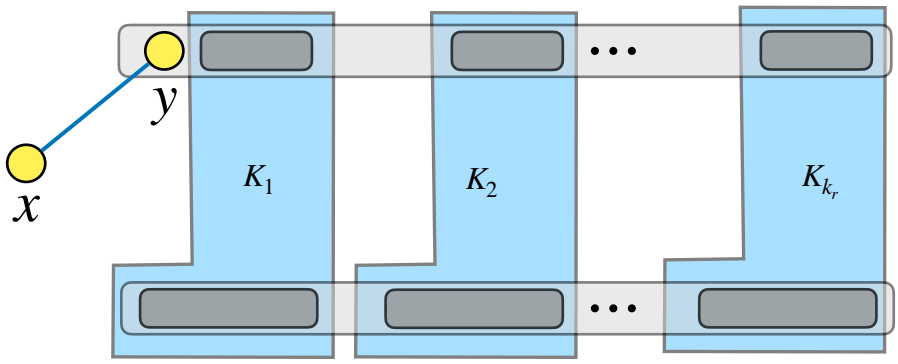


Figure 7.2: An illustration of sampling partial $(r-1)$-round sub-instances. Here, from the perspective of x, the edge (x, y) plus any of K_i's for $i \in [k_r]$ form a proper input of x in an instance sampled from $\mathcal{G}_{r-1}$. One of these instances corresponds to the special inner instance G^*, while for all other instances, the instance is not complete, as in there are no edges between Y- and Z-part of the instance.

By setting $k_r = d_r/(n_{r-1})^2$, we can show that for inputs sampled from $\mathcal{G}_r$, with high

probability, we can find such sub-instances for every vertex x and any of the r possible types of channels in G^* (basically, x has sampled d_r many channels of each type so one can re-order them to create k_r many partial sub-instances). Moreover, from the perspective of x in protocol π_r, the special part of the input, namely, $G^*_{-(x,y)}$ can be any of these k_r many partial sub-instances. This, with some more technical but standard ideas that we omit here, allows us to essentially write:

$$\mathbb{I}(\mathsf{M}^{(1)}_{x,y}\,;\mathsf{G}^*_{-(x,y)} \mid \mathsf{G}^*_{x,y},\mathsf{N}^*) \leq \mathbb{I}(\mathsf{M}^{(1)}_{x,y}\,;\mathsf{G}^*_{-(x,y)} \mid \mathsf{k}_1,\ldots,\mathsf{k}_{k_r},\mathsf{G}^*_{x,y},\mathsf{N}^*) \leq \frac{1}{k_r}\cdot \mathsf{bw}(\pi_r),$$

by effectively arguing that the information revealed by $\mathsf{M}^{(1)}_{x\to y}$ is spread over k_r many possible choices for $\mathsf{G}^*_{-(x,y)}$ among $K_1,\ldots,K_{k_r}$ from the perspective of x (and similarly for $\mathsf{M}^{(1)}_{y\to x}$ and y). Here, with an abuse of notation, conditioning on $\mathsf{k}_1,\ldots,\mathsf{k}_{k_r}$ means conditioning on the existence of $\mathcal{C}_r$ channels to them, but the type of those channels are not fully determined (only that they are sampled from the marginal distribution of $\mathcal{G}_{r-1} \mid \mathit{type}(x,y)$).

Given the choice of parameters in Eq (7.1), across all choices of $x,y \in G^*$, the total contribution of the first term of RHS of Eq (7.2) to the distances between distributions is

$$(n_{r-1})^2 \cdot \frac{1}{k_r}\cdot \mathsf{bw}(\pi_r) = \frac{(n_{r-1})^4}{d_r}\cdot \mathsf{bw}(\pi_r) \leq \frac{1}{\mathrm{poly}(n_{r-1})}\cdot \mathsf{bw}(\pi_r) \leq 1/\mathrm{poly}(n_{r-1}).$$

Second term in RHS of Eq (7.2). This term measures the information revealed by the messages between x,y about some of *other messages*, conditioned on the entire special sub-instance G^*. This is where we run into another serious problem.

Consider a protocol wherein every vertex x samples a random coin privately and sends its value as part of the message to all its neighbors. In such a protocol, knowing the message $\mathsf{M}^{(1)}_{x,y}$ for any y reveals (at least) one bit of information about $\mathsf{M}^{(1)}_{x,z}$ for any other $z \neq y$ as well, making the RHS of Eq (7.2) at least 1 which is way too large for our purpose.

Thus, our second attempt at designing protocol π_{r-1} also fails, although this time we made some further progress. The **second lesson** is that we need to break the correlation between messages originating from a single vertex to other vertices in G^* (but certainly not all of G).

7.3.2.4 Third Attempt on Round Elimination: Public and Pair Sampling of Messages

We now use yet another sampling process for obtaining π_{r-1} from π_r. To simplify the exposition, we only write this sampling from the perspective of a single pair $(x,y) \in G^*$ and only the message $M^{(1)}_{x\to y}$ from x to y, *assuming* other messages $M^{(1)}_{x\to -y}$ of x to $G^* \setminus \{y\}$ have already been (somehow) sampled using their own pair randomness. Doing this for all pairs of vertices has several additional challenges but we skip those in this discussion to provide the high level intuition (see also Figure 7.3).

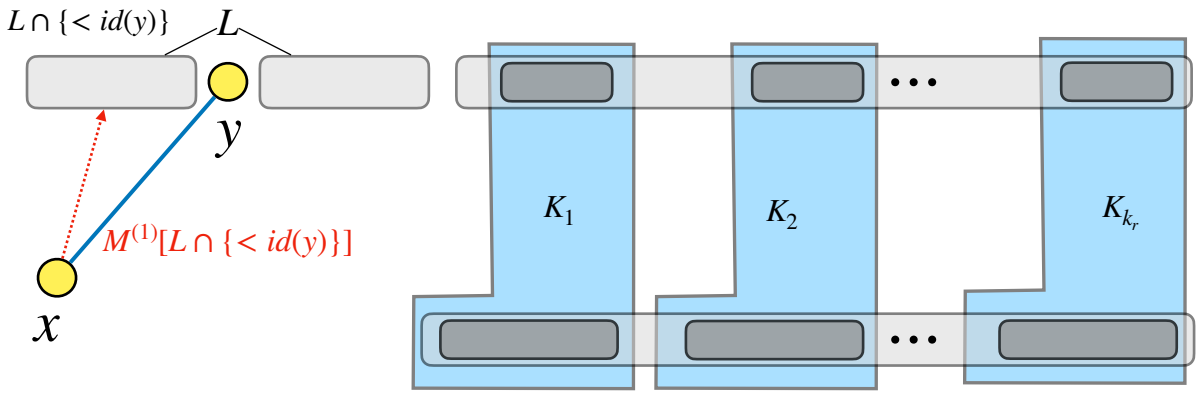


Figure 7.3: An illustration of sampling messages via a mixture of public and pair randomness. The type of the channel between (x, y) any (x, y') for any $y' \in L$ is the same.

1. The players of π_{r-1} use public randomness to sample vertices (A^*, B^*, C^*) and define G^* of G_r based on their input G_{r-1}, as before. We also assume messages $M^{(1)}_{x\to -y}$ have been sampled using pair randomness at this point (so, in particular, are unknown to y).

2. Player x uses public randomness to sample

$$K_1, \ldots, K_{k_r} \subseteq Y \cup Z, \qquad L \subseteq Y, \qquad \text{and} \qquad M^{(1)}_x[L \cap \{< id(y)\}];$$

   - K_i is a disjoint set of $(2n_{r-1} - 1)$ vertices in G, which together with (x, y) can form an $(r-1)$-round instance from $\mathcal{G}_{r-1}$; we let x have $\mathcal{C}_r$-channels to all these vertices but do not sample their types yet (also, value of k_r will be determined later);
   - L is a disjoint set of ℓ_r (to be determined later) random vertices in Y, each of which has a $type(x, y)$-channel to x;
   - $M^{(1)}_x[L \cap \{< id(y)\}]$ is the messages x sends to its neighbors in $L \cap \{< id(y)\}$ in π_r conditioned on all the *publicly* sampled information ($id(y) \in Y^*$ is the id assigned to $y \in G_{r-1}$).

3. Finally, x and y sample $M^{(1)}_{x\to y}$ via pair randomness conditioned on $type(x, y)$ and all the *publicly* sampled variables (so specifically, independent of $M^{(1)}_{x\to -y}$).

As for the parameters ℓ_r and k_r, we pick them such that

$$\ell_r = \text{poly}(n_{r-1}), \qquad k_r = \text{poly}(\ell_r), \qquad \text{and} \qquad d_r = \text{poly}(k_r). \tag{7.3}$$

We note that the entire goal of these sampling steps is to generate an input G_r and first-round messages $M^{(1)}$ for the protocol π_r so that players in π_{r-1} can run this protocol on the sampled input. The above process only specifies some part of the input to one of the players x, with respect to one of the pairs (x, y) in the original input of π_{r-1}. We will need

to repeat this process for all pairs of vertices in π_{r-1} and complete the remainder of inputs and messages of players (including x) before we can run π_r. However, we can already start analyzing the difference between variables involved in this process and the actual distribution of $\mathcal{G}_r$ and π_r. In the following, it helps to think of y as a vertex in the original input G_{r-1} and $id(y)$ as the vertex y maps to in G_r.

- The right distribution of involved variables is (we define $\mathsf{N}^* := (\mathsf{A}^*, \mathsf{B}^*, \mathsf{C}^*)$, $\mathsf{k} := (\mathsf{k}_1, \ldots, \mathsf{k}_{k_r})$, and $\mathsf{W} := (\mathsf{N}^*, \mathsf{k}, \mathsf{L})$ to reduce the clutter):

$$\left(\mathsf{W}, \mathsf{G}^*, \mathsf{M}^{(1)}_{x\to -y}\right) \times \left(\mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}] \mid \mathsf{W}, \mathsf{G}^*, \mathsf{M}^{(1)}_{x\to -y}\right) \times \left(\mathsf{M}^{(1)}_{x\to y} \mid \mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}], \mathsf{W}, \mathsf{G}^*_x, \mathsf{M}^{(1)}_{x\to -y}\right),$$

  where G^*_x in the last term denotes types of all channels incident on x to vertices in G^*. We note that technically, here, we should have also conditioned on all of G^* and not only G^*_x. However, it is easy to see that the distribution of messages sent by x is independent of the rest of G^* once we condition on G^*_x. This is because G^*_x provides the input to x, thus conditioning on G^*_x or G^* is the same.

- And, the distribution of variables in π_{r-1} is:

$$\underbrace{\left(\mathsf{W}, \mathsf{G}^*, \mathsf{M}^{(1)}_{x\to -y}\right)}_{\text{input plus public and pair randomness}} \times \underbrace{\left(\mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}] \mid \mathsf{W}\right)}_{\text{public randomness}} \times \underbrace{\left(\mathsf{M}^{(1)}_{x\to y} \mid \mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}], \mathsf{W}, \mathsf{G}^*_{x,y}\right)}_{\text{pair randomness}}$$

  where $\mathsf{G}^*_{x,y}$ (after conditioning on N^*) will simply be the type of the pair (x, y).

Let us focus on the third terms for now. Bounding the difference between these terms (as before using Fact 4.3.10 and Fact 4.3.4) boils down to upper bounding

$$\mathbb{I}(\mathsf{M}^{(1)}_{x\to y}; \mathsf{G}^*_{x,-y}, \mathsf{M}^{(1)}_{x\to -y} \mid \mathsf{W}, \mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}], \mathsf{G}^*_{x,y}) = \qquad (\mathsf{G}^*_{x,-y} \text{ is } \mathsf{G}^*_x \text{ minus } \mathsf{G}^*_{x,y})$$

$$\mathbb{I}(\mathsf{M}^{(1)}_{x\to y}; \mathsf{G}^*_{x,-y} \mid \mathsf{W}, \mathsf{G}^*_{x,y}, \mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}]) + \mathbb{I}(\mathsf{M}^{(1)}_{x\to y}; \mathsf{M}^{(1)}_{x\to -y} \mid \mathsf{W}, \mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}], \mathsf{G}^*_x), \tag{7.4}$$

where the equality is a direct application of chain rule (Fact 4.3.1-(7)). Notice that the LHS here is quite similar to the LHS of Eq (7.2): modulo the extra conditioning on some new random variables, LHS of Eq (7.4) also measures the correlation of message of x to y with its remaining input $G^*_{x,-y}$ as well as x's other messages (in Eq (7.2) also, we could have replaced $G_{-(x,y)}$ by $G_{x,-y}$). While we (provably) could have not bound the RHS of Eq (7.2) (in particular its second term), we can bound the RHS of Eq (7.4) with the help of these extra conditionings.

Before getting to the sketch of the proof, it is worth briefly checking the protocol outlined at the end of the last subsection when discussing the second term of Eq (7.2). For that specific protocol, the moment we condition on any message of x in $M^{(1)}_x[L \cap \{< id(y)\}]$ (as in the RHS of Eq (7.4)), we already have fixed the random coin toss of the protocol across its different messages and thus can at least hope to say messages $M^{(1)}_{x\to y}$ and $\mathsf{M}^{(1)}_{x\to -y}$ have low correlation.

First term in RHS of Eq (7.4). We have

$$\begin{aligned}
&\mathbb{I}(\mathsf{M}^{(1)}_{x\to y}\,;\mathsf{G}^{*}_{x,-y} \mid \mathsf{W}, \mathsf{G}^{*}_{x,y}, \mathsf{M}^{(1)}_{x}[\mathsf{L}\cap\{<\mathsf{id}(y)\}]) \\
&\quad= \mathbb{I}(\mathsf{M}^{(1)}_{x\to y}\,;\mathsf{G}^{*}_{x,-y} \mid \mathsf{W}\setminus\mathsf{k}, \mathsf{k}_1,\ldots,\mathsf{k}_{k_r}, \mathsf{G}^{*}_{x,y}, \mathsf{M}^{(1)}_{x}[\mathsf{L}\cap\{<\mathsf{id}(y)\}]) \\
&\qquad\qquad\qquad

given that $\mathsf{M}^{(1)}_{x\to -y}$ contains $(n_{r-1}-1)$ messages with $\mathsf{bw}(\pi_r)$ bits each. Finally, by the choice of ℓ_r in Eq (7.3), we can bound the contribution of this term, across all $x, y \in G^*$, by $1/\text{poly}(n_{r-1})$.

Back to the distributions. We are not done yet however as we only compared the third terms in the distributions of variables induced by π_{r-1} versus their actual distribution in π_r. We now need to compare the second terms also, namely, bound

$$\mathbb{I}(\mathsf{M}^{(1)}_x[\mathsf{L} \cap \{< \mathsf{id}(y)\}]\,; \mathsf{G}^*, \mathsf{M}^{(1)}_{x\to -y} \mid \mathsf{W}),$$

which is the information revealed by some of the messages of x about the inner graph G^* (and some other messages). But this seems to bring us to the very beginning: we again have a collection of publicly sampled messages and need to bound their correlation with the inner special sub-instance. The **key observation** here is that we moved away from messages communicated over channels of G^* and now we can indeed hope that at least messages $M^{(1)}_x[L]$ sampled for x (and eventually other vertices in G^*) which are going to channels *outside* of G^* may not be too correlated with G^*. So, while the current protocol does not handle this part, we can hope to fix this issue using an argument similar to Section 7.3.1 which, unlike in Section 7.3.2.2, is no longer doomed to fail.

7.3.2.5 The Final Attempt: Our Round Elimination Protocol

At this point, hopefully, we have provided enough intuition about the need for the rather peculiar sequence of sampling of variables in our final round elimination protocol. Describing this protocol at the level of details of previous subsections requires us to provide another lengthy set of definitions. Instead, equipped with the lessons and observations of previous subsections, we only state our final protocol π_{r-1} at a very high-level as follows.

Public randomness. The players of π_{r-1} use public randomness to sample

- (A^*, B^*, C^*) defines the vertices of the inner graph G^* and allows players to map their input G_{r-1} to this induced subgraph;
- $(J^x_1, \ldots, J^x_{j_r})$ are full $(r-1)$-round instances sampled from $\mathcal{G}_{r-1}$ for each vertex x of G^* (similar in spirit to Section 7.3.1 but now for each individual vertex);
- $(K^{x,y,t}_1, \ldots, K^{x,y,t}_{k_r})$ are partial $(r-1)$-round instances that for each pairs of vertices $x, y \in G^*$ are sampled from $(\mathcal{G}_{r-1} \mid \mathit{type}(x,y) = t)$[6] so that together with the type of (x, y) they can form a full $(r-1)$-round instance (similar to Section 7.3.2.3);
- $(L^{x,t}_1, \ldots, L^{x,t}_{\ell_r})$ are sets of neighbors for each vertex $x \in G^*$ and a fixed type $t \in [r] \cup \{0\}$; (similar to Section 7.3.2.4);
- M_{public} contains publicly sampled messages to subsets of $(L^{x,t}_1, \ldots, L^{x,t}_{\ell_r})$ from each $x \in G^*$ and type $t \in [r] \cup \{0\}$ (similar to Section 7.3.2.4).

[6] Only x and y know $\mathit{type}(x,y)$ and so to sample these sets using public randomness, we instead need to sample the sets for all possible types from $[r] \cup \{0\}$ and then let x, y pick the right type for themselves.

Pair randomness. Each pair of vertices $(x, y) \in G^*$ samples the messages $M^{(1)}_{x,y}$ conditioned on $type(x, y)$ and all publicly sampled variables (similar to Section 7.3.2.3 and Section 7.3.2.4).

Private randomness. By the above steps, each vertex $x \in G^*$ has sampled many of its channels in G. It then simply samples any remaining channels so that it has d_r channels for each type in each layer, conditioned on all the publicly sampled variables and all variables sampled with pair randomness that are known to x (i.e., are related to channels incident on x).

The parameters of the protocol are chosen as follows (notice that the dependencies are acyclic):

$$\ell_r = \text{poly}(n_{r-1}), \quad k_r = \text{poly}(\ell_r), \quad j_r = \text{poly}(k_r), \quad d_r = \text{poly}(j_r), \quad n_r = \text{poly}(d_r).$$

These variables then, using a combination of ideas outlined in the previous subsections, allow us to

1. L-variables and M_{public}: break the correlation between messages $M^{(1)}_{x,y}$ sampled by pair randomness for all pairs $(x, y) \in G^*$;
2. K-variables: break the correlation between messages $M^{(1)}_{x,y}$ sampled by pair randomness and input graph $G^*_{-(x,y)}$ for all pairs $(x, y) \in G^*$;
3. J-variables: break the correlation between M_{public} sampled by public randomness and G^*.

Simulation step. We are not yet done because we also need to handle messages and inputs of players in G that are outside G^*. While for some other problems, this can be quite challenging (see, e.g. [AKZ22] and their "partial-input embedding" technique for handling this), for us this step is quite straightforward[7]. Any vertex x in $G \setminus G^*$ only has a single channel by construction and this channel is to a vertex $y \in G^*$; thus, in our protocol, either the channel (x, y) has been sampled publicly or vertex y has sampled this channel privately. In either case, y knows the entire input of x and since x only communicates with y, vertex y can completely simulate the work of x on its own by sampling its message also.

This means that players of π_{r-1} on G_{r-1} have all the information needed to simulate the protocol π_r on the graph G they have collectively created, use the sampled messages $M^{(1)}$ for the first round of π_r without themselves communicating at all, and then communicating messages of π_r in its remaining $(r-1)$ rounds as part of their own $(r-1)$ rounds of communication. This way, and using our analysis for the sampling steps, we can prove that π_{r-1} satisfies

$$\mathsf{suc}(\pi_{r-1}, \mathcal{G}_{r-1}) \geq \mathsf{suc}(\pi_r, \mathcal{G}_r) - 1/\text{poly}(n_{r-1}),$$

using only $(r-1)$ rounds and a similar bandwidth as π_r.

[7]This also means our techniques and [AKZ22], besides their starting point in [ANRW15], are almost entirely disjoint: all our efforts in this paper is in the handling of "special inner" vertices whereas in [AKZ22], this part is done exactly as in [ANRW15] and instead their main focus is on the remaining vertices.

Continuing like this allows us to obtain a 0-round protocol for $\mathcal{G}_0$ with success probability much better than 7/8 (which is easy to show is the optimal bound on the distribution $\mathcal{G}_0$ for 0-round protocols), a contradiction. Finally, in terms of parameters, given that the size of instances grows polynomially from $\mathcal{G}_{r-1}$ to $\mathcal{G}_r$, we have

$$n_r \geq 2^{2^{\Omega(r)}}$$

which means an r-round lower bound on $\mathcal{G}_r$ translates to an

$$r = \Theta(\log \log n_r)$$

lower bound as desired by Theorem 5.

7.4 A Hard Distribution and its Properties

Throughout, we use $\mathcal{G}_r(n_r)$ to denote the hard distribution for r-rounds over tripartite graphs with n_r vertices in each layer. In this section, we describe our hard distribution and some of its properties, and prove the following theorem (minus its technical details) that formalizes Theorem 5.

Theorem 12. *There exists a family of distributions $\{\mathcal{G}_r(n_r)\}_{r\geq 0}$ over tripartite graphs with n_r vertices in each layer, such that for $n_r > r^{4\cdot 34^r}$, any deterministic protocol π_r with,*

$$\mathsf{round}(\pi_r) = r \qquad \text{and} \qquad \mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) \geq 15/16,$$

must have,

$$\mathsf{bw}(\pi_r) \geq (n_r)^{(1/2)\cdot(1/34^r)} \cdot (480)^{-2}.$$

Let us see how the hardness of distribution $\mathcal{G}_r(n_r)$ proves Theorem 5.

Proof of Theorem 5. By the easy direction of Yao's minimax principle (i.e, an averaging argument), it is sufficient to argue that for deterministic protocols using $o(\log \log n)$ rounds, $O(\log n)$ length messages are not sufficient to solve triangle detection with probability of success at least 15/16 when input graphs are drawn from distribution $\mathcal{G}_r(n_r)$.

In Theorem 12, when $r = o(\log \log n_r)$, we know that the initial condition of $n_r > r^{4\cdot 34^r}$ is satisfied. The bandwidth required, however, is,

$$(n_r)^{(1/2)\cdot(1/34^r)} \cdot (480)^{-2} \gg \text{poly}\log(n_r),$$

which is more than the $O(\log n)$ bandwidth allotted in the CONGEST model. ∎

We begin by describing the hard distribution for 0-round protocols, and we recursively construct hard distributions for r-rounds using the distribution for $(r-1)$-rounds. We also talk about the joint distribution of the input and first round messages to set us up for the analysis.

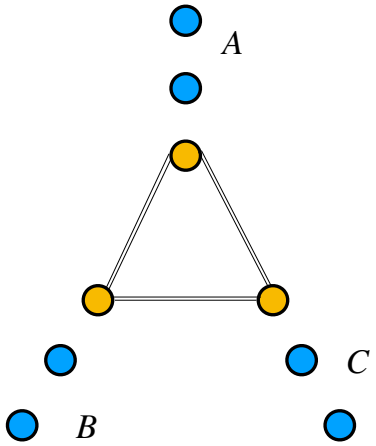


Figure 7.4: An illustration of an instance from the hard distribution for 0-rounds with $n_0 = 3$. The middle (yellow) vertices are a^*, b^*, c^*. The double line indicates that the edge is present with probability $1/2$.

7.4.1 Base Case: Hard Distribution for 0-rounds

In this subsection, we will describe the hard distribution for 0-rounds, and prove some of its necessary properties. By a 0-round protocol, we mean that there is no communication and the players must output whether a triangle exists just based on their own input. This will serve as the base case for our inductive proof of Theorem 12.

Distribution 2. Distribution $\mathcal{G}_0(n_0)$ over graphs G for 0-round protocols: (See Figure 7.4 for an illustration.)

(1) Start with a tripartite graph with n_0 vertices in each layer, and with vertex set $V = A \sqcup B \sqcup C$, and $A = \{a_1, a_2, \ldots, a_{n_0}\}, B = \{b_1, \ldots, b_{n_0}\}, C = \{c_1, \ldots, c_{n_0}\}$.

(2) Choose three random vertices $a^* \in A$, $b^* \in B$ and $c^* \in C$.

(3) For all three pairs of $(a^*, b^*), (b^*, c^*), (c^*, a^*)$, add an edge between each pair with probability $1/2$ uniformly at random and independently.

First, we prove a simple property about 0-round instances sampled from Distribution 2.

Observation 7.4.1. *In any graph $G \sim \mathcal{G}_0(n_0)$, a triangle exists between vertices $\{a^*, b^*, c^*\}$ with probability $1/8$, and there is no triangle otherwise.*

Proof. The edges are only between a^*, b^*, and c^* and exist independently with probability $1/2$. ∎

It is straightforward that protocols which do not communicate at all cannot have a high probability of success for 0-round instances; we formalize this statement next for completeness.

Claim 7.4.2. *Any deterministic protocol π with 0-rounds detects whether a triangle exists in $G \sim \mathcal{G}_0(n)$ with success probability at most 7/8.*

Proof. In the following, we additionally condition on any fixed choice of (a^*, b^*, c^*) and assume this choice is known to all players. This can only strengthen the lower bound.

First consider any vertex $w \notin \{a^*, b^*, c^*\}$. If w outputs *Yes*, namely, that there is a triangle in G, then, by Observation 7.4.1, the answer is only correct with probability 1/8 as the choice of edges between a^*, b^*, c^* is independent. So, we assume all vertices other than (a^*, b^*, c^*) output *No*.

Now, consider any vertex $x^* \in \{a^*, b^*, c^*\}$. When degree of x^* is anything other than two, it can simply output *No* and will be correct in this case. There are now two cases:

- *All* vertices in $\{a^*, b^*, c^*\}$ still output *No* even when their degree is two: in that case, the protocol always outputs *No* (by our assumption earlier) and thus is wrong with probability 1/8 by Observation 7.4.1.
- At least one vertex $x^* \in \{a^*, b^*, c^*\}$ outputs *Yes* when its degree is two: in that case, with probability half the edge between $y^* \neq z^* \in \{a^*, b^*, c^*\}$ may not appear, thus making the protocol wrong with probability 1/2 in this case. The probability that x^* has degree two here is 1/4 and conditioned on this, its output will be wrong with probability 1/8. Thus, in this case also the protocol is wrong with probability 1/8 again.

Overall the success probability of the protocol is at most 7/8. ∎

7.4.2 Hard Distribution for r-rounds

In this subsection, we describe the hard distribution for r-round protocols for $r \geq 1$.

Notation. All vertices in the graph know which layer among $\{A, B, C\}$ they belong to. Each vertex in $X \in \{A, B, C\}$ has a unique identity $x_i \in X$ for $i \in [n_r]$, which distinguishes it from other vertices in X.

For $i \in [n_r]$, the input of each vertex $x_i \in X$ is given as two ordered vectors of length n_r each labeled as $\mathcal{N}^{x_i \to Y}$ and $\mathcal{N}^{x_i \to Z}$, denoting the types of all vertex pairs (x, w) for $w \in Y$ and $w \in Z$, respectively. The j^{th} entry in the vector corresponding to layer Y (denoted by $\mathcal{N}^{x_i \to Y}[y_j]$) contains the type of the vertex pair $x_i, y_j \in [r+1] \cup \{0\}$, for $j \in [n_r]$ (and similarly for $\mathcal{N}^{x_i \to Z}$ and Z).

Sometimes, we also use $\mathcal{N}^{x_i}$ to denote the vectors of x_i together. We use $\mathcal{N}^{x_i}[S]$ for any $S \subseteq Y \cup Z$ to denote the types of all the vertex pairs of the form (x_i, w) for $w \in S$.

To avoid confusion, we sometimes write $X(G)$ to denote the layer $X \in \{A, B, C\}$ of the graph G (to specify the graph). See also our note about the notation of X, etc. in Section 7.2.

Distribution 3. Distribution $\mathcal{G}_r(n_r)$ over graphs G for r-round protocols with $r \geq 1$:
(See Figure 7.5 for an illustration.)

(1) Sample a graph $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ for some parameter n_{r-1} to be fixed later in Eq (7.5).

(2) For each $X \in \{A, B, C\}$ in G, sample n_{r-1} distinct indices $X^* = \{x_1^*, x_2^*, \ldots, x_{n_{r-1}}^*\}$ from X uniformly. The vertex x_i in layer $X(G_{r-1})$ of G_{r-1} takes on the identity x_i^* in G.

(3) Set $type(x_i^*, y_j^*)$ in G to be $type(x_i, y_j)$ for any x_i, y_j in different layers of G_{r-1} for $i, j \in [n_{r-1}]$.

(4) Sample $2n_{r-1} \cdot (r+1)$ subsets $S_t^{y \to X}, S_t^{z \to X} \subseteq X \setminus X^*$, for each $y, z \in G_{r-1}$, and type $t \in [r] \cup \{0\}$, all disjoint from each other. The sets are of size d_r each, and are sampled uniformly at random. Parameter d_r will be fixed later in Eq (7.5).

(5) For $i \in [n_{r-1}]$ and $x_i \in X = X(G_{r-1})$, each other layer $Y \neq X$, and type $t \in [r] \cup \{0\}$, add channels of type t between x_i^* to just enough sampled vertices from $S_t^{x_i \to Y}$ so that the total number of neighboring channels between x_i^* to any other layer Y of type t is exactly d_r.[a]

(6) Set the type of all pairs of vertices, the types of which have not been fixed yet to be $r+1$.

[a] x_i^* may have some type-t channels from G_{r-1}, so we add more channels to increase its type-t channels to d_r.

Let us fix the parameters in the hard distribution as follows,

$$n_{r-1} = (n_r)^{1/34} \qquad d_r = (n_{r-1})^{13}. \tag{7.5}$$

We sample n_{r-1} vertices from X for each $x_i \in G_{r-1}$ in step (1). We also sample d_r distinct vertices for each y_i, z_j and type t with $i, j \in [n_{r-1}]$, and $t \in [r] \cup \{0\}$ in step (4). In total,

$$n_{r-1} + (r+1) \cdot d_r \cdot 2n_{r-1}$$

distinct vertices are sampled from X uniformly at random. We have to check that n_r is large enough to make this sampling process feasible:

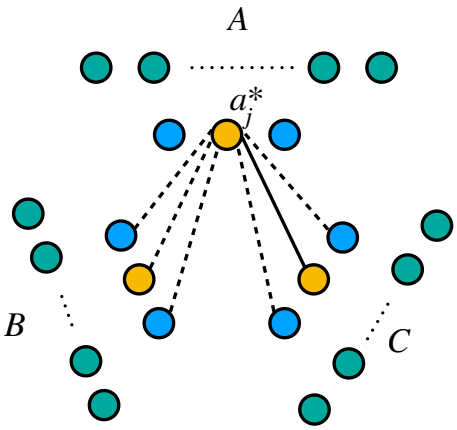
$$n_{r-1} + (r+1) \cdot d_r \cdot 2n_{r-1} = n_{r-1}(2d_r(r+1)+1) \leq n_{r-1}^{16} < n_r.$$

Thus, we have more than enough room in n_r for step (4).

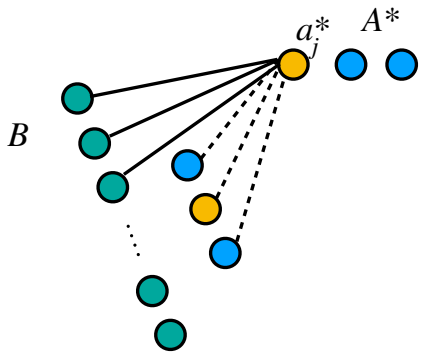
We call the vertices in X^*, Y^*, Z^* to be **inner vertices** (sampled in step (2)), and all other vertices as **outer vertices**. We refer to the graph G_{r-1} from step (1) as the **inner graph**. We prove some simple properties of $\mathcal{G}_r(n_r)$ now.

Observation 7.4.3. *The number of neighboring channels of type $t \in [r] \cup \{0\}$ of any inner vertex in G is exactly d_r. The channel-degree of all outer vertices is at most one.*

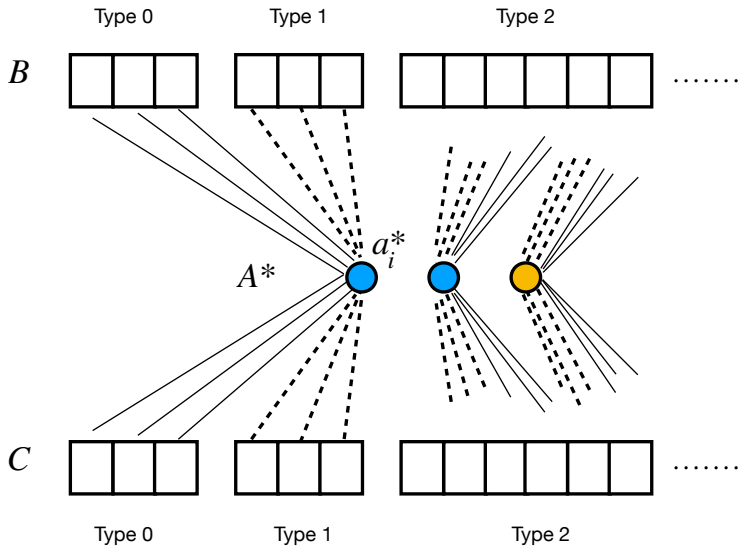
Proof. In step (5), we ensure the number of channels of each type $t \in [r] \cup \{0\}$ for each inner



(a) Realization of all types of channels between one inner vertex a_j^* to other layers. The other inner vertices (blue and yellow) also have simlar channels to each other if they are in different layers.



(b) Addition of channels of type 0 for one inner vertex a_j^* to outer vertices to ensure correct channel-degree for all types (some parts of the instance are omitted).



(c) An illustration of the input of one inner vertex. The other inner vertices are connected only to outer vertices u such that type of (a_i^*, u) is 2. Vertex a_i does not know which vertices in B, C are inner vertices. In the figure, only A^* and the relevant vertices of B, C are shown.

Figure 7.5: An illustration of some parts of the 1-round instance. Inner vertices from 0-round instance are blue and yellow (in accordance with Figure 7.4), and outer vertices are green. We used $n_0 = d_1 = 3$ for illustration, but this is certainly not true in the actual instance. Dashed lines indicate a channel of type 1, straight lines indicate a channel of type 0, and other pairs of vertices with no lines are of type 2. The input of blue and yellow vertices of the inner graph have exactly 3 channels of type 0 and type 1 in total.

vertex is exactly d_r explicitly, by adding enough channels to sets sampled in step (4). Any outer vertex gets sampled at most once in the sets of step (4), as all the sets are disjoint. Therefore all outer vertices have channel-degree at most one. ∎

We only ever refer to the input of inner vertices to talk about graph $G \sim \mathcal{G}_r(n_r)$, due to the following observation.

Observation 7.4.4. *In any $G \sim \mathcal{G}_r(n_r)$, fixing the identity x_i^* and the two n_r size neighborhood vectors, $\mathcal{N}^{x\to Y}, \mathcal{N}^{x\to Z}$ for all inner vertices $x_i \in G_{r-1}$ fixes the graph G.*

Proof. The only randomness involved in the distribution $\mathcal{G}_r(n_r)$ is in the graph G_{r-1}, the identities chosen for each inner vertex, and the at most $(r+1)\cdot d_r$ many neighboring channels sampled for each inner vertex. Thus, the variables in the observation statement fix the graph G. ∎

We can say something further about the structure of $\mathcal{N}^{x\to Y}$ for every inner vertex x and layer Y with $x \notin Y$.

Observation 7.4.5. *For every inner vertex $x \in G_{r-1}$, the marginal distribution of $\mathcal{N}^{x\to Y}$ for every other layer Y is obtained by:*

- *Setting the types of all vertices in Y^* and Z^* based on the type in G_{r-1}.*
- *Sampling the rest of $\mathcal{N}^x$ uniformly random, conditioned on the types of $Y^* \cup Z^*$ so that there are exactly d_r vertices of type $t \in [r] \cup \{0\}$, and $n_r - d_r(r+1)$ vertices of type $r+1$.*

Proof. We know that the number of channels of type $t \in [r] \cup \{0\}$ for each vertex $x_i \in G_{r-1}$ is exactly d_r from Observation 7.4.3. Therefore the total number of channels with types in $[r] \cup \{0\}$ is $(r+1)\cdot d_r$. The remaining $n_r - d_r \cdot (r+1)$ channels are of type $r+1$.

The sets $S_t^{x\to Y}$ and $S_t^{x\to Z}$ are also chosen uniformly at random, so that they are disjoint from Y^* and Z^*. Hence, the final distribution of $\mathcal{N}^{x\to Y}$ is such that uniformly random subsets are chosen for each type. The total number of vertices in Y^* is exactly n_{r-1}, and even if all of them had the same type, we know that there is still enough room for d_r vertices of each type $t \in [r] \cup \{0\}$ as $d_r > n_{r-1}$ from Eq (7.5). ∎

The identities of inner vertices are chosen at random, so the distribution of the neighborhoods are symmetric over all n_r vertices of X. More formally, we have the following observation.

Claim 7.4.6. *For $r \geq 0$, in $\mathcal{G}_r(n_r)$ for all $i \in [n_{r-1}]$ and $x_i \in X(G_{r-1})$, the marginal distribution of the two neighborhood vectors given to the inner vertex x_i in G_{r-1} is independent of x and i.*

Proof. It is sufficient to show that the marginal distribution of the input given to all vertices in distribution $\mathcal{G}_{r-1}(n_{r-1})$ are the same, and independent of x and i for all $r \geq 1$. We prove the claim by induction on r. For the base case when $r = 1$, the inner graph is sampled from

$\mathcal{G}_0(n_0)$. We know that the vertex x^* is chosen uniformly at random from $[n_0]$. Therefore, each vertex in layer X has equal probability of being chosen as x^* in $\mathcal{G}_0(n_0)$.

For any $r > 1$, we know that all the identities of inner vertices $x_1^*, x_2^*, \ldots, x_{n_{r-2}}^*$ in $\mathcal{G}_{r-1}(n_{r-1})$ are chosen uniformly at random from $[n_{r-1}]$. Therefore, any vertex from $X(G_{r-1})$ is equally likely to be chosen as the identity of a vertex from G_{r-2} (the inner graph of G_{r-1}). The marginal distribution of the neighborhood in G_{r-2} is identical by induction hypothesis.

For the outer vertices in X, again, the sets sampled in step (4) are uniform over $[n_{r-1}]$, and vertices from $[n_{r-1}]$ are equally likely to be chosen to be a part of these sets in distribution $\mathcal{G}_{r-1}(n_{r-1})$. Our construction is fully symmetric over the three layers and the vertices in each layer. Therefore, the marginal distribution of the input given to each vertex in graph G_{r-1} is the same and independent of x and i. ∎

Lastly, we need to argue about the existence of triangles in $G \sim \mathcal{G}_r(n_r)$.

Observation 7.4.7. *In any graph $G \sim \mathcal{G}_r(n_r)$, a triangle exists if and only if a triangle is present in G_{r-1} sampled in step (1) while sampling G.*

Proof. Firstly, the channel-degree of all outer vertices is at most one by Observation 7.4.3. Thus, all of them have at most one edge also. Hence, only the inner vertices can be a part of a triangle. We do not add or remove any edges between inner vertices, we only add channels which do not form edges. Hence, triangle existence in G_{r-1} is preserved in graph G. ∎

7.4.3 Another Way of Sampling from Hard Distribution

In this subsection, we talk more about the hard distribution $\mathcal{G}_r(n_r)$, and give an alternate way of sampling it. This makes the analysis much easier, as it gives us another perspective of looking at the hard distribution. This alternate way is not without loss, however, but we show that these distributions are quite close in total variation distance.

We only ever talk about the input and identities of inner vertices, as this fixes distribution $\mathcal{G}_r(n_r)$ by Observation 7.4.4.

Notation. We use id^X to denote the variables $x_1^*, x_2^*, \ldots, x_{n_{r-1}}^*$ (elements of X^*) for $X \in A, B, C$. We use ids to denote the variables (id^A, id^B, id^C) collectively. We use $id(x_i)$ to denote x_i^* for $i \in [n_{r-1}]$. We use $\mathcal{N}_{\text{in}}^{x_i}$ to denote the input of inner vertex $x_i \in G_{r-1}$. This is two vectors of length n_{r-1} each, containing types to layers Y and Z in G_{r-1}.

We use $\mathcal{D}_{\text{in}}$ to denote the marginal distribution of $\mathcal{N}_{\text{in}}^{x_i}$ for each inner vertex x_i. Formally, $\mathcal{D}_{\text{in}}$ is the distribution of $\mathcal{N}^{x_i}$ in an instance $G \sim \mathcal{G}_{r-1}(n_{r-1})$ for $i \in [n_{r-1}]$. Note that this is identical for each $i \in [n_{r-1}]$ by Claim 7.4.6.

We need to define three more parameters to proceed.

$$\alpha_r = (n_{r-1})^{11} \qquad \beta_r = (n_{r-1})^5 \qquad \gamma_r = (n_{r-1})^6. \tag{7.6}$$

We know that d_r is much larger than n_{r-1} from Eq (7.5). However, we want to argue that, for any inner vertex $x_i \in G_{r-1}$, given only its input $\mathcal{N}^{x_i}$ in graph G, there are multiple choices for what the other inner vertices may be; formally, we need the following for each inner vertex $x_i \in G_{r-1}$:

- An α_r-size collection of sets of size $2n_{r-1}$ each, all of which are distributed independently and exactly according $\mathcal{D}_{\text{in}}$ with channels to x_i.
- For each type $t \in [r] \cup \{0\}$ and any other inner vertex $y \notin X(G_{r-1})$, a β_r-size collection of sets of size $2n_{r-1} - 1$ each, all of which are distributed independently of each other according to the $(r-1)$-round input on $2n_{r-1} - 1$ vertices, conditioned on the channel to the one remaining vertex being of type t.
- For each type $t \in [r] \cup \{0\}$ and any other inner vertex $y \notin X(G_{r-1})$, a set of size γ_r with channels of type t to x_i.

We show it is possible to sample from $\mathcal{G}_r(n_r)$ while satisfying these conditions with high probability.

Distribution 4. Distribution $\widetilde{\mathcal{G}}_r(n_r)$ for r-round protocols with $r \geq 1$:

(1) Sample G_{r-1} and sets X^*, Y^*, Z^* as before from Distribution 3.

(2) Fix any inner vertex $x \in G_{r-1}$, and from the other sets $Y \setminus Y^*$ and $Z \setminus Z^*$, sample the following subsets, disjoint from each other for all x: (Observation 7.4.9 shows that this is feasible)

(a) A collection $\mathcal{J}^x = (J_1^x, \ldots, J_{\alpha_r}^x)$, where each J_i^x for $i \in [\alpha_r]$ is a set containing n_{r-1} elements of $Y \setminus Y^*$ and n_{r-1} elements of $Z \setminus Z^*$.

(b) For each type $t \in [r] \cup \{0\}$, value $i \in [n_{r-1}]$, each other layer Y, Z a collection

$$\mathcal{K}_{t,i}^{x \to Y} = (K_{t,i,1}^{x \to Y}, \ldots, K_{t,i,\beta_r}^{x \to Y})$$

where each $K_{t,i,j}^{x \to Y}$ for $j \in [\beta_r]$ is a set containing $(n_{r-1} - 1)$ elements from $Y \setminus Y^*$ and n_{r-1} elements from $Z \setminus Z^*$. Similarly, $\mathcal{K}_{t,i}^{x \to Z}$ is a collection of β_r-sets, each with n_{r-1} elements from $Y \setminus Y^*$ and $n_{r-1} - 1$ elements from $Z \setminus Z^*$.

(c) For each type t and value $i \in [n_{r-1}]$ and each other layer Y, a set $L_{t,i}^{x \to Y}$ of γ_r elements from $Y \setminus Y^*$.

(3) For each inner vertex $x \in G_{r-1}$, do the following:

(a) For each $i \in [\alpha_r]$, sample $(\mathcal{N}^x[J_i^x])$ from $\mathcal{D}_{\text{in}}$ independently of other i.

(b) For each $t \in [r] \cup \{0\}$, each other layer Y, $i \in [n_{r-1}]$ and $j \in [\beta_r]$, sample $\mathcal{N}^x[K_{t,i,j}^{x \to Y}]$ so that $\mathcal{N}^x[K_{t,i,j}^{x \to Y} \cup \{y_i^*\}]$ is sampled from $\mathcal{D}_{\text{in}}$, conditioned on $type(x, y_i^*) = t$.

(c) For each $t \in [r] \cup \{0\}, i \in [n_{r-1}]$, and other layer Y, set $\mathcal{N}^{x \to Y}[y]$ for each $y \in L_{t,i}^{x \to Y}$ to be type t.

(d) Sample the rest of the input of $\mathcal{N}^x \sim \mathcal{G}_r(n_r)$ conditioned on the random variables sampled in earlier steps independent of the input of other inner vertices (see Claim 7.4.10.)

(4) For all pairs of vertices whose types are not fixed yet, set them to be of type $r+1$.

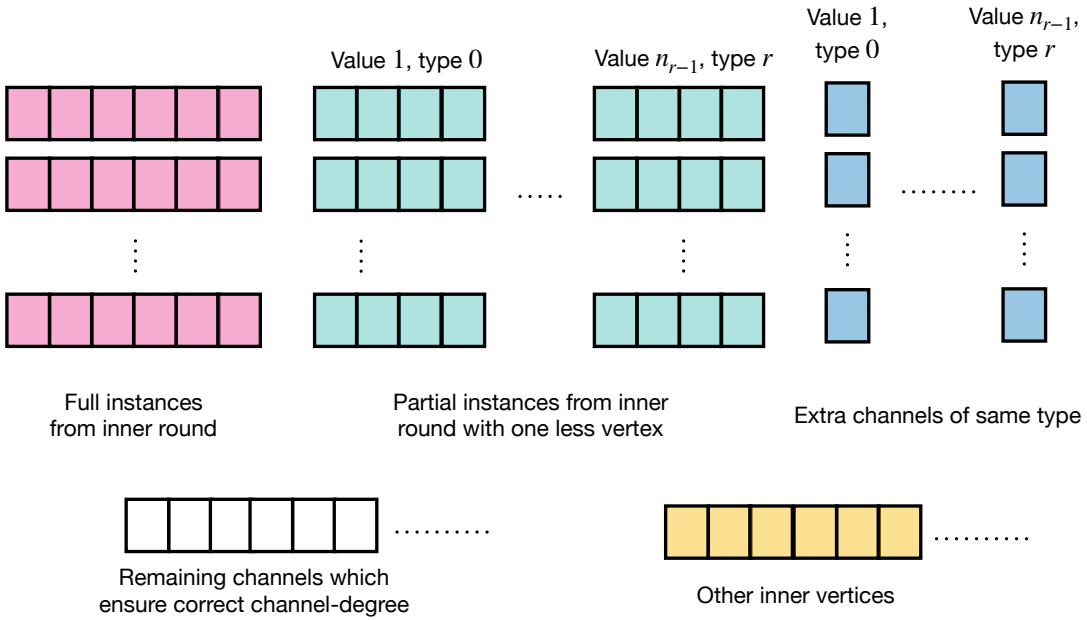


Figure 7.6: An illustration of what the input of one inner vertex x looks like in $\widetilde{\mathcal{G}}_r$. Pink outer vertices correspond to collections $\mathcal{J}$, green outer vertices correspond to collections $\mathcal{K}$ (only for one other layer Y are shown, this is repeated once more), blue outer vertices correspond to sets L (again, only for Y are shown, this is repeated for Z), yellow vertices correspond to vertices in $Y^* \cup Z^*$, and the unshaded vertices are the other outer vertices with various types to ensure correct number of channels of each type $t \in [r] \cup \{0\}$.

The parameters in Eq (7.5) and Eq (7.6) are chosen so that the sets in step (2) can be sampled:

Observation 7.4.8. *In step (2) of Distribution 4, for each inner vertex x, and layer Y with $x \notin Y$, the total number of identities sampled is at most $2(n_{r-1})^{12}$.*

Proof. For each inner vertex $x \in X$, to another layer Y, we sample:

- α_r-many sets of size n_{r-1} each.
- A collection of β_r-many sets of size at most n_{r-1} each for type $t \in [r] \cup \{0\}$, value $i \in [n_{r-1}]$ and each other layer which does not contain x.
- A collection of size γ_r for $(r+1)$ types and n_{r-1} values.

In total, we sample, at most

$$\alpha_r \cdot n_{r-1} + 2\beta_r \cdot (n_{r-1})^2 \cdot (r+1) + \gamma_r \cdot (r+1) \cdot n_{r-1} \leq (n_{r-1})^{12} + 2(n_{r-1})^8 + (n_{r-1})^8 \leq 2(n_{r-1})^{12}$$
(as $r+1 < n_{r-1}$ for large n_r and by Eq (7.6) and for large enough n_{r-1})

vertices for each inner vertex not in Y. ∎

Observation 7.4.9. *In Step (2) of Distribution 4, for each layer Z, the total number of vertices sampled from $Z \setminus Z^*$ is at most $4(n_{r-1})^{13} < n_r - n_{r-1}$.*

Proof. By Observation 7.4.8, for each inner vertex not in Z, we sample at most, $2(n_{r-1})^{12}$ vertices from Z. There are exactly $2n_{r-1}$ inner vertices which are not in Z. In total, from set $Z \setminus Z^*$, we sample at most $4(n_{r-1})^{13}$ vertices, which is less than $n_r - n_{r-1}$ by construction. ∎

We show that the parameters are set so that step (3) can be performed.

Claim 7.4.10. *In the definition of $\widetilde{\mathcal{G}}_r(n_r)$ from Distribution 4, conditioned on G_{r-1} and values of X^*, Y^* and Z^*, in step (3), each $\mathcal{N}^x$ can be sampled marginally the same way as in distribution $\mathcal{G}_r(n_r)$ from Distribution 3.*

Proof. We will prove that after the sampling process in step (2) is done, it is feasible to sample the input $\mathcal{N}^x$ of any inner vertex x from $\mathcal{G}_r(n_r)$ marginally. We refer to Observation 7.4.5. We know that the types of all vertices in $Y^* \cup Z^*$ are set correctly.

We argue that it is still possible to sample the input so that uniformly random d_r-size subsets are chosen for each type $t \in [r] \cup \{0\}$ for each layer Y, Z.

By Observation 7.4.8, we know that the total number of entries in $\mathcal{N}^{x \to Y}$ for each other layer Y whose types are fixed by the first three steps of (3) are at most $2(n_{r-1})^{12}$, which is still less than d_r by Eq (7.5). Therefore, we have enough room to marginally sample the rest of $\mathcal{N}^x$ so that the conditions in Observation 7.4.5 are satisfied. ∎

Finally, we argue that these distributions are close overall.

Claim 7.4.11. *Distribution $\widetilde{\mathcal{G}}_r(n_r)$ which is obtained by marginally sampling each $\mathcal{N}^{x \to Y}$ as in Distribution 4 is close in total variation distance to $\mathcal{G}_r(n_r)$.*

$$\|\mathcal{G}_r(n_r) - \widetilde{\mathcal{G}}_r(n_r)\|_{\text{tvd}} \leq \frac{1}{n_{r-1}}.$$

Proof. In distribution $\mathcal{G}_r(n_r)$, after the types of vertex pairs (x, y) are set according to G_{r-1} for inner vertices x, y, we sample *disjoint* subsets of the remaining $X \setminus X^*$ to add as neighbors to inner vertices in Y^* and Z^*. This ensures that all outer vertices have channel-degree at most one, as we have stated in Observation 7.4.3.

In distribution $\widetilde{\mathcal{G}}_r(n_r)$, all the subsets of outer vertices sampled in step (2) are disjoint from each other by construction. Even though channels of some type $t \in [r] \cup \{0\}$ are added to all these outer vertices, exactly one channel is added, and they have channel-degree at most one.

However, the rest of the input for each inner vertex x is sampled independenly of the input of other inner vertices in $\widetilde{\mathcal{G}}_r(n_r)$. This process corresponds to sampling at most d_r vertices for each type $t \in [r] \cup \{0\}$ for each inner vertex, so that the condition stated in Observation 7.4.3 is satisfied. Namely, the condition that the number of channels of each type $t \in [r] \cup \{0\}$ for each inner vertex is exactly d_r. Outer vertices sampled for this purpose *may* have collisions, which will cause outer vertices to have channel-degree more than one.

We will show that even though the rest of the outer vertices in step (3) for each inner vertex are sampled independently of each other, it is highly unlikely that some vertex is sampled twice. Let $\mathcal{E}_{\text{uniq}}$ be the event that all the outer vertices sampled to fill up the d_r

slots for each type, and for each inner vertex are unique for all layers X, Y, Z. We will show that probability of $\mathcal{E}_{\text{uniq}}$ happening is large. We have,

$$\Pr[\mathcal{E}_{\text{uniq}}] \leq (\Pr[(2n_{r-1} \cdot (r+1) \cdot d_r + n_{r-1}) \text{ vertices sampled u.a.r. from } [n_r] \text{ do not overlap}])^3,$$

where we bound the probability of a super set of $\mathcal{E}_{\text{uniq}}$. In $\mathcal{E}_{\text{uniq}}$, fewer than $d_r \cdot (r+1)$ vertices are sampled from any layer Y for any inner vertex $x \notin Y$. We will show that even if $d_r \cdot (r+1)$ vertices are sampled for each inner vertex, in addition to the set Y^* uniformly at random and independently of each other, no vertex is sampled more than once except with a very small probability.

We choose $2n_{r-1} \cdot (r+1) \cdot d_r + n_{r-1} < n_{r-1}^{16} := \theta$ vertices uniformly at random and independently from n_r vertices in layer Y. The probability that none of them are sampled more than once is,

$$\binom{n_r}{\theta} \cdot \theta! \cdot \frac{1}{(n_r)^\theta} = \prod_{i=0}^{\theta-1}(1 - i/n_r) \geq \exp\left(-2 \cdot \sum_{i=0}^{\theta} i/n_r\right) = \exp(-(\theta-1) \cdot \theta/n_r).$$

We do this for all three layers A, B and C. The probability that no element is sampled twice in any layer is at least,

$$\Pr[\mathcal{E}_{\text{uniq}}] \geq \exp(-3 \cdot (\theta-1) \cdot \theta/n_r) \geq \exp(-3/n_{r-1}^2) \geq 1 - 1/n_{r-1},$$

where we have used the value of $n_r = n_{r-1}^{34} = \theta^2 \cdot n_{r-1}^2$ from Eq (7.5) and that $n_{r-1} \geq 3$.

Conditioned on event $\mathcal{E}_{\text{uniq}}$, we argue that the distribution of $\mathcal{G}_r(n_r)$ and $\widetilde{\mathcal{G}}_r(n_r)$ are identical. We already know that the distributions of the input of each inner vertex is marginally the same by Claim 7.4.10. When $\mathcal{E}_{\text{uniq}}$ happens, all the outer vertices have channel-degree at most one, and the neighbors of each inner vertex are disjoint barring X^*, Y^*, Z^*. This satisfies all the properties of instances sampled from $\mathcal{G}_r(n_r)$.

Let I_{uniq} be the indicator random variable for event $\mathcal{E}_{\text{uniq}}$.

$$\begin{aligned}
\|\mathcal{G}_r(n_r) - \widetilde{\mathcal{G}}_r(n_r)\|_{\text{tvd}} &\leq \mathop{\mathbb{E}}_{I \sim \mathsf{I}_{\text{uniq}}} \|(\mathcal{G}_r(n_r) \mid \mathsf{I}_{\text{uniq}} = I) - (\widetilde{\mathcal{G}}_r(n_r) \mid \mathsf{I}_{\text{uniq}} = I)\|_{\text{tvd}} \\
&\qquad \text{(by "overconditioning" as in Fact 4.3.7)} \\
&\leq \Pr[\mathsf{I}_{\text{uniq}} = 1] \cdot \|(\mathcal{G}_r(n_r) \mid \mathcal{E}_{\text{uniq}}) - (\widetilde{\mathcal{G}}_r(n_r) \mid \mathcal{E}_{\text{uniq}})\|_{\text{tvd}} + \Pr[\mathsf{I}_{\text{uniq}} = 0] \\
&= 0 + \Pr[\mathsf{I}_{\text{uniq}} = 0] \\
&\qquad \text{(conditioned on } \mathcal{E}_{\text{uniq}}\text{, the distributions are the same)} \\
&\leq \frac{1}{n_{r-1}},
\end{aligned}$$

where, in the last inequality we have used our upper bound on the probability of $\neg\mathcal{E}_{\text{uniq}}$. ∎

We need some more notation about random variables in $\widetilde{\mathcal{G}}_r(n_r)$ for the later sections.

Notation. We use $\mathcal{J}_{\text{all}}$ to denote the joint random variable containing all $\mathcal{J}^x$ for all inner vertices x. Similarly, we use $\mathcal{K}_{\text{all}}, L_{\text{all}}$ to denote the joint random variable $\mathcal{K}_{t,i}^{x \to Y}, L_{t,i}^{x \to Y}$

respectively for all inner vertices x, layer Y with $x \notin Y$, type $t \in [r] \cup \{0\}$, and value $i \in [n_{r-1}]$. We call the random varibles $\mathcal{J}_{\text{all}}, \mathcal{K}_{\text{all}}, L_{\text{all}}$ as **auxiliaries**, and together we use aux to denote them.

Observation 7.4.12. *We have the following independence properties for the inputs of vertices in graph $G \sim \widetilde{\mathcal{G}}_r(n_r)$:*

(i) Conditioned on $(\mathsf{G}_{r-1}, \mathsf{aux}, \mathsf{ids})$, the inputs of each of the inner vertices in G are independent of each other.

(ii) The input $\mathcal{N}^{x\to Y}, \mathcal{N}^{x\to Z}$ for any inner vertex x in the graph G is independent of all other random variables in G_{r-1} conditioned on $\mathcal{N}^x_{\text{in}}, \mathsf{ids}$ and aux.

(iii) The random variable G_{r-1} is independent of ids and aux.

Proof. For part (i), we know that for each inner vertex x, once the new identities ids and auxiliaries aux are fixed, and graph G_{r-1} is fixed, its input is sampled independently of the other inner vertices in step (3).

Part (ii) follows by a similar argument: once $\mathcal{N}^x_{\text{in}}, ids$ and aux are fixed, the entire input of x is fixed by randomness completely independent of G_{r-1}.

For part (iii), it is easy to see that $\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}$ and $\boldsymbol{L}_{\text{all}}$ are sampled from A, B, C in graph G, irrespective of the inner graph G_{r-1}. ∎

7.4.4 Distribution of Input and Messages

In this subsection, we talk about the joint distribution of input $\mathcal{G}_r(n_r)$ and the messages sent across all channels in the first round of communication. Let π_r be a protocol for solving triangle detection on an input G sampled from $\mathcal{G}_r(n_r)$.

Notation. Let $\mathcal{D}^{\text{real}}$ denote the distribution of the inputs and messages in the first round of π_r. For any inner vertex x_i, let $\mathcal{M}^{x_i\to Y}$ denote the n_r-length vector of the messages sent by x_i to its channels in Y. When there is no channel between x_i to some vertex y_j, we use $\perp$ to denote the null message that x_i sends. We use $\mathcal{M}^{\text{out}\to x_i}$ to denote the messages x_i receives from outer vertices. We use $\mathcal{M}_{\text{all}}$ to denote all the messages sent and received by all inner vertices in the first round. Note that this is also all the messages sent over the first round as the type of all pairs of outer vertices is set as $r+1$, and they can send no messages to each other in any round.

We define a distribution $\widetilde{\mathcal{D}^{\text{real}}}$ for the input graph and first round messages below. We will show that this distribution is generated when input G is sampled from $\widetilde{\mathcal{G}}_r(n_r)$, and protocol π_r is run over this graph instead. We will also show that distribution $\widetilde{\mathcal{D}^{\text{real}}}$ is close to $\mathcal{D}^{\text{real}}$.

Distribution 5. Distribution $\widetilde{\mathcal{D}^{\text{real}}}$:

$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}})$ (the inner graph, identities, auxiliaries)

$$\times \Big(\bigtimes_{\substack{x\in\{a,b,c\},\\ i\in[n_{r-1}]}} \boldsymbol{\mathcal{N}}^{x_i\to Y}, \boldsymbol{\mathcal{N}}^{x_i\to Z} \mid \boldsymbol{\mathcal{N}}^{x_i}_{\text{in}}, \mathsf{aux}, \mathsf{ids}\Big) \qquad \text{(the inputs of all inner vertices)}$$

$$\times \Big(\bigtimes_{\substack{x\in\{a,b,c\},\\ i\in[n_{r-1}]}} \boldsymbol{\mathcal{M}}^{x_i\to Y}, \boldsymbol{\mathcal{M}}^{x_i\to Z} \mid \boldsymbol{\mathcal{N}}^{x_i\to Y}, \boldsymbol{\mathcal{N}}^{x_i\to Z}, \mathsf{aux}, \mathsf{ids}\Big)$$

(the messages sent by inner vertices)

$$\times \Big(\bigtimes_{\substack{x\in\{a,b,c\},\\ i\in[n_{r-1}]}} (\boldsymbol{\mathcal{M}}^{\text{out}\to x_i} \mid \boldsymbol{\mathcal{N}}^{x_i\to Y}, \boldsymbol{\mathcal{N}}^{x_i\to Z}, \mathsf{aux}, \mathsf{ids})\Big)$$

(the messages sent by outer vertices to inner vertices)

Observation 7.4.13. *The distribution of inputs in $\widetilde{\mathcal{D}^{\text{real}}}$ is sampled from $\widetilde{\mathcal{G}}_r(n_r)$.*

Proof. In distribution $\widetilde{\mathcal{G}}_r(n_r)$, we know that all the $\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}$ and $\boldsymbol{L}_{\text{all}}$ are sampled conditioned on each other. And, they are independent of G_{r-1} from Observation 7.4.12-*(iii)*. Hence, we have that the distributions of G_{r-1}, ids and aux are exactly as in $\widetilde{\mathcal{G}}_r(n_r)$ in $\widetilde{\mathcal{D}^{\text{real}}}$.

The input of the inner vertices are independent of each other conditioned on G_{r-1}, ids and aux from Observation 7.4.12-*(i)*. Therefore, they can be sampled separately as in $\widetilde{\mathcal{D}^{\text{real}}}$.

Lastly, we know that for any inner vertex x, its input is independent of G_{r-1} when conditioned on $\boldsymbol{\mathcal{N}}^x_{\text{in}}$, ids and aux by Observation 7.4.12-*(ii)*, so the other random variables in G_{r-1} can be ignored also when sampling these inputs. ∎

Observation 7.4.14. *For any graph G sampled from $\mathcal{G}_r(n_r)$, the distribution of all the first round messages is the same in $\mathcal{D}^{\text{real}}$ and $\widetilde{\mathcal{D}^{\text{real}}}$ conditioned on the input graph being G.*

Proof. The protocol π is deterministic, hence the messages sent by any vertex are only a function of its input. For all the inner vertices, in $\widetilde{\mathcal{D}^{\text{real}}}$, the messages are sampled conditioned on their entire input, and are distributed the same as in $\mathcal{D}^{\text{real}}$.

For outer vertices, when input graph G lies in support of $\mathcal{G}_r(n_r)$, we know that the channel-degree of all these vertices is at most one, by Observation 7.4.3. For any outer vertex u connected to inner vertex x, the entire input given to u is known to x. Thus, x can sample the messages that u would send to x only with its input. ∎

Claim 7.4.15. *Distributions $\mathcal{D}^{\text{real}}$ and $\widetilde{\mathcal{D}^{\text{real}}}$ are close to each other:*

$$\|\mathcal{D}^{\text{real}} - \widetilde{\mathcal{D}^{\text{real}}}\|_{\text{tvd}} \le 1/n_{r-1}.$$

Proof. By Observation 7.4.13 and Fact 4.3.8,

$$\begin{aligned}\|\mathcal{D}^{\text{real}} - \widetilde{\mathcal{D}^{\text{real}}}\|_{\text{tvd}} &\le \|\mathcal{G}_r(n_r) - \widetilde{\mathcal{G}}_r(n_r)\|_{\text{tvd}} + \mathop{\mathbb{E}}_{G\sim\mathcal{G}_r(n_r)} \|\mathcal{D}^{\text{real}}(\boldsymbol{\mathcal{M}}_{\text{all}} \mid G) - \widetilde{\mathcal{D}^{\text{real}}}(\boldsymbol{\mathcal{M}}_{\text{all}} \mid G)\|_{\text{tvd}} \\ &\le \frac{1}{n_{r-1}} + \mathop{\mathbb{E}}_{G\sim\mathcal{G}_r(n_r)} \|\mathcal{D}^{\text{real}}(\boldsymbol{\mathcal{M}}_{\text{all}} \mid G) - \widetilde{\mathcal{D}^{\text{real}}}(\boldsymbol{\mathcal{M}}_{\text{all}} \mid G)\|_{\text{tvd}} = \frac{1}{n_{r-1}},\end{aligned}$$

where the last inequality is by Claim 7.4.11 and the equality follows from Observation 7.4.14. ∎

7.4.5 Proof of Theorem 12 Barring Round Elimination

In the next section, we construct a round elimination protocol, with the following parameters.

Lemma 7.4.16. *For any $s, r \geq 1$ and $\delta \in (0,1)$, given a deterministic protocol π_r for instances $G_r \sim \mathcal{G}_r(n_r)$, with the following parameters:*

$$\mathsf{round}(\pi_r) = r \qquad \mathsf{bw}(\pi_r) = s \qquad \mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) \geq \delta,$$

we can construct a deterministic protocol π_{r-1} which takes instances $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ and has the following parameters:

$$\mathsf{round}(\pi_{r-1}) = r-1 \qquad \mathsf{bw}(\pi_{r-1}) \leq s \qquad \mathsf{suc}(\pi_{r-1}, \mathcal{G}_{r-1}(n_{r-1})) \geq \delta - \frac{1}{n_{r-1}} - 15\sqrt{\frac{s}{n_{r-1}}}.$$

The proof of Lemma 7.4.16 is the focus of Section 7.5. We now prove Theorem 12 using Lemma 7.4.16.

Proof of Theorem 12. Assume towards a contradiction that there exists a protocol π_r with r-rounds,

$$\mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) \geq 15/16, \qquad \text{and} \qquad \mathsf{bw}(\pi_r) = s < n_r^{(1/2)\cdot(1/34^r)} \cdot \frac{1}{(480)^2}.$$

We prove the theorem by repeatedly applying Lemma 7.4.16 for r times on protocol π_r, to get a protocol π_0 for $\mathcal{G}_0(n_0)$. The value of n_0, using Eq (7.5) is $n_0 = {n_1}^{1/34} = \cdots = {n_r}^{1/34^r}$. From the large value of n_r in the statement of the theorem, we have,

$$n_0 = n_r^{1/34^r} > r^4. \tag{7.7}$$

The probability of success of π_0 is at least,

$$\begin{aligned}
&\mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) - \sum_{\ell=1}^{r} \frac{1}{n_{r-\ell}} - 15 \cdot \sqrt{s} \cdot \sum_{\ell=1}^{r} \frac{1}{(n_{r-\ell})^{1/2}} \\
&\geq \frac{15}{16} - \sum_{\ell=1}^{r} \frac{1}{n_{r-\ell}} - 15 \cdot \sqrt{s} \cdot \sum_{\ell=1}^{r} \frac{1}{(n_{r-\ell})^{1/2}} \\
&\qquad\qquad \text{(by value of } \mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) \text{ from the theorem statement)} \\
&\geq \frac{15}{16} - r \cdot (\frac{1}{n_0}) - 15 \cdot \sqrt{s} \cdot r \cdot (\frac{1}{n_0^{1/2}}) \qquad \text{(as } n_0, n_1, \ldots, n_{r-1} \text{ are increasing)} \\
&\geq \frac{15}{16} - \frac{1}{32} - 15 \cdot \sqrt{s} \cdot r \cdot (\frac{1}{n_0^{1/2}}) \qquad \text{(as } r < 32 n_0 \text{ from Eq (7.7))} \\
&\geq \frac{29}{32} - 15 \cdot \frac{1}{480} \cdot r \cdot (1/n_0^{-0.25}) \qquad \text{(by value of } s, \text{ and } n_0 = n_r^{1/34^r})
\end{aligned}$$

$$> \frac{29}{32} - \frac{1}{32} > \frac{7}{8}. \qquad \text{(as } r < {n_0}^{1/4} \text{ from Eq (7.7))}$$

This contradicts Claim 7.4.2, thereby proving the theorem. ∎

7.5 Round Elimination: Description

In this section, we give the desired protocol π_{r-1} in Lemma 7.4.16 for solving instances $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$, and provides parts of the analysis of the protocol. This protocol constructs an instance $G_r \sim \mathcal{G}_r(n_r)$ and embeds its input G_{r-1} into it. It also samples the first round messages that protocol π_r sends using the different sources of randomness in our model (see Section 7.2.2).

We want to sample the input and messages from $\widetilde{\mathcal{D}^{\text{real}}}$. As we have established in Claim 7.4.15, this distribution is close to the correct distribution $\mathcal{D}^{\text{real}}$. However, sampling from $\widetilde{\mathcal{D}^{\text{real}}}$ is not feasible either.

We sample from a distribution we label as $\mathcal{D}^{\text{fake}}$, and show that this is close in total variation distance to $\widetilde{\mathcal{D}^{\text{real}}}$ and thus $\mathcal{D}^{\text{real}}$. The description of π_{r-1} is given in steps where the random variables in $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$ are sampled progressively.

7.5.1 Key Steps of the Protocol

In this subsection, we describe some prominent steps of the protocol π_{r-1}.

7.5.1.1 Public Random Variables

Given a graph $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$, it is easy to sample new identites for each of the vertices using public randomness. It is also easy to sample the auxiliary random variables. This is the first step of the protocol. To break the correlation between messages that each inner vertex sends, we also sample some messages publicly.

Protocol 1. Step (1) of protocol π_{r-1} for $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ given π_r for $G_r \sim \mathcal{G}_r(n_r)$:

(1) Sample the random variables ids and also, $\boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}$, and $\boldsymbol{L}_{\text{all}}$ which jointly form aux using public randomness.

(2) For $i \in [n_{r-1}]$, the vertex with identity x_i in G_{r-1} takes on the identity of $x_i^* \in X^*$ in random variable ids. It changes the identities of all its neighbors in G_{r-1} correspondingly.

(3) For each inner vertex x and layer Y with $x \notin Y$, set $\mathcal{N}^{x \to Y}[L_{t,i}^{x \to Y}]$ to all be type t for $t \in [r] \cup \{0\}$ and $i \in [n_{r-1}]$.

(4) For each inner vertex x, and layer Y with $x \notin Y$, sample the message that x sends to vertex $y_j \in L_{t,i}^{x \to Y} \cap \{< y_i^*\}$, for all types $t \in [r] \cup \{0\}$, and $i \in [n_{r-1}]$, with public randomness.

We use $\mathcal{N}_{\text{pub}}^x$ and $\mathcal{M}_{\text{pub}}^x$ to denote the input and messages sent by each inner vertex x that are sampled publicly in this protocol.

Observation 7.5.1. *The total number of messages sampled in $\mathcal{M}^x_{\text{pub}}$ for each inner vertex x is at most $2 \cdot \gamma_r \cdot (r+1) \cdot n_{r-1}$.*

Proof. The size of set $L^{x \to Y}_{t,i}$ is exactly γ_r for all $i \in [n_{r-1}]$ and type $t \in [r] \cup \{0\}$, and other layer Y with $x \notin Y$. At most γ_r many messages are sampled for each of the $r+1$ types and n_{r-1} values of i, and the two choices of Y. ∎

We do not have the complete description of $\mathcal{D}^{\text{fake}}$ yet, this will be clear as we formalize more of protocol π_{r-1}. But we can prove that what we have sampled so far is the same way as in $\widetilde{\mathcal{D}^{\text{real}}}$.

Observation 7.5.2. *The distribution of $\mathsf{G}_{r-1}, \mathsf{ids}$ and aux are the same in $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$.*

Proof. Random variable G_{r-1} is the same in $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$, as it is just sampled from $\mathcal{G}_{r-1}(n_{r-1})$.

In step (1) of Protocol 1, the random variables ids and aux are jointly sampled independently of G_{r-1}. However, even in $\widetilde{\mathcal{D}^{\text{real}}}$, G_{r-1} is independent of $\mathsf{ids}, \mathsf{aux}$ by Observation 7.4.12-*(iii)*. ∎

7.5.1.2 Sampling Messages Between Inner Vertices

We now use pair randomness to sample messages between inner vertices.

In the description of $\mathcal{G}_r(n_r)$ from Distribution 3, the type of each pair (x, y) in G_{r-1} is unchanged. The types of all these vertices is upper bounded by r in graph $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$. Hence, all pairs $(x, y) \in G_{r-1}$ are present in $\mathcal{C}_r$ of instance G from $\mathcal{G}_r(n_r)$, and they can send messages to each other in the first round of π_r. We describe how to sample these messages now.

Protocol 2. Step (2) of protocol π_{r-1} for $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ given π_r for $G_r \sim \mathcal{G}_r(n_r)$:

For each pair $x, y \in G_{r-1}$, with $y \notin X$, sample the message that x sends to y in the first round conditioned on ids, $type(x, y), aux, \mathcal{N}^x_{\text{pub}}$ and $\mathcal{M}^x_{\text{pub}}$ using pair randomness between x and y.

We use $\mathcal{M}^x_{\text{in}}$ to denote all the messages that inner vertex x sends to some other inner vertex $y \in G_{r-1}$ and $y \notin X$. We use $\mathcal{M}^x_{\text{in}}(y)$ to denote the message that x sends to the inner vertex $y \in G_{r-1}$, and $\mathcal{M}^x_{\text{in}}(-y)$ to denote all the messages that x sends to other inner vertices that are not y. These messages are all sampled by this protocol.

7.5.1.3 Sampling Rest of the Input and Messages

We can now sample the rest of the random variables in distribution $\widetilde{\mathcal{D}^{\text{real}}}$.

Protocol 3. Step (3) of protocol π_{r-1} for $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ given π_r for $G_r \sim \mathcal{G}_r(n_r)$:

Each inner vertex $x_i \in G_{r-1}$ does the following:

(a) Sample the rest of its input from distribution $\widetilde{\mathcal{D}^{\text{real}}}$ and the remaining messages that x_i sends to its neighbors with private randomness conditioned on *ids*, *aux*, $\mathcal{N}^{x_i}_{\text{pub}}$, $\mathcal{M}^{x_i}_{\text{pub}}$, $\mathcal{M}^{x_i}_{\text{in}}$, and its inner input $\mathcal{N}^{x_i}_{\text{in}}$.

(b) Sample all the messages that x_i receives from outer vertices, conditioned on the entire input of x_i, *ids* and *aux*.

We use $\mathcal{N}^{x_i}_{\text{rest}}$ to denote all the random variables that x_i samples in step (3) of Protocol 3 using private randomness. This is comprised of the rest of the input of x_i, the messages it sends to outer neighbors, and all the messages it receives from its outer neighbors.

We can prove that the random variables $\mathcal{N}^{x_i}_{\text{rest}}$ for each inner vertex x_i are sampled from the correct distribution $\widetilde{\mathcal{D}^{\text{real}}}$.

Observation 7.5.3. *In $\mathcal{D}^{\text{fake}}$, conditioned on any choice of* ids, aux, G_{r-1}, $\boldsymbol{\mathcal{N}}^{x}_{\text{pub}}, \boldsymbol{\mathcal{M}}^{x}_{\text{pub}}$ *and* $\boldsymbol{\mathcal{M}}^{x}_{\text{in}}$ *for each inner vertex x,* $\boldsymbol{\mathcal{N}}^{y}_{\text{rest}}$ *for all inner vertices y are jointly sampled from $\widetilde{\mathcal{D}^{\text{real}}}$.*

Proof. In $\widetilde{\mathcal{D}^{\text{real}}}$, for each inner vertex x, its input, all the messages it sends, and the messages it receives from outer vertices are sampled independently of the other inner vertices and independently of G_{r-1} when conditioned on $\boldsymbol{\mathcal{N}}^{x}_{\text{in}}$, ids and aux by Observation 7.4.12-(*i*) and (*ii*).

Therefore, in step (3) of Protocol 3, when we sample the remaining input of each inner vertex x it is sufficient to condition on $\boldsymbol{\mathcal{N}}^{x}_{\text{in}}$, $\boldsymbol{\mathcal{M}}^{x}_{\text{in}}$, $\boldsymbol{\mathcal{M}}^{x}_{\text{pub}}$, $\boldsymbol{\mathcal{N}}^{x}_{\text{pub}}$, ids and aux. This is true of the remaining messages sent by x also.

From the definition of $\widetilde{\mathcal{D}^{\text{real}}}$ from Distribution 5, we know that all the messages that x receives from its outer neighbors are sampled only conditioned on the entire input of x, the value of *aux* and *ids*. This process is the same as in step (3) in Protocol 3. ∎

We have sampled all the random variables associated with $\widetilde{\mathcal{D}^{\text{real}}}$. However, the distribution of input and first round messages we get is different from $\widetilde{\mathcal{D}^{\text{real}}}$, even though parts of them may be the same from Observation 7.5.2 and Observation 7.5.3.

7.5.2 Full Protocol π_{r-1} and its Distribution

We finally put together the multiple parts of protocol π_{r-1}, and see the full description of $\mathcal{D}^{\text{fake}}$.

Protocol 4. Protocol π_{r-1} for $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ given π_r for $G_r \sim \mathcal{G}_r(n_r)$:

(1) The random variable G_{r-1} is given as input. Random variables $\mathsf{ids}, \mathsf{aux}$, and

$\boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ for all inner vertices x are sampled according to Protocol 1. Identities are changed appropriately as given in step (1) from Protocol 1.

(2) All messages between pairs of inner vertices $x, y \in G_{r-1}$ which have a channel in G_{r-1} are sampled as given in Protocol 2.

(3) The remaining input for each inner vertex x_i, the other messages it sends and all the messages it receives from outer vertices are sampled in Protocol 3.

(4) All the vertices continue to run π_r starting from the second round, assuming first round messages are as sampled, and output the same answer as π_r. See Claim 7.5.4 that specifies how to implement this step.

First, let us argue that the vertices can run the protocol π_r starting from the second round.

Claim 7.5.4. *Given any input graph G which lies in the support of $\mathcal{G}_r(n_r)$, all inner vertices can run protocol π_r correctly starting from second round based on the information they have access to.*

Proof. First, we argue that all the inner vertices have access to all the messages they send and receive in the first round. For any message sent by inner vertex x to another inner vertex y, we use pair randomness between x, y to sample it in step (2) from Protocol 2. Thus, both x, y have access to the message. All messages sent by any inner vertex x to any other inner vertex is known to x. All messages received by x sent by some other inner vertex y is known to x as well.

The remaining messages that x sends to outer vertices, and all the messages it receives from outer vertices are sampled with private randomness in step (3) from Protocol 3, and are all known to the inner vertex x.

When the input graph G lies in the support of $\mathcal{G}_r(n_r)$, the channel-degree of all outer vertices is at most one from Observation 7.4.3. Therefore, any inner vertex x with outer neighbor u can simulate the messages u sends to x, as it has access to the entire input of u.

As for the inner vertices, they have access to all their inputs. We have argued they also have access to all the first round messages sent and received by them at the end of step (3) of π_{r-1}. They can continue to run protocol π_r from the second round by simulating the behavior of all outer vertices connected to them. This proceeds exactly as in π_r till the last round, and the inner vertices can find the output of π_r (again, since they can fully simulate outer vertices). ∎

Next, let us talk about the distribution $\mathcal{D}^{\text{fake}}$, which is the joint distribution of the input and first round messages that we generate when running protocol π_{r-1}.

Distribution 6. Distribution $\mathcal{D}^{\text{fake}}$:

$$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}) \qquad \text{(the inner graph, identities and auxiliaries)}$$

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux} \Big)$$

(the public parts of input and messages from step (1) in Protocol 1)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \bigtimes_{\substack{y \in G_{r-1} \\ y \notin X}} (\boldsymbol{\mathcal{M}}^x_{\text{in}}(y) \mid \mathsf{type}(x,y), \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}) \Big)$$

(messages sent by inner vertices to each other in step (2) from Protocol 2)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \Big).$$

(rest of the input and messages for inner vertices in step (3) from Protocol 3)

Claim 7.5.5. *The distribution of the input and first round messages in π_{r-1} is exactly $\mathcal{D}^{\text{fake}}$ as defined in Distribution 6.*

Proof. We know that $\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}$ and $\boldsymbol{L}_{\text{all}}$ are sampled jointly with public randomness in π_{r-1}, independent of G_{r-1}. Therefore, their distributions are exactly as given in Distribution 6.

The random variables $\boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ are sampled with public randomness as well, but we know by Observation 7.4.12-(i),(ii) that they are independent of random variables associated with other inner vertices. Hence, they are sampled only conditioned on ids and aux. (There is no access to $\boldsymbol{\mathcal{N}}^x_{\text{in}}$ with public randomness.)

For each pair of inner vertices x, y, we sample $\boldsymbol{\mathcal{M}}^x_{\text{in}}(y)$ in step (2) of Protocol 2 conditioned on exactly the random variables stated in the claim: ids, aux from public randomness, $\mathsf{type}(x,y)$ that both inner vertices x, y have access to, and $\boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$, again from public randomness.

Finally, each inner vertex x samples the rest of its input, the remaining messages it sends and the messages it receives from outer vertices conditioned on $\mathsf{ids}, \mathsf{aux}$ from public randomness, $\boldsymbol{\mathcal{M}}^x_{\text{in}}$ that it sampled using the pair randomness to other inner vertices, $\boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ again from public randomness and $\boldsymbol{\mathcal{N}}^x_{\text{in}}$ it has from graph G_{r-1} in step (3) from Protocol 3. This is as stated in the description of $\mathcal{D}^{\text{fake}}$ in Distribution 6. ∎

We can see that $\mathcal{D}^{\text{fake}}$ and $\widetilde{\mathcal{D}^{\text{real}}}$ are not the same distributions. But, we can prove that they are close to each other in total variation distance. This is the focus of the next section.

Lemma 7.5.6 ("$\mathcal{D}^{\text{fake}}$ and $\mathcal{D}^{\text{real}}$ are not that different")**.**

$$\|\mathcal{D}^{\text{real}} - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \leq \frac{1}{(n_{r-1})} + 15\sqrt{\frac{s}{n_{r-1}}}.$$

Proof of Lemma 7.5.6 can be found in Section 7.6.2 and constitutes the main technical part of our argument. This concludes the description of the round elimination protocol.

7.5.3 Proof of Lemma 7.4.16

We can prove Lemma 7.4.16, restated below, using Lemma 7.5.6.

Lemma (Restatement of Lemma 7.4.16)**.** *For any $s, r \geq 1$ and $\delta \in (0,1)$, given a deterministic protocol π_r for instances $G_r \sim \mathcal{G}_r(n_r)$, with the following parameters:*

$$\mathsf{round}(\pi_r) = r \qquad \mathsf{bw}(\pi_r) = s \qquad \mathsf{suc}(\pi_r, \mathcal{G}_r(n_r)) \geq \delta,$$

we can construct a deterministic protocol π_{r-1} which takes instances $G_{r-1} \sim \mathcal{G}_{r-1}(n_{r-1})$ with the following parameters:

$$\mathsf{round}(\pi_{r-1}) = r-1 \qquad \mathsf{bw}(\pi_{r-1}) \leq s \qquad \mathsf{suc}(\pi_{r-1}, \mathcal{G}_{r-1}(n_{r-1})) \geq \delta - \frac{1}{n_{r-1}} - 15\sqrt{\frac{s}{n_{r-1}}}.$$

Proof of Lemma 7.4.16. We will prove that protocol π_{r-1} from Protocol 4 obeys the conditions in the statement of the lemma.

For the number of rounds in π_{r-1}, this is exactly $r-1$ as π_r has r rounds, and the first round messages are sampled without any communication between the vertices.

The total length of any message sent in the $r-1$ rounds of π_{r-1} is at most s, as all the messages sent by π_r are at most s bits.

We know by Observation 7.4.7 that for the graphs in the support of $\mathcal{G}_r(n_r)$, if protocol π_r outputs the correct answer, π_{r-1} is correct in detecting whether a triangle exists as well. The output of π_r and π_{r-1} are the same when inputs are sampled from $\mathcal{D}^{\text{real}}$, by Claim 7.5.4. Therefore, it is sufficient to lower bound the probability that π_r outputs the correct answer when the inputs and first round messages are sampled from $\mathcal{D}^{\text{fake}}$ as opposed to $\mathcal{D}^{\text{real}}$.

We use Fact 4.3.5 to lower bound the probability of success of π_{r-1} where $\mathcal{E}$ is the event that the answer on input G_{r-1} is correct:

$$\begin{aligned}
\Pr_{\mathcal{G}_{r-1}}[\pi_{r-1} \text{ is correct}] &= \Pr_{\mathcal{D}^{\text{fake}}}[\pi_r \text{ is correct}] && \text{(by the discussion above)} \\
&\geq \Pr_{\mathcal{D}^{\text{fake}}}[\pi_r \text{ is correct}] - \|\mathcal{D}^{\text{real}} - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \\
&\geq \delta - \left(\frac{1}{n_{r-1}} + 15\sqrt{\frac{s}{n_{r-1}}}\right),
\end{aligned}$$

where in the second inequality is by the assumption on the correctness probability of π_r and Lemma 7.5.6.

Finally, the protocol π_{r-1} is randomized, but it can be made deterministic by fixing the random string by the easy direction of Yao's minimax principle, without changing its probability of success, rounds, and bandwidth. The different sources of randomness do not cause an issue, as the randomness in all of these sources can be fixed to be a particular string. ∎

7.6 Round Elimination: Analysis

In this subsection, we analyze distributions $\mathcal{D}^{\text{fake}}$ (see Distribution 6) and $\mathcal{D}^{\text{real}}$ (see Distribution 5), and prove the upper bound on their total variation distance in Lemma 7.5.6. We already know that distribution $\mathcal{D}^{\text{real}}$ is close to $\widetilde{\mathcal{D}^{\text{real}}}$ by Claim 7.4.15. Hence, we focus on proving the distance between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$ is small. When we talk about messages in this section, we only refer to messages sent over the first round.

Our analysis is split into three parts:

Part 1. The messages sent by each inner vertex x are correlated with each other through the input of x. However, in step (2) of π_{r-1} from Protocol 2, they are sampled independently, only based on some parts of the input of x. We show that the correlation between these messages are low and thus can be sampled this way without too much loss.

Part 2. We publicly sample some messages sent by each inner vertex x for both layers Y, Z in step (1) of π_{r-1} from Protocol 1, independent of the input of x. We remove the correlation between these messages and the input of x from G_{r-1} to allow for this sampling.

Part 3. Messages sent by x to other inner vertices are sampled in step (2) of π_{r-1} from Protocol 2. These are sampled conditioned on some part of the input of x. We will show that the correlation between these messages and the rest of the input of x from G_{r-1} is low.

These are the only differences between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$. We have argued that the other random variables are sampled correctly in Observation 7.5.2 and Observation 7.5.3.

7.6.1 Setting up the Analysis

We move from $\widetilde{\mathcal{D}^{\text{real}}}$ to $\mathcal{D}^{\text{fake}}$ by way of multiple distributions (one for each part) which lie between them and use a hybrid argument. In this subsection, we describe these hybrid distributions.

First, let us cast distribution $\widetilde{\mathcal{D}^{\text{real}}}$ in terms of the random variables in $\mathcal{D}^{\text{fake}}$.

Claim 7.6.1. *Distribution $\widetilde{\mathcal{D}^{\text{real}}}$ can be written as:*

$$
\begin{aligned}
&\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}) && \text{(the inner graph, identities and auxiliaries)}\\
&\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}} \Big)\\
&\qquad \text{(the public parts of input and messages from step (1) in Protocol 1)}\\
&\times \Big(\bigtimes_{x \in G_{r-1}} (\boldsymbol{\mathcal{M}}^x_{\text{in}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}) \Big)\\
&\qquad \text{(messages sent by inner vertices to each other in step (2) from Protocol 2)}
\end{aligned}
$$

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \Big).$$

(rest of the input and messages for inner vertices in step (3) from Protocol 3)

Proof. It is evident that random variables $\mathsf{G}_{r-1}, \mathsf{ids}$ and aux are sampled the same way in the statement and in $\widetilde{\mathcal{D}^{\text{real}}}$.

From Observation 7.4.12-(i), we know that the inputs of all the inner vertices are independent of each other conditioned on $\mathsf{G}_{r-1}, \mathsf{ids}$ and aux. Moreover, we know from Observation 7.4.12-(ii) that the input of any inner vertex x only depends on the random variables $\boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}$ and aux in G_{r-1}. Therefore, the random variables $\boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ for each inner vertex in x are sampled only conditioned on $\mathsf{ids}, \mathsf{aux}$ and $\boldsymbol{\mathcal{N}}^x_{\text{in}}$.

As the protocol is determinstic, we know that the messages sent by any inner vertex x depend only on its input. Therefore, $\boldsymbol{\mathcal{M}}^x_{\text{in}}$ is also sampled from the right distribution in the statement of the claim. This argument applies to the rest of the input of x and the messages sent by x to outer vertices also.

Lastly, in distribution $\widetilde{\mathcal{D}^{\text{real}}}$, all the messages received by inner vertex x from outer vertices are sampled only based on the input of x, ids and aux, as is the case in our statement. ∎

To bound the distance between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$, we use weak chain rule of total variation distance from Fact 4.3.8 repeatedly. We define an ordering on the vertices and channels for this purpose.

Ordering vertices and channels. We use a lexicographic ordering on the inner vertices of the form $a_1, a_2, \ldots, a_{n_{r-1}}$, then $b_1, \ldots, b_{n_{r-1}}$ and lastly $c_1, \ldots, c_{n_{r-1}}$. For each inner vertex x, to order the channels that x has to inner vertices in the other two layers, we again use the lexicographic ordering we have over all the vertices. (The specific ordering is unimportant and we have picked the simplest one.)

We will now describe the hybrid distributions that we use as intermediate steps between the two distributions $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$.

Sampling Inner Messages Separately

The first hybrid distribution which lies between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$ is $\mathcal{H}_1$ that we define here. One key difference between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$ is in how the messages to other inner vertices sent by any inner vertex x are sampled. In $\widetilde{\mathcal{D}^{\text{real}}}$, they are sampled together, whereas this correlation is absent in $\mathcal{D}^{\text{fake}}$. We show that this correlation is small and can be broken.

Distribution 7. Distribution $\mathcal{H}_1$:

$$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}) \qquad \text{(the inner graph, identities and auxiliaries)}$$
$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}} \Big)$$

(the public parts of input and messages from step (1) in Protocol 1)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \bigtimes_{\substack{y \in G_{r-1} \\ y \notin X}} (\boldsymbol{\mathcal{M}}^x_{\text{in}}(y) \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \boldsymbol{\mathcal{N}}^x_{\text{in}})$$

(messages sent by inner vertices to each other in step (2) from Protocol 2)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}).$$

(rest of the input and messages for inner vertices in step (3) from Protocol 3)

We prove distributions $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$ are close to each other in Section 7.6.3.

Lemma 7.6.2 ("Messages sent by inner vertices can be sampled separately")**.**

$$\|\widetilde{\mathcal{D}^{\text{real}}} - \mathcal{H}_1\|_{\text{tvd}} \leq 6 \cdot (n_{r-1})^{5/2} \cdot \sqrt{\frac{s}{\gamma_r + 1}}.$$

Sampling Public Messages without Inner Inputs

The only difference between $\mathcal{H}_1$ and $\mathcal{D}^{\text{fake}}$, is that in some parts of $\mathcal{H}_1$ there is additional conditioning on some parts of the input $\boldsymbol{\mathcal{N}}^x_{\text{in}}$ of the inner vertices. We show that these conditionings can be slowly broken. To this end, we give our second hybrid distribution that lies between $\mathcal{H}_1$ and $\mathcal{D}^{\text{fake}}$, which samples public messages according to $\mathcal{D}^{\text{fake}}$.

Distribution 8. Distribution $\mathcal{H}_2$:

$$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}})$$ (the inner graph, identities and auxiliaries)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux})$$

(the public parts of input and messages from step (1) in Protocol 1)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \bigtimes_{\substack{y \in G_{r-1} \\ y \notin X}} (\boldsymbol{\mathcal{M}}^x_{\text{in}}(y) \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \boldsymbol{\mathcal{N}}^x_{\text{in}})$$

(messages sent by inner vertices to each other in step (2) from Protocol 2)

$$\times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}).$$

(rest of the input and messages for inner vertices in step (3) from Protocol 3)

The proof of the following lemma is similar to that of Lemma 7.6.2, and we skip it in this dissertation. The details can be found in the full version of the result [AS25].

Lemma 7.6.3 ("Low correlation between first round messages and inputs of inner vertices")**.**

$$\|\mathcal{H}_1 - \mathcal{H}_2\|_{\text{tvd}} \leq 3n_{r-1} \cdot \sqrt{\frac{s \cdot \gamma_r \cdot (r+1) \cdot n_{r-1}}{\alpha_r + 1}}.$$

Sampling Inner Messages with only Some Inner Inputs

We come to the last part of our analysis, which shows that distribution $\mathcal{H}_2$ is close to $\mathcal{D}^{\text{fake}}$. At this stage, the only difference between them is in how the messages between inner vertices are sampled. In $\mathcal{H}_2$, when sampling the message that $x_i \in G_{r-1}$ sends to y_j, we condition on the entire input of the inner vertex x_i, whereas in $\mathcal{D}^{\text{fake}}$, we only condition on the input that both inner vertices x_i and y_j are aware of. We show that these distributions are close together (see Distribution 6 for the distribution $\mathcal{D}^{\text{fake}}$).

Lemma 7.6.4 ("Low correlation between messages of inner vertices and parts of their inputs")**.**

$$\|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \leq 6(n_{r-1})^2 \cdot \sqrt{\frac{s}{2(\beta_r+1)}}.$$

Similarly, the proof of Lemma 7.6.4 is deferred to the full version of the result in [AS25].

7.6.2 Proof of Lemma 7.5.6

Proof of Lemma 7.5.6 now follows from Lemmas 7.6.2 to 7.6.4 with some minimal calculations using the values of α_r, β_r and γ_r from Eq (7.6).

Lemma (Restatement of Lemma 7.5.6)**.**

$$\|\mathcal{D}^{\text{real}} - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \leq \frac{1}{n_{r-1}} + 15\sqrt{\frac{s}{n_{r-1}}}.$$

Proof of Lemma 7.5.6. By Claim 7.4.15, we have

$$\|\mathcal{D}^{\text{real}} - \widetilde{\mathcal{D}^{\text{real}}}\|_{\text{tvd}} \leq \frac{1}{n_{r-1}}.$$

Thus, we can focus on bounding the distance between $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{D}^{\text{fake}}$ and use triangle inequality.

$$\begin{aligned}
\|\widetilde{\mathcal{D}^{\text{real}}} - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} &\leq \|\widetilde{\mathcal{D}^{\text{real}}} - \mathcal{H}_1\|_{\text{tvd}} + \|\mathcal{H}_1 - \mathcal{H}_2\|_{\text{tvd}} + \|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \\
&\qquad \text{(by triangle inequality)} \\
&\leq 6 \cdot \sqrt{\frac{s \cdot (n_{r-1})^5}{(n_{r-1})^6}} + \|\mathcal{H}_1 - \mathcal{H}_2\|_{\text{tvd}} + \|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \\
&\qquad \text{(by Lemma 7.6.2 and Eq (7.6))} \\
&\leq 6\sqrt{\frac{s}{n_{r-1}}} + 3n_{r-1} \cdot \sqrt{\frac{s \cdot \gamma_r \cdot (r+1) \cdot n_{r-1}}{\alpha_r + 1}} + \|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \\
&\qquad \text{(by Lemma 7.6.3)} \\
&\leq 6\sqrt{\frac{s}{n_{r-1}}} + 3 \cdot \sqrt{\frac{s \cdot (n_{r-1})^6 \cdot (r+1) \cdot (n_{r-1})^3}{2 \cdot (n_{r-1})^{11}}} + \|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}} \\
&\qquad \text{(by Eq (7.6))}
\end{aligned}$$

$$\leq 6\sqrt{\frac{s}{n_{r-1}}} + 3 \cdot \sqrt{\frac{s \cdot (n_{r-1})^6 \cdot (n_{r-1}) \cdot (n_{r-1})^3}{(n_{r-1})^{11}}} + \|\mathcal{H}_2 - \mathcal{D}^{\text{fake}}\|_{\text{tvd}}$$

(as $r+1 \leq n_{r-1}$ for large n_r)

$$\leq 9\sqrt{\frac{s}{n_{r-1}}} + 6(n_{r-1})^2 \cdot \sqrt{\frac{s}{2(\beta_r+1)}} \qquad \text{(by Lemma 7.6.4)}$$

$$\leq 9\sqrt{\frac{s}{n_{r-1}}} + 6 \cdot \sqrt{\frac{s \cdot (n_{r-1})^4}{2((n_{r-1})^5+1)}} \qquad \text{(by Eq (7.6))}$$

$$\leq 15\sqrt{\frac{s}{n_{r-1}}},$$

concluding the proof. ∎

The rest of this section contains the one part of the whole argument, namely, proving different steps of the round elimination protocol incur low losses.

7.6.3 Messages from One Vertex have Low Correlation

In this subsection, we prove Lemma 7.6.2. Let us recall the two distributions $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$ from Distribution 5 and Distribution 7 respectively.

Distribution $\widetilde{\mathcal{D}^{\text{real}}}$:

$$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}) \times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}} \Big) \times \Big(\bigtimes_{x \in G_{r-1}} (\boldsymbol{\mathcal{M}}^x_{\text{in}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}) \times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \Big).$$

Distribution $\mathcal{H}_1$:

$$\mathsf{G}_{r-1} \times (\mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}) \times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}} \Big) \times \Big(\bigtimes_{x \in G_{r-1}} \bigtimes_{\substack{y \in G_{r-1} \\ y \notin X}} (\boldsymbol{\mathcal{M}}^x_{\text{in}}(y) \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}) \times \Big(\bigtimes_{x \in G_{r-1}} \boldsymbol{\mathcal{N}}^x_{\text{rest}} \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}} \Big).$$

The difference is in the third line, in how the messages sent by x to other inner vertices in G_{r-1} are sampled. In $\widetilde{\mathcal{D}^{\text{real}}}$, they are jointly sampled conditioned on $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ and $\boldsymbol{\mathcal{N}}^x_{\text{in}}$. In $\mathcal{H}_1$, they are sampled independently of each other, conditioned on the same random variables $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$ and $\boldsymbol{\mathcal{N}}^x_{\text{in}}$.

Notation. We use $\mathcal{N}^{x_i}_{\text{in}}(y)$ to denote the $type(x_i, y)$ in G_{r-1} for inner vertex y in the input of x_i.

Claim 7.6.5. *For every inner vertex $x \in G_{r-1}$, and every other inner vertex $y_i \in Y$ for $i \in [n_{r-1}]$,*

$$\mathbb{I}(\boldsymbol{\mathcal{M}}^x_{\text{in}}(y_i) \,;\, \boldsymbol{\mathcal{M}}^x_{\text{in}}(-y_i) \mid \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}) \leq \frac{1}{\gamma_r + 1} \cdot 2n_{r-1} \cdot s.$$

Proof. By our notation, we have $\mathsf{id}(y_i) = y_i^*$. First, we have,

$$\begin{aligned}
&\mathbb{I}(\boldsymbol{\mathcal{M}}^x_{\text{in}}(y_i)\,;\boldsymbol{\mathcal{M}}^x_{\text{in}}(-y_i) \mid \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}})\\
&\mathbb{I}(\boldsymbol{\mathcal{M}}^x_{\text{in}}(y_i)\,;\boldsymbol{\mathcal{M}}^x_{\text{in}}(-y_i) \mid \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}, \mathsf{type}(x,y_i))\\
&\qquad\qquad\qquad

$$= \mathop{\mathbb{E}}_{y_j \in P} \mathbb{I}(\boldsymbol{\mathcal{M}}^{\vec{x}}[y_j]\,; \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathsf{id}(x), \boldsymbol{\mathcal{M}}^{\vec{x}}[P \cap \{< y_j\}], \mathsf{W}_{\text{rest}}, \mathsf{id}(y_i) = y_j, \mathcal{E}_{\text{cond}})$$

$$= \frac{1}{\gamma_r + 1} \cdot \sum_{y_j \in P} \mathbb{I}(\boldsymbol{\mathcal{M}}^{\vec{x}}[y_j]\,; \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathsf{id}(x), \boldsymbol{\mathcal{M}}^{\vec{x}}[P \cap \{< y_j\}], \mathsf{W}_{\text{rest}}, \mathsf{id}(y_i) = y_j, \mathcal{E}_{\text{cond}}).$$

(as $\mathsf{id}(y_i)$ is uniform over P)

Now, we argue that the event $\mathsf{id}(y_i) = y_j$ is independent of the joint distribution of all the other random variables in the mutual information term, conditioned on $\mathcal{E}_{\text{cond}}$. Let us list these random variables, and argue in steps (in each step, we condition on everything in the previous steps also).

- $\mathsf{id}(w)$ for $w \neq y_i$: these are disjoint from P and independent of the identity of y_i inside P.
- $\boldsymbol{\mathcal{N}}^{x \to Y}[P]$: this is deterministically fixed to all be t conditioned on $\mathcal{E}_{\text{cond}}$.
- $\boldsymbol{\mathcal{N}}^{x \to Y}$: the input of x to layer Y is already fixed to be type t for all elements of P. The rest of the input is independent of *which* of these choices are the actual identity of y_i.
- $\boldsymbol{\mathcal{N}}^{x \to Z}$: the input of x to layer Z is independent of which of the identities in P belong to y_i.
- $\boldsymbol{\mathcal{M}}^{x \to Y}, \boldsymbol{\mathcal{M}}^{x \to Z}$: we know the inputs of x are independent of the event $\mathsf{id}(y_i) = y_j$. As protocol π_r is deterministic, the messages sent by x are independent of this event also.
- $\boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}$: these are sets which are disjoint from P, and are also independent of $\mathsf{id}(y_i) = y_j$.
- $\boldsymbol{\mathcal{N}}^{x}_{\text{in}}$: this input to x comes from G_{r-1}, which is independent of the identity of y_i.
- $\boldsymbol{\mathcal{N}}^{x}_{\text{pub}}$ barring $\boldsymbol{\mathcal{N}}^{x \to Y}[P]$: this is comprised of inputs chosen for x on sets disjoint from P and are independent of which identity in P is given to y_i.
- $\boldsymbol{L}_{\text{all}}$ barring $\mathsf{L}^{x \to Y}_{t,i}$: these are all sets disjoint from P, independent of what happens inside P.
- W_{rest}: we have argued that all the random variables in W_{rest} are independent of P.

As such, the joint distribution of these random variables is independent of the event $\mathsf{id}(y_i) = y_j$ conditioned on $\mathcal{E}_{\text{cond}}$. Hence, we can continue as,

$$\frac{1}{\gamma_r + 1} \cdot \sum_{y_j \in P} \mathbb{I}(\boldsymbol{\mathcal{M}}^{\vec{x}}[y_j]\,; \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathsf{id}(x), \boldsymbol{\mathcal{M}}^{\vec{x}}[P \cap \{< y_j\}], \mathsf{W}_{\text{rest}}, \mathsf{id}(y_i) = y_j, \mathcal{E}_{\text{cond}})$$

$$= \frac{1}{\gamma_r + 1} \cdot \sum_{y_j \in P} \mathbb{I}(\boldsymbol{\mathcal{M}}^{\vec{x}}[y_j]\,; \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathsf{id}(x), \boldsymbol{\mathcal{M}}[P \cap \{< y_j\}], \mathsf{W}_{\text{rest}}, \mathcal{E}_{\text{cond}})$$

(as $\mathsf{id}(y_i) = y_j$ is independent of all random variables together conditioned on $\mathcal{E}_{\text{cond}}$)

$$= \frac{1}{\gamma_r + 1} \cdot \mathbb{I}(\boldsymbol{\mathcal{M}}^{\vec{x}}[P] \,;\, \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathsf{id}(x), \mathsf{W}_{\text{rest}}, \mathcal{E}_{\text{cond}})$$

(by the chain rule of mutual information in Fact 4.3.1-(7))

$$\leq \frac{1}{\gamma_r + 1} \cdot \mathbb{H}(\boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \mathcal{E}_{\text{cond}}) \qquad \text{(by Fact 4.3.1-(3))}$$

$$\leq \frac{1}{\gamma_r + 1} \cdot 2n_{r-1} \cdot s. \quad \text{(as } \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \text{ has } 2n_{r-1} - 1 \text{ messages of length at most } s \text{ sent by } x)$$

Putting things together, we have,

$$\begin{aligned}
&\mathbb{I}(\boldsymbol{\mathcal{M}}^{x}_{\text{in}}(y_i) \,;\, \boldsymbol{\mathcal{M}}^{x}_{\text{in}}(-y_i) \mid \boldsymbol{\mathcal{N}}^{x}_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^{x}_{\text{pub}}, \boldsymbol{\mathcal{M}}^{x}_{\text{pub}}) \\
&\qquad \leq \mathop{\mathbb{E}}_{\substack{\mathsf{type}(x,y_i) \\ =t}} \; \mathop{\mathbb{E}}_{\substack{\mathsf{L}^{x \to Y}_{t,i} \cup \{\mathsf{id}(y_i)\} \\ =P}} \left[\frac{1}{\gamma_r + 1} \cdot 2n_{r-1} \cdot s \right] \qquad \text{(by all the bounds above)} \\
&\qquad = \frac{1}{\gamma_r + 1} \cdot 2n_{r-1} \cdot s.
\end{aligned}$$

This completes the proof. ∎

We can now prove Lemma 7.6.2 using the weak chain rule of total variation distance. First, let us recall this chain rule from Fact 4.3.8. For any two distributions μ, ν on k random variables $\mathsf{w}^1, \mathsf{w}^2, \ldots, \mathsf{w}^k$, we have,

$$\|\mu - \nu\|_{\text{tvd}} \leq \sum_{i \in [k]} \mathop{\mathbb{E}}_{\mathsf{w}^{<i} \sim \mu} \|\mu(\mathsf{w}^i \mid \mathsf{w}^{<i}) - \nu(\mathsf{w}^i \mid \mathsf{w}^{<i})\|_{\text{tvd}}.$$

To bound the distance between $\widetilde{\mathcal{D}^{\text{real}}}$ from Distribution 5 and $\mathcal{H}_1$ from Distribution 7, we use the lexicographic ordering on all the vertices and inner channels.

We need to define some more random variables. These random variables are local to this subsection. They may be used for different purposes later.

- Variable $\mathsf{w}^{\text{start}}$: This is the joint random variable $\mathsf{G}_{r-1}, \mathsf{ids}, \boldsymbol{\mathcal{J}}_{\text{all}}, \boldsymbol{\mathcal{K}}_{\text{all}}, \boldsymbol{L}_{\text{all}}$ along with $\boldsymbol{\mathcal{N}}^{x}_{\text{pub}}, \boldsymbol{\mathcal{M}}^{x}_{\text{pub}}$ for each inner vertex x.
- Variables $\mathsf{w}^{x \to y}$ for inner vertices x, y in different layers : this is the random variable $\boldsymbol{\mathcal{M}}^{x}_{\text{in}}(y)$ for inner vertices x, y.
- Variables w^{x} for each inner vertex x: this is the joint random variable of $\mathsf{w}^{x \to y}$ for all inner vertices y which are not in the same layer as x. We use the lexicographic ordering defined earlier to order these random variables based on y.
- Variables w^{end}: Joint random variable $\boldsymbol{\mathcal{N}}^{x}_{\text{rest}}$ for each inner vertex x.

We need some simple observations.

Observation 7.6.6. *About random variables associated with* $\widetilde{\mathcal{D}^{\text{real}}}$ *and* $\mathcal{H}_1$*:*

(*i*) *The random variable* $\mathsf{w}^{\text{start}}$ *is distributed the same way in* $\widetilde{\mathcal{D}^{\text{real}}}$ *and* $\mathcal{H}_1$*.*

(*ii*) *In both* $\widetilde{\mathcal{D}^{\text{real}}}$ *and* $\mathcal{H}_1$, $\mathsf{w}^w \perp \mathsf{w}^{w'} \mid \mathsf{w}^{\text{start}}$ *for any distinct pair of inner vertices* $w, w' \in A \cup B \cup C$. *This is true regardless of whether* w, w' *are in the same layer or in different layers.*

(*iii*) *Conditioned on any choice of random variable* $\mathsf{w}^{\text{start}}$ *and* w^x *for all inner vertices* x, *distribution of* w^{end} *is the same in* $\widetilde{\mathcal{D}^{\text{real}}}$ *and* $\mathcal{H}_1$.

(*iv*) *In distribution* $\widetilde{\mathcal{D}^{\text{real}}}$ *and* $\mathcal{H}_1$, *for any inner vertex* x, $\mathsf{w}^x \perp \mathsf{w}^{\text{start}} \mid \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$.

(*v*) *In distribution* $\mathcal{H}_1$, *for any inner vertex* x, $\mathsf{w}^{x\to w} \perp \mathsf{w}^{x\to w'} \mid \boldsymbol{\mathcal{N}}^x_{\text{in}}, \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}$, *for any distinct pair of inner vertices* $w, w' \in Y \cup Z$ *of* G_{r-1}. *Again,* w *and* w' *may be in the same layer or different layers.*

Proof. Part (*i*) is apparent from the definition of distributions $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$, as all random variables in $\mathsf{w}^{\text{start}}$ are sampled the same way in $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$.

For part (*ii*), we know that when w^w is sampled in $\widetilde{\mathcal{D}^{\text{real}}}$, it is sampled conditioned on random variables $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^w_{\text{in}}$, $\boldsymbol{\mathcal{N}}^w_{\text{pub}}$, and $\boldsymbol{\mathcal{M}}^w_{\text{pub}}$ and ids. All these random variables are fixed by $\mathsf{w}^{\text{start}}$. Thus, w^w and $\mathsf{w}^{w'}$ are independent for two distinct inner vertices w, w' conditioned on $\mathsf{w}^{\text{start}}$ in distribution $\widetilde{\mathcal{D}^{\text{real}}}$.

Similarly in $\mathcal{H}_1$, random variable w^w is sampled only conditioned on $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^w_{\text{in}}, \boldsymbol{\mathcal{N}}^w_{\text{pub}}$, and $\boldsymbol{\mathcal{M}}^w_{\text{pub}}$, all of which are fixed by $\mathsf{w}^{\text{start}}$.

Part (*iii*) is clear, again from the definition of $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$ as $\boldsymbol{\mathcal{N}}^x_{\text{rest}}$ is sampled the same way in the two distributions for all inner vertices x for any choice of the other random variables.

Part (*iv*) is evident from how w^x are sampled. They are conditioned only on $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}}$, $\boldsymbol{\mathcal{N}}^x_{\text{pub}}$, and $\boldsymbol{\mathcal{M}}^x_{\text{pub}}$ and are independent of the other random variables in $\mathsf{w}^{\text{start}}$.

For part (*v*), in distribution $\mathcal{H}_1$, we know that when $\mathsf{w}^{x\to w} = \boldsymbol{\mathcal{M}}^x_{\text{in}}(w)$ is sampled, it is conditioned only on $\mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}$, and $\boldsymbol{\mathcal{M}}^x_{\text{pub}}$. It is independent of the messages sent by x to other inner vertices, conditioned on these random variables. ∎

First, we show that the distribution of w^x in $\widetilde{\mathcal{D}^{\text{real}}}$ and $\mathcal{H}_1$ are close to each other for all inner vertices x, conditioned on $\mathsf{w}^{\text{start}}$.

Claim 7.6.7. *For any inner vertex* x, *we have,*

$$\mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}}\sim\widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^x \mid \mathsf{w}^{\text{start}}) - \mathcal{H}_1(\mathsf{w}^x \mid \mathsf{w}^{\text{start}})\|_{\text{tvd}} \leq 2 \cdot (n_{r-1})^{3/2} \cdot \sqrt{\frac{s}{\gamma_r + 1}}.$$

Proof. Let $u_1, u_2, \ldots, u_{2n_{r-1}}$ be all the inner vertices not in the same layer as x, following the ordering we defined. We use w^j to denote the random variable $\mathsf{w}^{x\to u_j}$ for $j \in [2n_{r-1}]$. We use w^0 to denote $\mathsf{w}^{\text{start}}$. We use $\mathsf{w}^{<j}$ to denote the random variables $\mathsf{w}^0, \mathsf{w}^1, \ldots \mathsf{w}^{j-1}$. We have,

$$\mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}}\sim\widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^x \mid \mathsf{w}^{\text{start}}) - \mathcal{H}_1(\mathsf{w}^x \mid \mathsf{w}^{\text{start}})\|_{\text{tvd}}$$

$$\leq \sum_{j\in[2n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{<j}\sim\widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) - \mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j})\|_{\text{tvd}}$$

(by the weak chain rule of total variation distance Fact 4.3.8)

$$\leq \frac{1}{\sqrt{2}} \cdot \sum_{j\in[2n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{<j}\sim\widetilde{\mathcal{D}^{\text{real}}}} \sqrt{\mathbb{D}(\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) \mid\mid \mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j}))}$$

(by Pinsker's inequality Fact 4.3.10)

$$\leq \frac{1}{\sqrt{2}} \cdot \sum_{j\in[2n_{r-1}]} \sqrt{\mathop{\mathbb{E}}_{\mathsf{w}^{<j}\sim\widetilde{\mathcal{D}^{\text{real}}}} \mathbb{D}(\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) \mid\mid \mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j}))}.$$

(by Jensen's inequality and concavity of square root)

We bound the KL-divergence term separately for each $j \in [2n_{r-1}]$. Let us use $\boldsymbol{\mathcal{M}}^x_{\text{in}}(< u_j)$ to denote the random variables $\boldsymbol{\mathcal{M}}^x_{\text{in}}(u_1), \boldsymbol{\mathcal{M}}^x_{\text{in}}(u_2), \ldots, \boldsymbol{\mathcal{M}}^x_{\text{in}}(u_{j-1})$. We use W_{cond} to denote the random variables $\mathsf{aux}, \mathsf{ids}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}$, and $\boldsymbol{\mathcal{M}}^x_{\text{pub}}$ as they always appear in the conditioning.

We know by Observation 7.6.6-(iv) that the distribution of $\mathsf{w}^j \mid \mathsf{w}^{<j}$ in $\widetilde{\mathcal{D}^{\text{real}}}$ is,

$$\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) = \boldsymbol{\mathcal{M}}^x_{\text{in}}(u_j) \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}(< u_j), \mathsf{W}_{\text{cond}}.$$

We know by Observation 7.6.6-(v) and (iv) that the distribution of $\mathsf{w}^j \mid \mathsf{w}^{<j}$ in $\mathcal{H}_1$ is,

$$\mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j}) = \boldsymbol{\mathcal{M}}^x_{\text{in}}(u_j) \mid \mathsf{W}_{\text{cond}}.$$

Combining the above two equations gives us,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{w}^{<j}\sim\widetilde{\mathcal{D}^{\text{real}}}} \mathbb{D}(\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) \mid\mid \mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j})) \\
&\quad= \mathop{\mathbb{E}}_{\boldsymbol{\mathcal{M}}^x_{\text{in}}(<u_j),\mathsf{W}_{\text{cond}}} \mathbb{D}(\boldsymbol{\mathcal{M}}^x_{\text{in}}(u_j) \mid \boldsymbol{\mathcal{M}}^x_{\text{in}}(< u_j), \mathsf{W}_{\text{cond}} \mid\mid \boldsymbol{\mathcal{M}}^x_{\text{in}}(u_j) \mid \mathsf{W}_{\text{cond}}) \\
&\quad\leq \mathbb{I}(\boldsymbol{\mathcal{M}}^{x\to u_j}_{\text{in}}; \boldsymbol{\mathcal{M}}^x_{\text{in}}(< u_j) \mid \mathsf{W}_{\text{cond}}) && \text{(by Fact 4.3.4)} \\
&\quad\leq \mathbb{I}(\boldsymbol{\mathcal{M}}^{x\to u_j}_{\text{in}}; \boldsymbol{\mathcal{M}}^x_{\text{in}}(-u_j) \mid \mathsf{W}_{\text{cond}}) \\
&\qquad\qquad \text{(by Fact 4.3.1-(8), as } \boldsymbol{\mathcal{M}}^x_{\text{in}}(< u_j) \text{ is fixed by } \boldsymbol{\mathcal{M}}^x_{\text{in}}(-u_j)) \\
&\quad= \mathbb{I}(\boldsymbol{\mathcal{M}}^{x\to u_j}_{\text{in}}; \boldsymbol{\mathcal{M}}^x_{\text{in}}(-u_j) \mid \mathsf{ids}, \mathsf{aux}, \boldsymbol{\mathcal{N}}^x_{\text{in}}, \boldsymbol{\mathcal{N}}^x_{\text{pub}}, \boldsymbol{\mathcal{M}}^x_{\text{pub}}) && \text{(by definition of } \mathsf{W}_{\text{cond}}) \\
&\quad\leq s \cdot 2n_{r-1} \cdot 1/(\gamma_r + 1). && \text{(by Claim 7.6.5)}
\end{aligned}$$

Putting things together, we have,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}}\sim\widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^x \mid \mathsf{w}^{\text{start}}) - \mathcal{H}_1(\mathsf{w}^x \mid \mathsf{w}^{\text{start}})\|_{\text{tvd}} \\
&\quad\leq \frac{1}{\sqrt{2}} \cdot \sum_{j\in[2n_{r-1}]} \sqrt{\mathop{\mathbb{E}}_{\mathsf{w}^{<j}\sim\widetilde{\mathcal{D}^{\text{real}}}} \mathbb{D}(\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^j \mid \mathsf{w}^{<j}) \mid\mid \mathcal{H}_1(\mathsf{w}^j \mid \mathsf{w}^{<j}))} \\
&\qquad\qquad \text{(from our earlier bound on the total variation distance in the proof)} \\
&\quad\leq \frac{1}{\sqrt{2}} \cdot 2n_{r-1} \cdot \sqrt{s \cdot 2n_{r-1} \cdot 1/(\gamma_r + 1)} = 2 \cdot (n_{r-1})^{3/2} \cdot \sqrt{\frac{s}{\gamma_r + 1}}. \qquad \blacksquare
\end{aligned}$$

We prove Lemma 7.6.2, with another application of Fact 4.3.8.

Proof of Lemma 7.6.2. Let $u_1, u_2, \ldots, u_{3n_{r-1}}$ be the inner vertices with the lexicographic ordering. We use $\mathsf{w}^{u<\ell}$ to denote the joint random variable $\mathsf{w}^{u_1}, \mathsf{w}^{u_2}, \ldots, \mathsf{w}^{u_{\ell-1}}$ for $\ell \in [3n_{r-1}]$. We use w^{all} to denote all w^{u_ℓ} for $\ell \in [3n_{r-1}]$. Using Fact 4.3.8, we get,

$$
\begin{aligned}
&\|\widetilde{\mathcal{D}^{\text{real}}} - \mathcal{H}_1\|_{\text{tvd}} \\
&\quad \le \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{\text{start}}) - \mathcal{H}_1(\mathsf{w}^{\text{start}})\|_{\text{tvd}} + \sum_{\ell \in [3n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{u<\ell} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell}) - \mathcal{H}_1(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell})\|_{\text{tvd}} \\
&\qquad + \mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{\text{end}} \mid \mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}}) - \mathcal{H}_1(\mathsf{w}^{\text{end}} \mid \mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}})\|_{\text{tvd}} \\
&\quad = 0 + \sum_{\ell \in [3n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{u<\ell} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell}) - \mathcal{H}_1(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell})\|_{\text{tvd}} \\
&\qquad + \mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{\text{end}} \mid \mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}}) - \mathcal{H}_1(\mathsf{w}^{\text{end}} \mid \mathsf{w}^{\text{start}}, \mathsf{w}^{\text{all}})\|_{\text{tvd}} \\
&\qquad \text{(by Observation 7.6.6-}(i)\text{, we know } \widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{\text{start}}) \text{ and } \mathcal{H}_1(\mathsf{w}^{\text{start}}) \text{ are the same distribution)} \\
&\quad = \sum_{\ell \in [3n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{u<\ell} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell}) - \mathcal{H}_1(\mathsf{w}^{u_\ell} \mid \mathsf{w}^{u<\ell})\|_{\text{tvd}} + 0 \\
&\qquad \text{(by Observation 7.6.6-}(iii)\text{, distribution of } \mathsf{w}^{\text{end}} \text{ is the same)} \\
&\quad = \sum_{\ell \in [3n_{r-1}]} \mathop{\mathbb{E}}_{\mathsf{w}^{\text{start}} \sim \widetilde{\mathcal{D}^{\text{real}}}} \|\widetilde{\mathcal{D}^{\text{real}}}(\mathsf{w}^{x} \mid \mathsf{w}^{\text{start}}) - \mathcal{H}_1(\mathsf{w}^{x} \mid \mathsf{w}^{\text{start}})\|_{\text{tvd}} \\
&\qquad \text{(by Observation 7.6.6-}(ii)\text{)} \\
&\quad \le (3n_{r-1}) \cdot 2 \cdot (n_{r-1})^{3/2} \cdot \sqrt{\frac{s}{\gamma_r + 1}} \qquad \text{(by Claim 7.6.7)} \\
&\quad = 6 \cdot (n_{r-1})^{5/2} \cdot \sqrt{\frac{s}{\gamma_r + 1}}. \qquad \blacksquare
\end{aligned}
$$

Lemma 7.6.3 and Lemma 7.6.4 can be proved with the same framework, since we have described the distributions and the sampling process explicitly. This concludes the analysis of our round elimination protocol and the proof of Lemma 7.5.6, which in turn, as shown before, finalizes the entire proof of Lemma 7.4.16 and consequently Theorem 12.

Part III

Lower Bounds beyond Embedding

8 Chained Index

8.1 Introduction

We study a natural generalization of INDEX, called chained index ($\text{CHAIN}_{n,k}$ for $n, k \geq 1$) introduced by [CDK19]. There are k different instances of INDEX_n, correlated so that they have the same answer. They are "chained" together, where each player holds the index to the previous instance, and also the string for the next instance.

Definition 8.1.1 (Informal)**.** *In $\text{CHAIN}_{n,k}$, there are k instances of INDEX_n, all with the same answer. Players 1 and 2 take on the role of Alice and Bob respectively in the first instance, players 2 and 3 take on the role of Alice and Bob respectively for the second instance, and so on, all the way till players k and $k+1$ for the last instance.*

Communication is one-way from each player to the next in ascending order. The last player has to output the answer. The communication cost is the total number of bits in all the messages sent by the players. See Figure 8.1 for an illustration.

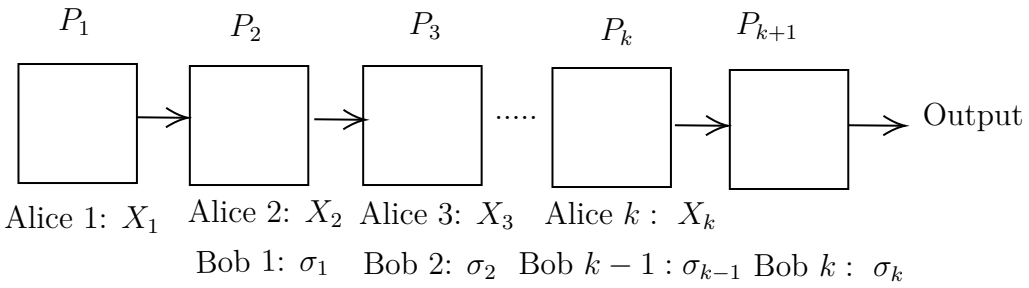


Figure 8.1: An illustration of the $\text{CHAIN}_{n,k}$ problem with k correlated sub-instances of INDEX_n from Definition 8.1.1. The arrows illustrate that the message is from P_i to P_{i+1} for $i \in [k]$.

In [CDK19], a reduction from CHAIN was employed to get a lower bound for approximation of maximum independent sets in vertex arrival streams. Before its introduction, [Kap12] used the problem implicitly to get a lower bound of $(1 - 1/e)$ in the approximation factor for maximum matching in $\tilde{O}(n)$ space in vertex arrival streams.

The problem has been used by the breakthrough result of [FNFSZ20] to study the multi-party communication complexity of submodular maximization. They proved that any randomized p-party protocol which maximizes a monotone submodular function $f : \{0,1\}^N \to \mathbb{R}$, subject to a cardinality constraint of at most p and an approximation factor of at least $(1/2 + \epsilon)$, uses $\Omega(N\epsilon/p^3)$ communication. This also gave a lower bound for streaming submodular maximization. The CHAIN problem was used by [FFSZ22] also for similar purposes, but subject to stronger matroid constraints. [BKO22] used a reduction from CHAIN to prove lower bounds for interval independent set selection in streams of split intervals.

We do not know tight bounds for the communication complexity of the CHAIN problem, despite finding varied applications of it. There is a trivial protocol of $O(n)$ bits, where any

player can send the entire string to the next player who holds the index. Another simple protocol is for each player to send $O(n/k)$ bits randomly sampled from their strings using public randomness, and with constant probability, in at least one of the k instances, we send the special bit to the player holding the index. However, this still takes $\Omega(n)$ total bits of communication.

In [CDK19], they prove a lower bound of $\Omega(n/k^2)$ for any $k \geq 1$ through a reduction from conservative multi-party pointer jumping problem, introduced by [DJS96]. They state without proof that a stronger lower bound of $\Omega(n/k)$ can be obtained for a restricted range of $k \leq O((n/\log n)^{1/4})$. They posed the following conjecture on the optimal communication lower bound.

Conjecture 8.1.2 ([CDK19])**.** *Any protocol that solves* $\textsc{chain}_{n,k}$ *requires* $\Omega(n)$ *bits of communication.*

[FNFSZ20] made some progress on Conjecture 8.1.2 by showing that among all the messages sent by the players, there is at least one message with $\Omega(n/k^2)$ bits for every $k \geq 1$. [MYHZ24] made further progress on Conjecture 8.1.2. They showed a lower bound of $\Omega(n/k + \sqrt{n})$ for oblivious protocols (where the length of the message sent by each player does not depend on the input), and a lower bound of $\Omega(n/k - k)$ for general protocols[1]. But the original conjecture is still open, and this is the focus of our work.

8.1.0.1 Our Results

We settle Conjecture 8.1.2 almost fully by proving the optimal lower bound of $\Omega(n)$ barring the corner case of when k is too large. As far as we know, this corner case is not a focus for existing reductions from $\textsc{chain}_{n,k}$.

Theorem 6 (restated). *For any* $n, k \geq 1$*, any protocol for* $\textsc{chain}_{n,k}$ *with probability of success at least 2/3, requires* $\Omega(n - k \log n)$ *total bits of communication.*

Therefore, as long as $k = o(n/\log n)$, we get the optimal $\Omega(n)$ lower bound from Theorem 6.

The proof of Theorem 6 can be found in Section 8.3. We prove the lower bound in the more general blackboard model of communication, instead of private messages between players (see Section 8.2.0.1 for details). The main idea is to analyze $\textsc{index}_n$ where Alice and Bob already have *some prior advantage* in guessing the answer. We elaborate on our techniques in Section 8.1.0.2.

As a direct corollary of Theorem 6, we get improvements in streaming lower bounds in [CDK19, FNFSZ20] through reductions from CHAIN immediately. In particular, we get that any algorithm which α-approximates the size of a maximum independent set in vertex arrival streams requires $\Omega(n^2/\alpha^5 - \log n)$ space, while the previous bound was $\Omega(n^2/\alpha^7)$ in [CDK19]. These applications can be found in the full version [Sun25].

[1]The authors of [MYHZ24] pointed out an important flaw in the arguments of an earlier version of this work which was posted at around the same time as [MYHZ24]. This flaw was subsequently fixed in the current version using a global change to the original argument, which now recovers optimal result for $k = o(n/\log n)$.

8.1.0.2 Our Techniques

In this subsection, we give an overview of the challenges in proving the lower bound and a summary of our techniques. We start by going over the prior techniques.

Prior Techniques. We will briefly talk about the technique used in [FNFSZ20] to prove that there is at least one player who sends $\Omega(n/k^2)$ bits for any $k \geq 1$. We will argue that these techniques can be extended to proving a lower bound of $\Omega(n/k)$ for the total number of bits, but not all the way to $\Omega(n)$ bits.

The first step in proving a lower bound for $\text{CHAIN}_{n,k}$ in [FNFSZ20] is a decorrelation step – the k instances of INDEX_n have the same answer, and they remove this correlation with a hybrid argument. These arguments have been used extensively in the literature (see e.g., [KKS15, AKSY20, KMT+22, AS23a]). Intuitively, any protocol that tries to solve $\text{CHAIN}_{n,k}$ may attempt to solve "many" of the instances of INDEX_n, albeit each with a "small" advantage over $1/2$, in the hope that the "small" advantages may accrue to get a constant probability of success overall (taking advantage of the fact that all the instances of INDEX_n have the same answer). The hybrid argument is a way to reduce proving a lower bound on the overall problem, to proving a lower bound for k different INDEX_n problems against these low advantages.

Let us assume, for simplicity, that the protocol tries to get an advantage of $\Omega(1/k)$ in each instance of INDEX_n, to get a constant total advantage. We can prove that any protocol that gets an advantage of $\Omega(1/k)$ for INDEX_n uses $\Omega(n/k^2)$ bits of communication using basic tools from information theory [CCKM13] (this is quite standard, see e.g., [Ass20] for a direct proof), and this is known to be tight. Therefore, for k instances, we get a lower bound of $\Omega(n/k)$ bits in total. Now, we will argue why this is not the optimal lower bound.

On one hand, for INDEX_n, it is known that for any $\delta \in (0, 1/2)$, there is a protocol that uses $O(n\delta^2)$ bits of communication to get a probability of success $1/2 + \delta$. This means that INDEX_n can be solved with advantage $\Omega(1/k)$ in $O(n/k^2)$ bits; in other words, each "hybrid step" of the previous lower bound argument is optimal. So, then, why can we not get a good protocol for $\text{CHAIN}_{n,k}$ by running the protocol for INDEX_n with $\delta = 1/k$ on all k instances? This protocol would have $O(n/k)$ bits of communication in total. The reason is that the $1/k$ small advantages do not add up as we would like them to. To illustrate this, we will briefly talk about the protocol that gets δ advantage in $O(n\delta^2)$ bits for INDEX_n.

Protocol for INDEX_n. First, we will sketch a protocol for $\delta = 1/\sqrt{n}$ that uses $O(1)$ bits of communication. Let us imagine that the input X is chosen uniformly at random from $\{0,1\}^n$ and the index σ is chosen uniformly at random from $[n]$. Then, Alice finds the majority bit from her string X and sends it to Bob. Bob just outputs the bit sent by Alice. We know, from simple anti-concentration bounds on the binomial distribution, that the number of indices with the majority bit in X is at least $n/2 + c\sqrt{n}$ with constant probability for some appropriate constant c. The protocol succeeds if Bob holds the index σ to a majority bit, and this happens with probability at least $\frac{1}{2} + \frac{c}{\sqrt{n}}$.

The assumption that the input X is chosen uniformly at random from $\{0,1\}^n$ can be removed using public randomness. Alice and Bob collectively sample a random string $A \in \{0,1\}^n$, and Alice changes her input to $X \oplus A$ so that each bit is 0 or 1 with equal probability.

Similarly, the assumption that the index σ is chosen uniformly at random from $[n]$ can be removed by Alice and Bob sampling a random permutation of $[n]$. Alice permutes string X according to this permutation, and Bob changes his input to the index that the permutation maps σ to. This is termed as the self-reducibility property of INDEX.

We can also extend the protocol to any δ by partitioning the string X into $n\delta^2$ blocks at random using public randomness and sending the majority bit in each block. Bob knows the block that his input index σ belongs to as public randomness is used.

Challenges for $\text{CHAIN}_{n,k}$. If we use the protocol we described for $\text{CHAIN}_{n,k}$, we are left with k bits from each instance of INDEX_n, which may be the correct answer to the problem with probability $1/2 + \Theta(1/k)$, but, the variance of each of these bits is $1/4 - \Theta(1/k^2)$. We have a protocol for $\text{CHAIN}_{n,k}$, where, out of the k instances of INDEX_n, it finds the right answer for $\approx k/2 + \Theta(1)$ instances in expectation. However, the standard deviation of the number of right answers is $\approx \sqrt{k}/2$, which is enough to mask the $\Theta(1)$ improvement over $k/2$ we get in expectation. If we use a hybrid argument over the total variation distance, each of the smaller "hybrid steps" is optimal, whereas the overall lower bound is not optimal. We cannot achieve a lower bound stronger than $\Omega(n/k)$.

Our Solution. Instead of keeping track of progress in terms of advantage gained in guessing the answer, we directly keep track of the "information" the message reveals about the answer. Formally, this translates to the change in entropy of the answer, after each successive message. Initially, the players have no information about the answer, and it is uniform over $\{0, 1\}$ (the entropy is 1). Any protocol with a large enough probability of success must reduce the entropy of the answer by a large factor (see Fano's inequality in Proposition 4.3.9). We prove that after each message, the entropy is reduced only by an additive factor *linear in the length of the message*, by a reduction to the INDEX problem. This will give us a lower bound on the total length of the messages.

For INDEX_n, in any protocol with probability of success at least ε, the entropy of the answer conditioned on the message is at most $H_2(\varepsilon)$ where $H_2(x) = x\log(1/x) + (1 - x)\log(1/(1 - x))$ is the binary entropy function. In the standard version where the initial entropy is 1, the reduction in entropy is $1 - H_2(\varepsilon)$, and it is known that the protocol requires $\Omega(n(1 - H_2(\varepsilon)))$ communication [JRS09]. This is not sufficient for our application due to the following reason: after the message of $\mathcal{P}_1$, when $\mathcal{P}_2$ and $\mathcal{P}_3$ attempt to solve INDEX_n, they already have some prior advantage that the message of $\mathcal{P}_1$ gives them. Therefore, we need to analyze INDEX_n when the answer is not uniform over $\{0, 1\}$.

Biased Index. We define the biased index problem, parametrized by $\theta \in [-1/2, 1/2]$. Alice and Bob receive input $Y \in \{0, 1\}^n$ and $\rho \in [n]$ respectively, such that the value at position ρ in Y (denoted by $Y(\rho)$) is 1 with probability $1/2 + \theta$ and 0 otherwise. Alice sends a message M to Bob and Bob has to output $Y(\rho)$. The initial entropy of the answer is $H_2(1/2 + \theta)$. We prove that entropy of $Y(\rho)$ conditioned on M is smaller by at most $O((|M| + \log n)/n)$ compared to the initial $H_2(1/2 + \theta)$, which is **our main contribution** (see Lemma 8.3.6).

We can show that the entropy of input to Alice, i.e. the random string Y, in such a distribution is at least $\Omega(n \cdot H_2(1/2 + \theta))$. Hence, after a message of length s from Alice, the

entropy of string Y reduces to $\Omega(n \cdot H_2(1/2+\theta) - s)$. For a randomly chosen position in Y after conditioning on the message, the entropy is at least $\approx H_2(1/2+\theta) - s/n$, which gives a lower bound on s.

In the distribution given to Alice and Bob, however, ρ is *not* chosen uniformly at random, and in fact, is correlated with the distribution of Y. Such versions of index where the distributions of Alice and Bob are correlated have been studied before (see Sparse Indexing in Appendix A of [AKL16] and Section 3.3 of [Sag11]). This correlation is the main issue in analyzing biased index with information theoretic tools.

Our solution to this correlation issue is using the sampling approach in Appendix A of [AKL16]. Let us assume that $\theta \geq 0$ for simplicity. The case when $\theta < 0$ can be analyzed similarly. We frame the process of sampling (Y, ρ) in three steps:

(i) First, we sample a subset T of size $n/(1+2\theta)$ uniformly at random.

(ii) Randomness in string Y is restricted only to indices in T: $n/2$ indices from T are chosen uniformly at random and set to 1, all other indices are set to 0.

(iii) Index ρ is chosen uniformly at random from T.

It is easy to see that this process gives the same distribution for (Y, ρ) as in the premise of biased index. Conditioned on T, however, distribution of Y and ρ are independent. Hence, this conditioning breaks the correlation, and we can also show that the loss in entropy of Y due to conditioning on T is not significant enough to hinder us. This restriction of the random process then allows us to use standard information theoretic tools to analyze biased index (see Section 8.3.0.3 for more details).

Independent and Concurrent Work. Independently and concurrently of this work, [MYHZ24] made progress on Conjecture 8.1.2. They showed a lower bound of $\Omega(n/k + \sqrt{n})$ for oblivious protocols (where the length of the message sent by each player does not depend on the input), and a lower bound of $\Omega(n/k - k)$ for general protocols. We show a lower bound of $\Omega(n - k \log n)$ for all protocols.[2]

Quantitatively, our lower bounds are a factor of almost k stronger, and are optimal for $k = o(n/\log n)$; moreover, our lower bound also holds for the Augmented Chain problem of [DDK23]. In terms of techniques, however, the two works are entirely disjoint: their proof is based on a new method of analysis through min-entropy and we use information theoretic approaches.

8.2 Preliminaries

In this section, we will present the required notation and definitions for our proof.

[2]The authors of [MYHZ24] pointed out an important flaw in the arguments of an earlier version of this work which was posted at around the same time as [MYHZ24]. This flaw was subsequently fixed in the current version using a global change to the original argument, which now recovers optimal result for $k = o(n/\log n)$.

Notation. For any tuple $A = (A_1, A_2, \ldots, A_m)$ of m items, we use $A_{<i}$ to denote the tuple $(A_1, A_2, \ldots, A_{i-1})$ for all $i \in [m]$. We use sans-serif font to denote random variables. For any random variable A, we use $A \sim \mathsf{A}$ to denote any A sampled from the distribution of the random variable A.

For any string $X \in \{0,1\}^n$, we use $X(\sigma)$ to denote the bit at position σ in X for $\sigma \in [n]$. We use $X(< \sigma)$ to denote the string of $\sigma - 1$ bits preceding $X(\sigma)$ in X. We use $X(S)$ for any $S \subseteq [n]$ to denote the bits at positions in set S.

For any $x \in [0,1]$, we use $H_2 : [0,1] \to [0,1]$ to denote the binary entropy function.

$$H_2(x) = -x \log x - (1-x) \log(1-x).$$

We need the following standard approximation of binomial coefficients (see Lemma 7 of Chapter 10 in [MS77]).

Fact 8.2.1 (c.f. [MS77])**.** *For any $p \geq 1$ and any $q \in [p-1]$, we have,*

$$2^{p \cdot H_2(q/p)} \cdot \sqrt{\frac{p}{8\pi q(p-q)}} \leq \binom{p}{q} \leq 2^{p \cdot H_2(q/p)} \cdot \sqrt{\frac{p}{2\pi q(p-q)}}$$

8.2.0.1 Communication Complexity Model

We use the standard number-in-hand multi-party model of communication. Only the basic definitions are given in this subsection. More details can be found in textbooks on communication complexity [RY20, KN97].

For any $k \geq 1$, let f be a function from $\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_k$ to $\{0,1\}$. There are k players $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ where $\mathcal{P}_i$ gets an input $a_i \in \mathcal{A}_i$ for $i \in [k]$. There is a shared blackboard visible to all the players. The players have access to a shared tape of random bits, along with their own private randomness. In any protocol π for f, the players send a message to the blackboard in increasing order ($\mathcal{P}_1$ sends a message followed by $\mathcal{P}_2$, and so on till $\mathcal{P}_k$). The last player $\mathcal{P}_k$, after all the messages are sent, outputs a single bit denoted by $\pi(a_1, a_2, \ldots, a_k)$. Protocol π is said to solve f with probability of success at least $1 - \delta$ if, for all $i \in [k]$, for any choice of $a_i \in \mathcal{A}_i$, we have,

$$\Pr[\pi(a_1, a_2, \ldots, a_k) \neq f(a_1, a_2, \ldots, a_k)] \leq \delta.$$

The **communication cost** of a protocol π is defined as the worst case **total communication** of all the players on the blackboard at the end of the protocol.

Definition 8.2.2. *The* **randomized communication complexity** *of f, with probability of error δ, is defined as the minimum communication cost of any protocol which solves f with probability of success at least $1 - \delta$.*

This concludes our preliminaries section.

8.3 The Lower Bound

In this section, we will prove our lower bound of $\Omega(n)$ on the communication complexity of the $\textsc{chain}_{n,k}$ problem for $k = o(n/\log n)$. Let us formally define the $\textsc{chain}_{n,k}$ communication problem first.

Definition 8.3.1. *The $\textsc{chain}_{n,k}$ communication problem is defined as follows. Given $k+1$ players $\mathcal{P}_i$ for $i \in [k+1]$ where,*

- *$\mathcal{P}_i$ has a string $X_i \in \{0,1\}^n$ for each $i \in [k]$, and,*
- *$\mathcal{P}_i$ for $1 < i \leq k+1$ has an index $\sigma_{i-1} \in [n]$,*

such that,

$$X_i(\sigma_i) = z,$$

for some bit $z \in \{0,1\}$. The players have a blackboard visible to all the parties. For $i \in [k]$ in ascending order, $\mathcal{P}_i$ sends a single message M_i, after which the index σ_i is revealed to the blackboard at no cost. $\mathcal{P}_{k+1}$ has to output whether z is 0 or 1. Refer to Figure 8.2 for an illustration.

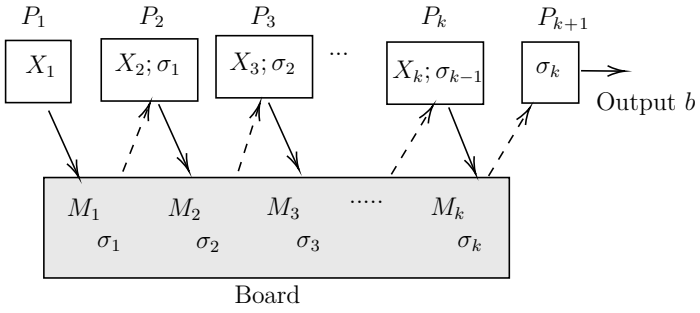


Figure 8.2: An illustration of the $\textsc{chain}_{n,k}$ problem from Definition 8.3.1. The solid arrows illustrate that player $\mathcal{P}_i$ writes a message M_i to the board. The dashed arrows indicate that $\mathcal{P}_i$ can read the contents of the board. It also shows the order in which the messages are sent by the players and indices are released.

Let us recall the statement of our main result.

Theorem 6 (restated). *For any $n, k \geq 1$, any protocol for $\textsc{chain}_{n,k}$ with probability of success at least 2/3, requires $\Omega(n - k\log n)$ total bits of communication.*

We give our hard distribution for $\textsc{chain}_{n,k}$ in Section 8.3.0.1 and give the proof of Theorem 6 in Section 8.3.0.2 except for the analysis of biased index, which is given in Section 8.3.0.3.

8.3.0.1 Setting Up the Problem

In this subsection, we start by defining the notation for our proof, and we describe the input distributions to $\textsc{chain}_{n,k}$.

The input hard distribution is as follows. Let $\mathcal{L} \subset \{0,1\}^n$ be the subset of strings where the number of ones is exactly equal to $n/2$.

Distribution $\mathcal{D}$ for $\textsc{chain}_{n,k}$:

1. Pick a bit z uniformly at random from $\{0,1\}$.
2. For each $i \in [k]$, sample (X_i, σ_i) uniformly at random from $\mathcal{L} \times [n]$ and independently conditioned on $X_i(\sigma_i) = z$.

Notation. We use X_i to denote the random variable corresponding to string X_i and $\boldsymbol{\sigma}_i$ to denote the random variable corresponding to index σ_i for $i \in [k]$. To denote the random variable corresponding to the first $i-1$ strings and indices, we use $\mathsf{X}_{<i}$ and $\boldsymbol{\sigma}_{<i}$ respectively.

We use M_i to denote the random variable corresponding to the message M_i sent by $\mathcal{P}_i$ to the blackboard. We use $M = (M_1, M_2, \ldots, M_k)$ to denote the tuple containing the messages of all the players, and M to denote the random variable corresponding to M.

Let π be a deterministic protocol for $\textsc{chain}_{n,k}$ with probability of success at least $2/3$ when the input is distributed according to $\mathcal{D}$. We use Γ to denote the random variable corresponding to the contents of the blackboard (referred to as a **transcript**), and we use γ to also denote transcripts sampled from Γ. We use Γ_i to denote the random variable of the tuple (M_i, σ_i) for $i \in [k]$. We use $\pi(\gamma_{<i}, X_i)$ to denote the output of $\mathcal{P}_i$ when the contents of the blackboard are $\gamma_{<i}$ and the input is X_i for $i \in [k]$.

Let s be the total length of all the messages sent by the players in γ. We assume that the total length of the messages is exactly s by padding. For any random variable A, we use $\mathcal{D}(\mathsf{A})$ to denote the distribution of the random variable A, as the input is distributed according to $\mathcal{D}$. We replace $\mathcal{D}(\mathsf{A} \mid \mathsf{B} = b)$ with $\mathcal{D}(\mathsf{A} \mid b)$ for ease of readability whenever it is clear from context.

Let Z denote the random variable corresponding to bit z. The bit z corresponds to the answer to $\textsc{chain}_{n,k}$. We show the lower bound of $\Omega(n - k \log n)$ for distinguishing between the case when $z = 0$ and $z = 1$ based on the contents of the blackboard.

We need one important observation about the distribution $\mathcal{D}$.

Observation 8.3.2. *For any $i \in [k]$, random variable $\mathsf{X}_i, \boldsymbol{\sigma}_i$ is independent of $\Gamma_{<i}$ conditioned on Z.*

Proof. For any fixed value of Z, $\mathsf{X}_i, \boldsymbol{\sigma}_i$ are chosen uniformly at random from $\mathcal{L} \times [n]$ such that $\mathsf{X}_i(\boldsymbol{\sigma}_i) = \mathsf{Z}$. This choice is independent of any $\mathsf{X}_j, \boldsymbol{\sigma}_j$ with $j \neq i$, and thus independent of $\Gamma_{<i}$. ∎

We are ready to proceed with the proof of our main theorem.

8.3.0.2 Proof of Lower Bound

We start by showing that in any successful protocol, the entropy of the distribution of Z conditioned on the message must be small.

Claim 8.3.3 (The transcript reveals information about Z.)**.**

$$\mathbb{H}(\mathsf{Z} \mid \Gamma) \leq 24/25.$$

Proof. We know that $\mathcal{P}_{k+1}$ successfully finds the value of z with probability of success at least $2/3$ using the transcript. Thus, using Fano's inequality in Proposition 4.3.9, we have,

$$\mathbb{H}(\mathsf{Z} \mid \Gamma) \leq H_2(2/3) \leq 24/25. \qquad \blacksquare$$

The main part of the proof is that we show a lower bound the entropy of Z conditioned on the transcripts using the entropy of the message.

Lemma 8.3.4. *For any protocol π,*

$$1 - \frac{12}{n} \cdot (\mathbb{H}(\mathsf{M}) + k \log n) \leq \mathbb{H}(\mathsf{Z} \mid \Gamma).$$

Before we prove Lemma 8.3.4, we can easily show that it implies Theorem 6.

Proof of Theorem 6. Combining Lemma 8.3.4 with Claim 8.3.3, we get,

$$1 - \frac{12}{n} \cdot (\mathbb{H}(\mathsf{M}) + k \log n) \leq \mathbb{H}(\mathsf{Z} \mid \Gamma) \leq \frac{24}{25}.$$

This gives that,

$$\mathbb{H}(\mathsf{M}) \geq \frac{n}{25 \cdot 12} - k \log n.$$

We know from Fact 4.3.1-(1) that $\mathbb{H}(\mathsf{M}) \leq \log(2^s) = s$, which proves that the total number of bits $s = \Omega(n - k \log n)$ for any deterministic protocol. By Yao's minimax principle, we get a lower bound of $\Omega(n - k \log n)$ on the randomized communication complexity of $\textsc{chain}_{n,k}$. $\blacksquare$

The proof of Lemma 8.3.4 employs a reduction to two player $\textsc{index}_n$. However, in these instances of $\textsc{index}_n$, Alice and Bob already have some partial information about the answer. We call this problem the **biased index** problem, and it is defined based on parameter $\theta \in [-1/2, 1/2]$, which is the initial bias known about the answer.

Definition 8.3.5. *The* **biased index distributional communication problem**, *denoted by* $\textsc{bias-ind}(\theta)$ *for $\theta \in [-1/2, 1/2]$, is defined as follows.*

Sample $W \in \{0, 1\}$ such that $W = 1$ with probability $1/2 + \theta$, and $W = 0$ otherwise. Sample (Y, ρ) uniformly at random from $\mathcal{L} \times [n]$ conditioned on $Y(\rho) = W$. Give string Y to Alice, and index ρ to Bob. Bob has to output $Y(\rho)$ after a single message $M_{\textsc{index}}$ from Alice.

Let $\pi_{\textsc{index}}$ be a deterministic protocol for $\textsc{bias-ind}(\theta)$. Let $\mathsf{W}, \mathsf{Y}, \boldsymbol{\rho}, \mathsf{M}_{\textsc{index}}$ denote the random variables corresponding to W, Y, ρ and $M_{\textsc{index}}$ respectively. Let $\mathcal{D}_\theta$ denote the joint distribution of $\mathsf{W}, \mathsf{Y}, \boldsymbol{\rho}$ and $\mathsf{M}_{\textsc{index}}$ in $\textsc{bias-ind}(\theta)$.

We prove the following lemma about $\textsc{bias-ind}(\theta)$ in Section 8.3.0.3.

Lemma 8.3.6 (Biased Index)**.** *For any protocol $\pi_{\textsc{index}}$ for $\textsc{bias-ind}(\theta)$,*

$$\mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\textsc{index}}, \boldsymbol{\rho}) \geq H_2(1/2 + \theta) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_{\textsc{index}}) + \log n).$$

Now we can prove Lemma 8.3.4.

Proof. We prove that, for any $0 \le \ell \le k$,

$$\mathbb{H}(\mathsf{Z} \mid \Gamma_{\le \ell}) \ge 1 - \frac{2}{n} \cdot (\sum_{j=1}^{\ell} \mathbb{H}(\mathsf{M}_j \mid \Gamma_{<j}) + \ell \log n),$$

by induction on ℓ. This implies the statement of the lemma because when $\ell = k$,

$$\begin{aligned}
\mathbb{H}(\mathsf{Z} \mid \Gamma) &= \mathbb{H}(\mathsf{Z} \mid \Gamma_{\le k}) \\
&\ge 1 - \frac{2}{n} \cdot (\sum_{j=1}^{k} \mathbb{H}(\mathsf{M}_j \mid \Gamma_{<j}) + k \log n) \\
&= 1 - \frac{2}{n} \cdot (\sum_{j=1}^{k} \mathbb{H}(\mathsf{M}_j \mid \mathsf{M}_{<j}, \boldsymbol{\sigma}_{<j}) + k \log n) \quad \text{(by defintion of random variable } \Gamma_{<j}) \\
&\ge 1 - \frac{2}{n} \cdot (\sum_{j=1}^{k} \mathbb{H}(\mathsf{M}_j \mid \mathsf{M}_{<j}) + k \log n) \\
&\qquad\qquad \text{(as conditioning reduces entropy, Fact 4.3.1-(4))} \\
&\ge 1 - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}) + k \log n). \qquad \text{(by chain rule of entropy, Fact 4.3.1-(6))}
\end{aligned}$$

The base case when $\ell = 0$ is trivial, as $\mathbb{H}(\mathsf{Z}) = 1$ when Z is uniform over $\{0, 1\}$, by Fact 4.3.1-(1). Let us assume that the statement is true for some $0 \le i - 1 < k$ for the induction hypothesis, and prove it for i.

For any $i \in [k]$ and any $\gamma_{<i}$ sampled from $\mathcal{D}(\Gamma_{<i})$, we define $\theta^{(\gamma_{<i})}$ to be the bias of distribution of Z conditioned on $\gamma_{<i}$ from uniform, that is,

$$\theta^{(\gamma_{<i})} = \Pr(Z = 1 \mid \Gamma_{<i} = \gamma_{<i}) - 1/2.$$

We also define distribution $\mathcal{D}^{(\gamma_{<i})}$ as the joint distribution of $\mathsf{Z}, \mathsf{X}_i, \boldsymbol{\sigma}_i$ and Γ_i conditioned on the event $\Gamma_{<i} = \gamma_{<i}$.

Now, we employ a reduction to BIAS-IND$(\theta^{(\gamma_{<i})})$ problem. Alice takes on the role of $\mathcal{P}_i$ and Bob takes on the role of $\mathcal{P}_{i+1}$ when the contents of the blackboard are fixed to be $\gamma_{<i}$. We will argue that distribution $\mathcal{D}^{(\gamma_{<i})}(\mathsf{Z}, \mathsf{X}_i, \boldsymbol{\sigma}_i, \mathsf{M}_i)$ of random variables in CHAIN$_{n,k}$ is identical to the distribution $\mathcal{D}_{\theta^{(\gamma_{<i})}}(\mathsf{W}, \mathsf{Y}, \boldsymbol{\rho}, \mathsf{M}_{\text{INDEX}})$ of the random variables in BIAS-IND$(\theta^{(\gamma_{<i})})$.

(*i*) Random variable Z is set to be 1 with probability $1/2 + \theta^{(\gamma_{<i})}$ and 0 otherwise, same as random variable W in $\mathcal{D}_{\theta^{(\gamma_{<i})}}$.

(*ii*) Random variable $\mathsf{Y}, \boldsymbol{\rho}$ is sampled from $\mathcal{L} \times [n]$ uniformly at random conditioned on $\mathsf{Y}(\boldsymbol{\rho}) = \mathsf{Z}$, and this is the same as the distribution of $\mathsf{X}_i, \boldsymbol{\sigma}_i$ in $\mathcal{D}^{(\gamma_{<i})}$. This is because $\mathsf{X}_i, \boldsymbol{\sigma}_i$ are independent of $\Gamma_{<i} = \gamma_{<i}$ conditioned on Z, by Observation 8.3.2.

(*iii*) As Alice takes on the role of $\mathcal{P}_i$, given the same input, she sends the same message as $\mathcal{P}_i$ when the contents of the blackboard are $\gamma_{<i}$. The distribution of M_i is the same as that of $\mathsf{M}_{\text{INDEX}}$, as distribution of $\mathsf{X}_i, \boldsymbol{\sigma}_i$ is the same as that of $\mathsf{Y}, \boldsymbol{\rho}$.

Hence using Lemma 8.3.6, we get,

$$\mathbb{H}(\mathsf{Z} \mid \Gamma_i, \Gamma_{<i} = \gamma_{<i}) \geq H_2(1/2 + \theta^{(\gamma_{<i})}) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_i \mid \Gamma_{<i} = \gamma_{<i}) + \log n). \tag{8.1}$$

We also observe that,

$$H_2(1/2 + \theta^{(\gamma_{<i})}) = \mathbb{H}(\mathsf{Z} \mid \Gamma_{<i} = \gamma_{<i}), \tag{8.2}$$

by definition of binary entropy.

Lastly, we have,

$$\begin{aligned}
\mathbb{H}(\mathsf{Z} \mid \Gamma_{\leq i}) &= \mathop{\mathbb{E}}_{\gamma_{<i} \sim \mathcal{D}(\Gamma_{<i})} \mathbb{H}(\mathsf{Z} \mid \Gamma_i, \Gamma_{<i} = \gamma_{<i}) && \text{(by the definition of entropy)} \\
&\geq \mathop{\mathbb{E}}_{\gamma_{<i} \sim \mathcal{D}(\Gamma_{<i})} \left[H_2(1/2 + \theta^{(\gamma_{<i})}) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_i \mid \Gamma_i = \gamma_{<i}) + \log n) \right] && \text{(by Eq (8.1))} \\
&= \mathop{\mathbb{E}}_{\gamma_{<i} \sim \mathcal{D}(\Gamma_{<i})} \left[\mathbb{H}(\mathsf{Z} \mid \Gamma_{<i} = \gamma_{<i}) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_i \mid \Gamma_i = \gamma_{<i}) + \log n) \right] && \text{(by Eq (8.2))} \\
&= \mathbb{H}(\mathsf{Z} \mid \Gamma_{<i}) - \frac{2}{n} \cdot \left(\mathop{\mathbb{E}}_{\gamma_{<i} \sim \mathcal{D}(\Gamma_{<i})} (\mathbb{H}(\mathsf{M}_i \mid \Gamma_{<i} = \gamma_{<i}) + \log n) \right) \\
&= \mathbb{H}(\mathsf{Z} \mid \Gamma_{<i}) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_i \mid \Gamma_{<i}) + \log n) \\
&\geq 1 - \frac{2}{n} \cdot \left(\sum_{j=1}^{i} \mathbb{H}(\mathsf{M}_j \mid \Gamma_{<j}) + i \log n \right), && \text{(by the induction hypothesis)}
\end{aligned}$$

finishing the proof. ∎

8.3.0.3 Biased Index

In this subsection, we will prove Lemma 8.3.6. Let us first recall the input distribution to BIAS-IND(θ).

Distribution $\mathcal{D}_\theta$: Sample $W = 1$ with probability $1/2 + \theta$ and set $W = 0$ otherwise. Sample $(Y, \rho) \in \mathcal{L} \times [n]$ conditioned on $Y(\rho) = W$.

In $\mathcal{D}_\theta$, the distribution of Y and $\boldsymbol{\rho}$ are highly correlated. We give an alternate way of sampling Y, ρ so that this correlation is removed partially.

Distribution $\mathcal{D}'_\theta$:
For $\theta \geq 0$:

(*i*) Sample set $T \subset [n]$ of size $b = n/(1 + 2\theta)$ uniformly at random.

(*ii*) Sample a set S of $n/2$ indices from T uniformly at random and set them to 1, and set $[n] \setminus S$ to 0 to get Y.

(*iii*) Sample ρ by sampling an index uniformly at random from T.

For $\theta < 0$:

(*i*) Sample set $T \subset [n]$ of size $b = n/(1-2\theta)$ uniformly at random.

(*ii*) Sample a set S of $n/2$ indices from T uniformly at random and set them to 0, and set $[n] \setminus S$ to 1 to get Y.

(*iii*) Sample ρ by sampling an index uniformly at random from T.

In this section, we assume that $\theta \geq 0$. For the case when $\theta < 0$, the proof follows in the same vein, and is not presented. Let t, S denote the random variables corresponding to set T and set S respectively. We will show that distributions $\mathcal{D}_\theta$ and $\mathcal{D}'_\theta$ are in fact identical.

Claim 8.3.7. *In distribution $\mathcal{D}_\theta$, for any $(Y, \rho) \in \mathcal{L} \times [n]$,*

$$\Pr(\mathsf{Y} = Y, \boldsymbol{\rho} = \rho) = \begin{cases} \frac{1+2\theta}{n \cdot \binom{n}{n/2}} & \text{when } Y(\rho) = 1, \\ \frac{1-2\theta}{n \cdot \binom{n}{n/2}} & \text{when } Y(\rho) = 0. \end{cases}$$

Proof. For any bit $c \in \{0, 1\}$, when $Y(\rho) = c$,

$$\begin{aligned} \Pr(\mathsf{Y} = Y, \boldsymbol{\rho} = \rho) &= \Pr(\mathsf{W} = c) \cdot \Pr((Y, \rho) \text{ is sampled conditioned on } \mathsf{W} = c) \\ &= \Pr(\mathsf{W} = c) \cdot \frac{2}{\binom{n}{n/2} \cdot n} \\ &\qquad \text{(as } Y, \rho \text{ are chosen uniformly at random with } Y(\rho) = c) \\ &= \frac{2 \cdot \Pr(\mathsf{W} = c)}{\binom{n}{n/2} \cdot n}. \end{aligned}$$

Substituting $\Pr(\mathsf{W} = 1)$ and $\Pr(\mathsf{W} = 0)$ as $1/2 + \theta$ and $1/2 - \theta$ respectively proves the claim. ∎

Claim 8.3.8. *Distribution $\mathcal{D}_\theta$ is the same as $\mathcal{D}'_\theta$.*

Proof. For a fixed Y, ρ we will show that the probability of sampling (Y, ρ) is same in both $\mathcal{D}_\theta$ and $\mathcal{D}'_\theta$. Let S' be the set of indices where the string Y takes value 1.

We begin with the case when $Y(\rho) = 1$ and $\rho \in S'$. In distribution $\mathcal{D}'_\theta$, for (Y, ρ) to be sampled, the random variable S must be fixed to S', and index ρ must be chosen from t.

Let us find the probability with which S is chosen to be S'. The distribution of S is uniform over all subsets of $[n]$ of size $n/2$, as t is uniform, and S is chosen uniformly from t. Therefore,

$$\Pr(\mathsf{S} = S') = \frac{1}{\binom{n}{n/2}}.$$

Now, we can find the probability that (Y, ρ) is sampled.

$$\begin{aligned}
\Pr((Y,\rho) \text{ is sampled from } \mathcal{D}'_\theta) &= \Pr(\mathsf{S} = S') \cdot \Pr(\rho \in S' \text{ is chosen from } \mathsf{t} \mid \mathsf{S} = S') \\
&= \frac{1}{\binom{n}{n/2}} \cdot \Pr(\rho \text{ is chosen from } \mathsf{t} \mid \rho \in \mathsf{t}) \\
&\qquad\qquad (\text{as } Y(\rho) = 1 \text{ and } \rho \in S' = \mathsf{S} \subset \mathsf{t}) \\
&= \frac{1}{\binom{n}{n/2}} \cdot \frac{1}{b} \qquad (\text{choosing one index among } b \text{ choices}) \\
&= \frac{(1+2\theta)}{\binom{n}{n/2} \cdot n}.
\end{aligned}$$

Now we look at the case when $Y(\rho) = 0$ and $\rho \notin S'$. Here, for (Y, ρ) to be sampled, we first need that $\mathsf{S} = S'$ and one index from the $b - n/2$ remaining indices of t must be ρ.

$$\begin{aligned}
\Pr(S' = \mathsf{S} \text{ and } \rho \in \mathsf{t}) &= \Pr(S' \cup \{\rho\} \subset \mathsf{t}) \cdot \Pr(S' = \mathsf{S} \mid S' \subset \mathsf{t}) \\
&= \binom{n/2 - 1}{b - n/2 - 1} \cdot \frac{1}{\binom{n}{b}} \cdot \Pr(S' = \mathsf{S} \mid S' \subset \mathsf{t}) \\
&\qquad (\text{as we choose remaining elements of } \mathsf{t} \text{ from } [n] \setminus (S' \cup \{\rho\})) \\
&= \binom{n/2 - 1}{b - n/2 - 1} \cdot \frac{1}{\binom{n}{b}} \cdot \frac{1}{\binom{b}{n/2}} \quad (\text{choosing one choice among all for } \mathsf{S}) \\
&= \frac{(b - n/2)}{n/2} \cdot \frac{(n/2)!}{(b-n/2)!(n-b)!} \cdot \frac{b!(n-b)!}{n!} \cdot \frac{(n/2)!(b-n/2)!}{b!} \\
&= (2b - n) \cdot \frac{1}{n \cdot \binom{n}{n/2}}
\end{aligned}$$

The final probability of sampling (Y, ρ) is,

$$\begin{aligned}
\Pr((Y,\rho) \text{ is sampled from } \mathcal{D}'_\theta) &= \Pr(S' = \mathsf{S} \text{ and } \rho \in \mathsf{t}) \cdot \Pr(\rho \text{ is chosen from } \mathsf{t} \mid \rho \in \mathsf{t}) \\
&= \frac{2b - n}{b} \cdot \frac{1}{n \cdot \binom{n}{n/2}} \\
&= \frac{1 - 2\theta}{n \cdot \binom{n}{n/2}}.
\end{aligned}$$

Using Claim 8.3.7 we complete the proof. ∎

Using this alternate way of sampling, it is easy to see that random variables Y and $\boldsymbol{\rho}$ are independent of each other conditioned on t. It can also be extended to include random variable $\mathsf{M}_{\text{INDEX}}$, as it is only a function of Y.

Observation 8.3.9. *In distribution $\mathcal{D}'_\theta$, conditioned on $\mathsf{t} = T$, for any $i \in T$, distribution of random variables $\mathsf{Y}, \mathsf{M}_{\text{INDEX}}$ is independent of event $\boldsymbol{\rho} = i$.*

Proof. Conditioned on $\mathsf{t} = T$, string Y is chosen by picking an $n/2$ size set S uniformly at random from T, and setting these indices to 1, and this choice also fixes $\mathsf{M}_{\text{INDEX}}$ as the protocol is deterministic. And index ρ is chosen uniformly at random from T, independently of the choice of S, by definition of $\mathcal{D}'_\theta$. Thus, choice of $\boldsymbol{\rho} = i$ is independent of random variables $\mathsf{Y}, \mathsf{M}_{\text{INDEX}}$. ∎

Next, we will show that even conditioned on t, the entropy of Y remains large.

Claim 8.3.10.

$$\mathbb{H}(\mathsf{Y} \mid \mathsf{t}) \geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) - 2\log n.$$

Proof. We assume that $\theta < 1/2$, as otherwise, the statement is vacuously true. $H_2(1) = 0$ by definition, and entropy is always non-negative by Fact 4.3.1-(1).

Conditioned on $\mathsf{t} = T$, we know that Y is fixed by choosing set S uniformly at random. Thus,

$$\begin{aligned}
\mathbb{H}(\mathsf{Y} \mid \mathsf{t}) &= \log\left(\binom{b}{n/2}\right) && \text{(by Fact 4.3.1-(1))}\\
&\geq \log\left(2^{bH_2(n/2b)} \cdot \sqrt{\frac{b}{8(n/2)(b-n/2)}}\right) && \text{(by Fact 8.2.1, and } n/2 < b, \text{ as } \theta < 1/2)\\
&= b \cdot H_2(n/2b) + \frac{1}{2} \cdot \log\left(\frac{b}{8(n/2)(b-n/2)}\right)\\
&= \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) + \frac{1}{2}\log\left(\frac{1}{4n \cdot (1/2-\theta)}\right)\\
&\geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) + \frac{1}{2}\log(1/4n) && \text{(as } 1/2-\theta \leq 1)\\
&\geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) - 2\log n. && ∎
\end{aligned}$$

We are ready to prove Lemma 8.3.6.

Proof of Lemma 8.3.6. We can lower bound the entropy of W conditioned on $\mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}$ as,

$$\begin{aligned}
\mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) &\geq \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}, \mathsf{t}) && \text{(as conditioning reduces entropy, Fact 4.3.1-(4))}\\
&= \mathop{\mathbb{E}}_{\mathsf{t}=T}\left[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho} = \rho, \mathsf{t} = T)\right] && \text{(as } \boldsymbol{\rho} \text{ is uniform over } T)\\
&= \mathop{\mathbb{E}}_{\mathsf{t}=T}\left[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{Y}(\rho) \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho} = \rho, \mathsf{t} = T)\right]\\
& && \text{(as } W = Y(\rho) \text{ by definition of } \mathcal{D}_\theta)\\
&= \mathop{\mathbb{E}}_{\mathsf{t}=T}\left[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{Y}(\rho) \mid \mathsf{M}_{\text{INDEX}}, \mathsf{t} = T)\right]\\
& && \text{(as } \mathsf{Y}, \mathsf{M}_{\text{INDEX}} \perp (\boldsymbol{\rho} = \rho) \mid \mathsf{t} = T, \text{ by Observation 8.3.9)}
\end{aligned}$$

$$
\begin{aligned}
&\geq \mathop{\mathbb{E}}_{\mathsf{t}=T}\left[\frac{1}{b}\cdot\mathbb{H}(\mathsf{Y}(T)\mid \mathsf{M}_{\text{INDEX}},\mathsf{t}=T)\right] \\
&\qquad\qquad\qquad

9 Streaming Shortest Paths

9.1 Introduction

We study the **single-source shortest path** problem in **undirected** (and possibly **weighted**) graphs in this model. Alongside the introduction of the semi-streaming model, [FKM+04] presented a single-pass semi-streaming algorithm for $O\left(\frac{\log n}{\log\log n}\right)$-approximation of single-source shortest paths, which was shown to be optimal soon after in [FKM+05]. This led [FKM+04] to pose the following open question for undirected shortest paths:

> [FKM+04]: *"Is there a multi-pass algorithm that improves the approximation ratio?"*

Since then, there has been a long line of research on this question using a wide range of techniques, primarily focusing on proving upper and lower bounds on the number of passes needed for obtaining a constant or $(1+\varepsilon)$-approximation [EZ04, GO16, HKN16, EN18, EN19, AR20, BFKL21, CKP+21a, ET22, AJJ+22, AS23a].

Upper bounds. The first of such algorithms appeared in the work of [EZ04] who obtained a semi-streaming $O_\varepsilon(n^\delta)$-pass algorithm for any small *constant* $\delta > 0$. This algorithm is based on the idea of *hybrid spanners* (see [EP04]). The next big improvements came from using *hopsets* to reduce the passes to $(1/\varepsilon)^{O(\sqrt{\log n \cdot \log\log n})}$ [HKN16, EN19], which is $n^{o(1)}$ for constant $\varepsilon > 0$. Finally, [BFKL21], using *continuous optimization* tools, reduced the number of passes to $\text{polylog}\,(n)/\varepsilon^2$, and [AJJ+22] further improved the ε-dependency in passes to $1/\varepsilon$; the dependency on $\log n$ in both these algorithms is some unspecified large polynomial (which appears to be at least $(\log n)^7$ in [BFKL21] and $(\log n)^{10}$ in [AJJ+22]).

Lower bounds. On the lower bound front, [FKM+05] first proved that finding $(1+\varepsilon)$-approximate shortest paths in the semi-streaming model requires $\Omega(1/\varepsilon)$ passes as long as $\varepsilon \geq (\frac{\log\log n}{\log n})$. This lower bound was extended to the (algorithmically easier) problem of $(1+\varepsilon)$-approximation of the shortest path *distance* (and not necessarily finding a path) between a fixed source and target in [GO16]. The space complexity in the latter result has been improved to $n^{2-o(1)}$ space in a series of work [AR20, CKP+21a, AS23a], although they do not imply new pass lower bounds over [GO16] for semi-streaming algorithms.

The current state-of-affairs for constant-factor approximation of shortest paths is thus the following:

- polylog(n)-pass upper bounds for arbitrarily small constant approximation ratios, and
- $\omega(1)$-pass lower bounds for sufficiently small constant approximation ratios.

9.1.0.1 Our Contributions

We make progress on this problem by establishing new upper and lower bounds.

Our upper bound holds for the general problem of computing $(1+\varepsilon)$-approximate single-source shortest paths on weighted graphs undergoing edge insertions and deletions. On the other hand, our lower bound holds for the simplest version of the problem where we only need to compute a large constant approximation of the distance between two fixed vertices in an unweighted graph (without having to find a shortest path). Our results can be stated more generally in terms of space-pass tradeoffs beyond the semi-streaming space restrictions. Our upper bound is as follows.

Result 2. *For any integer $k \geq 1$ and parameter $\varepsilon \in (0,1)$, there is a randomized streaming algorithm for weighted graphs that, with (exponentially) high probability, computes $(1+\varepsilon)$-approximate shortest paths from a given source to all other vertices in an n-vertex graph using*

$$O\left(\frac{1}{\varepsilon} \cdot kn^{1+1/k} \log n\right) \text{ space} \quad \text{and} \quad O\left(\frac{k^2}{\varepsilon}\right) \text{ passes.}$$

Moreover, this algorithm can be extended to dynamic streams that include both edge insertions and deletions. It can also be derandomized on insertion-only streams by increasing the space with some fixed $\text{poly}(\log(n), 1/\varepsilon)$ factors.

Plugging in $k = \log n / \log\log n$ to Result 2 implies a randomized semi-streaming algorithm for $(1+\varepsilon)$-approximate single-source shortest paths with

$$O\left(\frac{1}{\varepsilon} \cdot n \log^3 n\right) \text{ space} \quad \text{and} \quad O\left(\frac{1}{\varepsilon} \cdot \left(\frac{\log n}{\log\log n}\right)^2\right) \text{ passes.}^{1}$$

This improves upon prior algorithms for this problem by reducing the pass complexity from some unspecified polylogarithmic number of passes to only $\approx \log^2(n)$ passes (and keeping ε-dependency the same). Another benefit of this algorithm, which we will elaborate on further when we discuss our techniques, is that it is arguably much simpler than all existing approaches for this problem, especially the state-of-the-art methods that utilize convex optimization tools [BFKL21, AJJ+22].

For the most general version of the problem on dynamic streams, the current state-of-the-art algorithms (to the best of our knowledge) are due to [ET22]. These algorithms, operating in a similar space of $\widetilde{O}(n^{1+1/k})$, require $\left(\frac{k}{\varepsilon}\right)^k$ passes for unweighted graphs and $\left(\frac{k \cdot \log n}{\varepsilon}\right)^k$ passes for weighted graphs. Result 2 reduces this pass complexity to only $O(k^2/\varepsilon)$, even for weighted graphs.

Switching to lower bounds, we can prove that the pass complexity of Result 2 is within at most a quadratic factor of the optimal bounds for any constant-factor approximation of the problem.

[1] We measure space using $O(\log n)$-bit machine words.

Result 3. *For any $\alpha \geq 1$ and any integer k sufficiently smaller than $\log(n)/\alpha$, any streaming algorithm on unweighted n-vertex graphs that, with constant probability of success, can output an α-approximation to the distance between a fixed pair of vertices requires*

$$\Omega\left(\frac{1}{\text{poly}(k)} \cdot n^{1+\frac{1}{\alpha \cdot k}}\right) \text{ space} \quad \text{or} \quad \Omega(k) \text{ passes.}$$

Again, for semi-streaming algorithms in insertion-only streams and on unweighted graphs, our Result 3 implies that *any* constant approximation of shortest paths requires

$$\Omega\left(\frac{\log n}{\log \log n}\right) \text{ passes,}$$

as stated in Theorem 7. Previously, only single-pass lower bounds were known for approximation ratios beyond two [FKM+05], and only constant pass lower bounds for less-than-two approximation factors [GO16, AR20, CKP+21a] (even on weighted graphs, for finding all single-source shortest paths, and on dynamic streams).

Our results collectively show that the pass-complexity of (constant factor) approximation of single-source shortest paths in undirected graphs is between $\approx \log(n)$ to $\approx \log^2(n)$ passes. In this dissertation, we only focus on the lower bound.

9.1.0.2 Our Techniques

Although our proof is not very short, the ideas are simple, so we skip an elaborate overview. We briefly summarize the tools used in this subsection.

Our lower bound follows the work of [GO16] (itself inspired by [FKM+05]) by reducing the problem to *many* instances of the pointer chasing problem on n-vertex graphs. The key difference is in the analysis of the underlying pointer chasing instances. The lower bound of [GO16] works on a *product* distribution of inputs which inherently only corresponds to finding the shortest path distance to within an *additive* factor of two. On the other hand, we introduce and analyze a *correlated* input distribution that allows us to prove a lower bound for constant factor *multiplicative* approximation of shortest path distance. This part is the key technical contribution of our lower bound and is proven using a combination of probabilistic and combinatorial arguments to address the introduced correlation in the inputs.

Finally, similar to [GO16], we also extend this lower bound to "OR of many pointer chasing instances" using information complexity [CSWY01, BJKS02, BBCR10]. But, this part of our argument also differs almost entirely from [GO16] since our underlying hard distribution is no longer a product distribution. We implement this step by relying on the vast body of work on message compression and information complexity in a blackbox manner—instead of the adhoc arguments of [GO16]—and prove the lower bound even for two-player communication complexity.

9.2 Preliminaries

Notation. Given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ over the same vertex set V, we use $G_1 \cup G_2$ to denote the edge union of graphs G_1 and G_2, i.e., $G_1 \cup G_2 = (V, E_1, \cup E_2)$. Likewise, we define $G_1 \cap G_2$ as $(V, E_1 \cap E_2)$. Given a (possibly weighted) graph G containing vertices u and v, we use $\text{dist}_G(u, v)$ to denote the weighted shortest path distance between u and v in G.

9.2.0.1 Model of Communication

We work with the standard two party model of communication with players Alice and Bob. There is a function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ that Alice and Bob want to compute. Alice is give $X \in \mathcal{X}$ and Bob is given $Y \in \mathcal{Y}$. Alice and Bob have a shared tape of random bits they can access called public randomness. They also have their own private sources of randomness.

The communication happens in rounds, and there is one message sent by Alice or Bob to the other player in each round. The players alternate who sends the message over the rounds, and **Bob speaks first**. More specifically, Bob sends a message in the first round, and in all subsequent odd numbered rounds. Alice sends a message in all even numbered rounds. At the end of all communication, the player who receives the last message is required to output $f(X, Y)$.

A protocol π is said to succeed with probability p if, for all $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, the output of π is the same as $f(X, Y)$ with probability at least p. The communication cost of protocol π is the maximum length of any message sent over all the rounds in the protocol.

9.3 The Lower Bound

In this section, we prove that approximating the length of the shortest path between any two vertices s, t up to constant multiplicative factors requires $\Omega(\log n / \log \log n)$ passes in semi-streaming.

Theorem 13. *For any $\alpha \geq 1$ and any integer p sufficiently smaller than $\log(n)/\alpha$, any randomized streaming algorithm on unweighted graphs that can output an α-approximation to the distance between a fixed pair of vertices with probability at least $2/3$ requires*

$$\Omega\left(p^{-7} \cdot n^{1+\frac{1}{4\alpha\cdot(p+2)}}\right) \text{ space} \quad \text{or} \quad p \text{ passes.}$$

This directly proves Result 3 because for any constant $\alpha \geq 1$, for any $p = (1/\alpha) \cdot o(\log n / \log \log n)$, the lower bound on space is $\omega(n \cdot \text{poly} \log n)$, which is more than semi-streaming space.

We begin by stating the different steps of our lower bound proof. Initially, we are given a streaming algorithm $\mathcal{A}$ that estimates the shortest path distance between input vertices s, t in input graph G. We proceed as follows.

(*i*) We translate algorithm $\mathcal{A}$ into a protocol π_{OR} for the `OR-PPC` (see Definition 9.3.0.1) problem, which corresponds to the OR of many copies of the paired pointer chasing problem (`PPC`, Definition 9.3.0.1). This is covered in Lemma 9.3.5.

(*ii*) We use protocol π_{OR} to get a protocol π for solving the PPC problem with low communication using direct sum and message compression techniques, which is covered in Proposition 9.3.6.

(*iii*) Lastly, we prove a lower bound on the communication required to solve PPC instances on our hard distribution, stated in Lemma 9.3.7. This is the most novel part of our proof.

In this section, we first set up the graph constructions used in the lower bound. Then, we prove Theorem 13, assuming the lower bound for PPC. Lastly, we give our lower bound for PPC.

9.3.0.1 Layered Graphs and Paired Pointer Chasing

In this subsection, we develop our definitions of the paired pointer chasing problem (PPC) and the OR-PPC problem. We will conclude the section by completing Item (*i*) of our lower bound framework, stated in Lemma 9.3.5.

We begin by defining a graph family that we will use in our problem formulations.

Definition (Layered Graph)**.** *For integers $w, d \geq 1$, a layered graph is any graph $G = (V, E)$ that satisfies the following properties:*

(*i*) *The vertex set V can be partitioned into d equal size layers $V_1, V_2, \ldots, V_d$, each of size w. We let the set V_i be identified by $\{v_{i,1}, v_{i,2}, \ldots, v_{i,w}\}$.*

(*ii*) *The edge set E can be partitioned into $d-1$ matchings $M_1, M_2, \ldots, M_{d-1}$ such that for $i \in [d-1]$, M_i is a perfect matching between vertices in V_i and vertices in V_{i+1}.*

We use $\mathcal{L}_{d,w}$ to denote the set of all layered graphs with d layers and w nodes in each layer. For any $j \in [w]$, we use $\textsc{last}_G(v_{1,j})$ to denote the unique vertex $v_{d,\ell} \in V_d$ for some $\ell \in [w]$ that has a path to $v_{1,j} \in V_1$ in graph G.

We define a *product* operation on layered graphs that will be useful in our analysis in Section 9.3.0.3. Roughly speaking, the product operation creates a new graph out of two layered graphs by taking the Cartesian product of their vertices as the new vertex set and then defining edges of each product vertex based on the edges of each of its coordinates in the first two graphs.

Definition (Product of Layered Graphs)**.** *For any integers $w, d \geq 2$, given any two graphs $G^1 = (V^1, E^1), G^2 = (V^2, E^2) \in \mathcal{L}_{d,w}$, the product graph of G^1, G^2, denoted by $G^1 \times G^2 = (V^\times, E^\times)$ is defined as follows:*

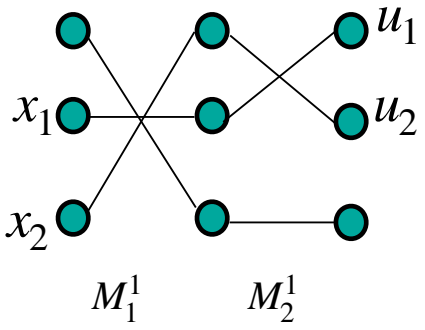
(*i*) *The vertex set $V^\times$ is $V_1^\times \sqcup V_2^\times \sqcup \ldots \sqcup V_d^\times$ with $V_i^\times = (V_i^1 \times V_i^2) = \{(x, y) \mid x \in V_i^1, y \in V_i^2\}$, where $V_1^1 \sqcup V_2^1 \sqcup \ldots \sqcup V_d^1$ is the (layered) partition of V^1 and $V_1^2 \sqcup V_2^2 \sqcup \ldots \sqcup V_d^2$ is for V^2.*

(*ii*) *The edge set $E^\times$ is comprised of $d-1$ matchings $M_1^\times, M_2^\times, \ldots, M_{d-1}^\times$ with*

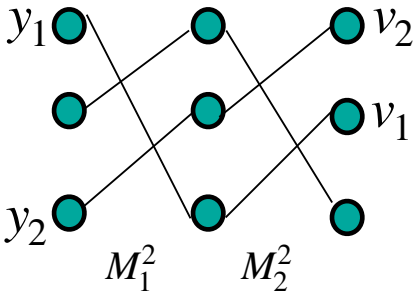
$$M_i^\times = \{((x_i, y_i), (x'_{i+1}, y'_{i+1})) \mid (x_i, x'_{i+1}) \in M_i^1, (y_i, y'_{i+1}) \in M_i^2\},$$

where M_i^1 and M_i^2 are the i^{th} matchings in G^1 and G^2 respectively, for $i \in [d-1]$.

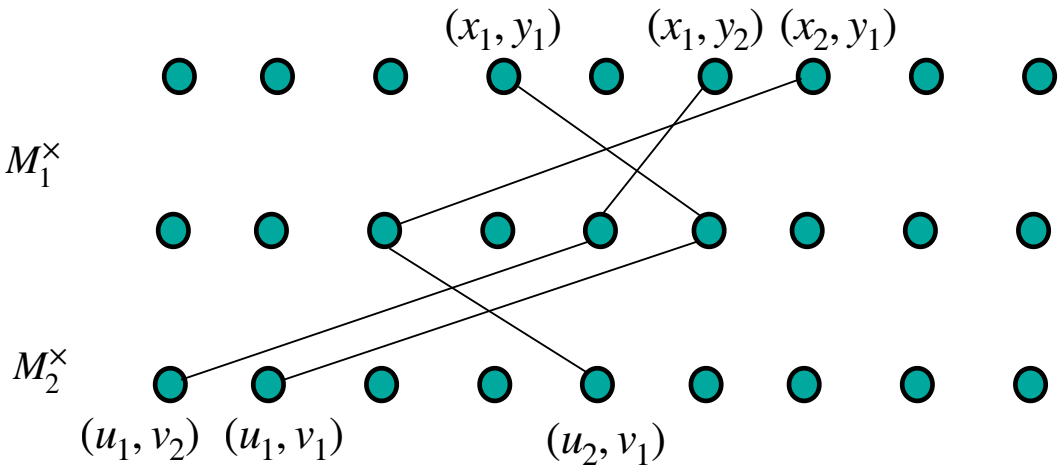
See Figure 9.1 for an illustration.



(a) Layered graph G^1.



(b) Layered graph G^2.



(c) Graph $G^\times = G^1 \times G^2$ from set $\mathcal{L}_{3,9}$. Also illustrates that $\text{LAST}_{G^\times}((x_1, y_1)) = (\text{LAST}_{G^1}(x_1), \text{LAST}_{G^2}(y_1)) = (u_1, v_1)$.

Figure 9.1: Product of two layered graphs from $\mathcal{L}_{3,3}$ lying in $\mathcal{L}_{3,9}$. Some edges are omitted.

We will make use of the following simple observations about products of layered graphs.

Observation 9.3.1. *For any $G^1, G^2 \in \mathcal{L}_{d,w}$, and graph $G = G^1 \times G^2$, we have,*

(i) The graph G belongs to $\mathcal{L}_{d,w^2}$.

(ii) For any $(x, y) \in V_1^\times$, $\textsc{last}_G((x, y)) = (\textsc{last}_{G^1}(x), \textsc{last}_{G^2}(y))$.

Proof. For part (i), we know that the number of vertices in each layer of $V^\times$ is w^2. For $i \in [d-1]$, each matching $M_i^\times$ in $E^\times$ takes any $(x, y) \in V_i^\times$ to a unique vertex in $V_{i+1}^\times$, and is a perfect matching. Part (ii) is also evident from the definition of the product operation. See Figure 9.1 for an illustration of these properties. ∎

We also define a *join* operation on layered graphs, which will be useful for establishing Item (*i*) of our lower bound framework in Lemma 9.3.5. The join operation takes two layered graphs, "reverses" the order of the second one, and connects the two graphs using a fixed perfect matching between their last layers. This reversal of the order of the second layered graph is necessary, as this operation allows us to check for equality between $\textsc{last}_{G^1}(v_{1,j})$ and $\textsc{last}_{G^2}(v_{1,j})$ in two different layered graphs G^1 and G^2 for any $j \in [w]$. See Observation 9.3.2 for details.

Definition (Join of Layered Graphs)**.** *For integers $w, d \geq 2$, given two graphs $G^1 = (V^1, E^1), G^2 = (V^2, E^2) \in \mathcal{L}_{d,w}$, the join graph of G^1, G^2, denoted by $G^1 + G^2 = (V^+, E^+)$ is defined as follows:*

(*i*) *The vertex set V^+ is a union of $2d$ many layers of size w each where,*

$$V_i^+ = \begin{cases} V_i^1 \text{ if } i \in [d], \\ V_{2d+1-i}^2 \text{ if } d < i \leq 2d, \end{cases}$$

where $V_1^1 \sqcup V_2^1 \sqcup \ldots \sqcup V_d^1$ is the (layered) partition of V^1 and $V_1^2 \sqcup V_2^2 \sqcup \ldots \sqcup V_d^2$ is for V^2.

(*ii*) *The edge set E^+ is,*

$$E^+ = E^1 \cup E^2 \cup \{(v_{d,j}^1, v_{d,j}^2) \mid j \in [w]\},$$

where $v_{d,j}^1$ is the j^{th} vertex in V_d^1 and $v_{d,j}^2$ is the j^{th} vertex in V_d^2.

See Figure 9.2 for an illustration.

We will need the following observations about the join operation.

Observation 9.3.2. *For any $G^1, G^2 \in \mathcal{L}_{d,w}$, and graph $G = G^1 + G^2$, we have,*

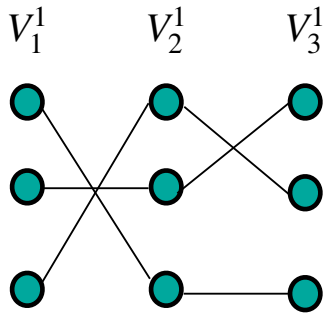
(*i*) *The graph G belongs to $\mathcal{L}_{2d,w}$.*

(*ii*) *For any $i \in [w]$, and vertex $v_{1,i} \in V_1^+$, $\textsc{last}_G(v_{1,i}) = v_{2d,i}$ if and only if $\textsc{last}_{G^1}(v_{1,i})$ and $\textsc{last}_{G^2}(v_{1,i})$ are equal.*

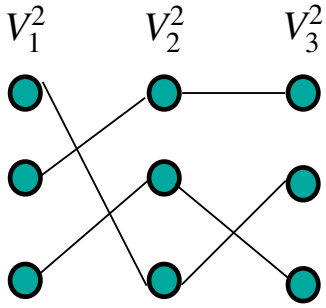
Proof. Part (*i*) follows directly from the join operation on two graphs $G^1, G^2 \in \mathcal{L}_{d,w}$.

We now verify part (*ii*). We observe that $\textsc{last}_{G^1}(v_{1,i})$ is the unique vertex in layer $V_d^1 = V_d^+$ connected to $v_{1,i}$ in G. This vertex is connected to $v_{1,i} \in V_1^2$ in G if and only if $\textsc{last}_{G^2}(v_{1,i})$ is the same as $\textsc{last}_{G^1}(v_{1,i})$. ∎

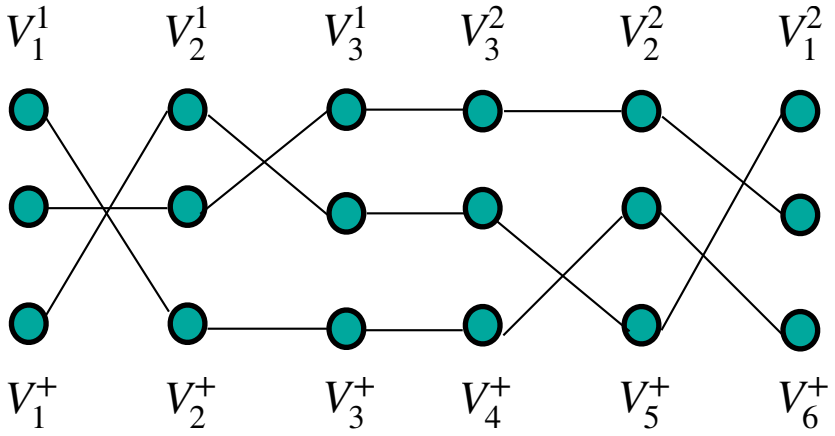
We are ready to define the paired pointer chasing problem now. In the original pointer chasing problem, we have a layered graph and a fixed starting vertex, and the goal is to determine the unique vertex in the last layer reachable from the starting point. In this new problem, we have two instances of the pointer chasing problem that are correlated with each other so that with probability half they both reach the same vertex in the last layer, and otherwise the two instances are independent.



(a) Layered graph G^1.



(b) Layered graph G^2.



(c) Graph $G^\times = G^1 + G^2$ from set $\mathcal{L}_{6,3}$. The name of the layers in G^1 and G^2 is specified above.

Figure 9.2: Join of two layered graphs from $\mathcal{L}_{3,3}$ lying in $\mathcal{L}_{6,3}$.

Definition (Paired Pointer Chasing)**.** *For integers $w, d \geq 2$, the paired pointer chasing problem, denoted by $\mathtt{PPC}_{d,w}$ is defined as follows.*

(*i*) *There is a bit $b \in \{0,1\}$ chosen uniformly at random.*

(*ii*) *The input is two graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ from $\mathcal{L}_{d,w}$. We use M_i^j to denote the i^{th} matching in graph G^j for $i \in [d-1]$ and $j \in \{1,2\}$.*

(*iii*) *Alice receives M_i^j for all odd $i \in [d-1]$ and Bob receives M_i^j for all even $i \in [d-1]$ for both $j \in \{1,2\}$.*

(*iv*) *When $b = 0$, all the matchings in graphs G^1 and G^2 are chosen uniformly at random.*

(*v*) *When $b = 1$, all the matchings in graphs G^1 and G^2 are chosen uniformly at random conditioned on $\textsc{last}_{G^1}(v_{1,1}) = \textsc{last}_{G^2}(v_{1,1})$.*

(*vi*) *The players have $d-2$ rounds to communicate. At the end of all $d-2$ rounds, the player who receives the last message has to output the value of bit b.*

We use $\mu_{\texttt{PPC}}$ to denote the distribution of the input in $\texttt{PPC}_{d,w}$ problem. We use $\mu^0_{\texttt{PPC}}, \mu^1_{\texttt{PPC}}$ to denote the distribution of the input conditioned on $b = 0$ and $b = 1$ respectively.

We can use a streaming algorithm that estimates shortest path lengths to check whether the OR of many instances of PPC evaluate to 1. This is a promise problem with t-many instances of PPC, distributed so that exactly one of these instances is drawn from $\mu^1_{\texttt{PPC}}$, or all of them are drawn from $\mu^0_{\texttt{PPC}}$. Notice that if all t-instances are sampled $\mu^0_{\texttt{PPC}}$, all matchings are chosen independently and uniformly, but this is not true even if one one PPC instance is sampled from $\mu^1_{\texttt{PPC}}$. The correlation between different matchings in PPC problem persists in OR-PPC also. Formally, we define the OR-PPC problem distributionally as follows.

Definition (OR of many PPC instances)**.** *For integers $t, w, d \geq 2$, the t-OR of PPC problem, denoted by $\texttt{OR-PPC}_{t,d,w}$ is defined as follows.*

(i) There is a bit $b \in \{0, 1\}$ chosen uniformly at random.

(ii) There are t instances of PPC, denoted by (G^1_i, G^2_i) for $i \in [t]$. The input matchings of these instances are given to Alice and Bob as in Definition 9.3.0.1.

(iii) When $b = 0$, (G^1_i, G^2_i) is sampled from $\mu^0_{\texttt{PPC}}$ for all $i \in [t]$.

(iv) When $b = 1$, $i^\star \in [t]$ is chosen uniformly at random. $(G^1_{i^\star}, G^2_{i^\star})$ is sampled from $\mu^1_{\texttt{PPC}}$, and (G^1_i, G^2_i) is sampled from $\mu^0_{\texttt{PPC}}$ for all $i \in [t]$ with $i \neq i^\star$.

(v) The players have $d - 2$ rounds to communicate. At the end of all $d - 2$ rounds, the player who receives the last message has to output the value of bit b.

Recall that Item *(i)* of our lower bound framework is to reduce solving the OR-PPC problem to computing a constant-factor approximation of the s-t shortest path distance in the streaming model. With that goal in mind, for each OR-PPC instance we define an associated graph whose shortest path distances can aid in finding the value of bit b.

More formally, consider an instance of OR-PPC corresponding to t instances of PPC, denoted by (G^1_i, G^2_i) for $i \in [t]$. Let $G_i = G^1_i + G^2_i$ for each $i \in [t]$. By Observation 9.3.2, graph G_i belongs to $\mathcal{L}_{2d,w}$ for each $i \in [t]$. Then graphs G_i, $i \in [t]$, can be interpreted as sharing the same vertex set specified in Definition 9.3.0.1, so $V(G_i) = V(G_j)$ for all $i, j \in [t]$.

We define the **collection graph** $G = (V, E)$ of our OR-PPC instance as the edge union of graphs G_i, $i \in [t]$, so that

$$G = G_1 \cup \cdots \cup G_t.$$

We use $V = V_1 \sqcup V_2 \sqcup \ldots \sqcup V_{2d}$ with $V_j = \{v_{j,\ell} \mid \ell \in [w]\}$ to denote the $2d$ layers of the collection graph. The collection graph contains all the matchings in each of the graphs G_i, $i \in [t]$.

We can prove the following properties about the collection graph.

Observation 9.3.3. *For any instance of $\texttt{OR-PPC}_{t,d,w}$, if $b = 1$, then the distance between vertices $v_{1,1}$ and $v_{2d,1}$ in collection graph G is at most $\text{dist}_G(v_{1,1}, v_{2d,1}) \leq 2d - 1$.*

Proof. When $b = 1$, there is an index $i^* \in [t]$ such that $(G^1_{i^*}, G^2_{i^*})$ is sampled from $\mu^1_{\texttt{PPC}}$. By the definition of $\mu^1_{\texttt{PPC}}$, we have that $\textsc{last}_{G^1_{i^*}}(v_{1,1}) = \textsc{last}_{G^2_{i^*}}(v_{1,1})$. Additionally, by Observation 9.3.2 it follows that $\textsc{last}_{G_{i^*}}(v_{1,1}) = v_{2d,1}$, so graph G_{i^*} contains a path of length $2d-1$ between $v_{1,1}$ and $v_{2d,1}$. Since G_{i^*} is a subgraph of the collection graph G, we conclude that there is a path of length $2d-1$ between $v_{1,1}$ and $v_{2d,1}$ in G as well. ∎

Claim 9.3.4. *For any instance of* $\texttt{OR-PPC}_{t,d,w}$*, if* $b = 0$*, then with probability at least* $1 - \frac{(2t)^{2\alpha d}}{w}$*, the distance between vertices* $v_{1,1}$ *and* $v_{2d,1}$ *in collection graph* G *is at least*

$$\text{dist}_G(v_{1,1}, v_{2d,1}) > \alpha \cdot 2d.$$

Proof. When $b = 0$, graph pairs (G^1_i, G^2_i) are sampled from $\mu^0_{\texttt{PPC}}$ for each $i \in [t]$. Then the matchings in graph G^1_i and in graph G^2_i are chosen uniformly at random for each $i \in [t]$.

The collection graph $G = (V, E)$ is composed of $2d$ layers. The edge set E of collection graph G can be characterized as follows:

- Every edge $e \in E$ is between adjacent layers V_i and V_{i+1} for some $i \in [2d-1]$.
- For each $i \in [1, d-1] \cup [d+1, 2d-1]$, the edge set $E(V_i, V_{i+1})$ is composed of t perfect matchings between V_i and V_{i+1}, each chosen uniformly at random.
- The edge set between layers V_d and V_{d+1} is the perfect matching $\{(v_{d,j}, v_{d+1,j}) \mid j \in [w]\}$.

Let $B \subseteq V$ denote the set of vertices in G at a distance at most $\alpha \cdot 2d$ from $v_{1,1}$. Since every vertex in G has degree at most $2t$, we have that $|B| \leq (2t)^{\alpha \cdot 2d}$. Additionally, for each $u, v \in V_{2d}$,

$$\Pr[u \in B] = \Pr[v \in B].$$

This equality follows from a simple bijection between the probability events in the distribution of the collection graph G. Let the set of graphs $\mathcal{Z} = \{H_1, \ldots, H_z\}$ denote the support of random graph G. For each graph $H \in \mathcal{Z}$, let $\phi(H)$ denote the graph H with the labels for vertices u and v swapped. Then $\phi : \mathcal{Z} \mapsto \mathcal{Z}$ is a bijection. Moreover, $\Pr[G = H] = \Pr[G = \phi(H)]$ for each $H \in \mathcal{Z}$, by our earlier characterization of collection graph G. Then $\Pr[u \in B] = \Pr[v \in B]$, as claimed.

Since each vertex in V_{2d} is contained in B with equal probability, we know that

$$|V_{2d}| \cdot \Pr[v_{2d,1} \in B] = \sum_{v \in V_{2d}} \Pr[v \in B] \leq \mathbb{E}[|B|] \leq (2t)^{2\alpha d},$$

where the final inequality follows from our earlier observation that $|B| \leq (2t)^{2\alpha d}$. Rearranging gives $\Pr[v_{2d,1} \in B] \leq (2t)^{2\alpha d}/w$, completing the proof. ∎

Using Observation 9.3.3 and Claim 9.3.4, we can now complete Item (*i*) of our reduction with the following lemma.

Lemma 9.3.5. *Suppose there exists a streaming algorithm* $\mathcal{A}$ *that* α*-approximates the distance between* u, v *in an* n*-vertex graph with success probability at least* $2/3$ *in* p *passes and* s

bits of space, for some $\alpha \geq 1$ *and* $p \leq \frac{\log n}{4\alpha} - 2$. *Then there is a protocol* π *for the* $\texttt{OR-PPC}_{t,d,w}$ *problem with parameters*

$$t = n^{\frac{1}{4\alpha(p+2)}}, \quad d = p + 2, \quad \text{and} \quad w = \frac{n}{2(p+2)},$$

such that π *has success probability at least* $2/3 - \log(n) \cdot n^{-1/4}$ *and communication cost* s.

Proof. Let $t = n^{\frac{1}{4\alpha(p+2)}}, d = p + 2,$ and $w = \frac{n}{2(p+2)}$. Let $G = (V, E)$ be the collection graph corresponding to an instance of $\texttt{OR-PPC}_{t,d,w}$. Notice that G has $|V| = 2dw = n$ vertices.

Protocol for $\texttt{OR-PPC}_{t,d,w}$**.** Alice and Bob will simulate streaming algorithm $\mathcal{A}$ to compute an α-approximation of the shortest path distance between $v_{1,1}$ and $v_{2d,1}$ in G. In rounds where Bob sends the message, Bob first runs algorithm $\mathcal{A}$ locally, using his edges in G as the input stream to $\mathcal{A}$. Then Bob sends the final work state $\mathcal{W}$ of $\mathcal{A}$ to Alice using s bits of communication. Alice then continues running algorithm $\mathcal{A}$ starting from work state $\mathcal{W}$, while streaming her edges in G to the algorithm. In this manner, Alice and Bob simulate one pass of $\mathcal{A}$ using one round and s communication. We can use a similar protocol in rounds where Alice sends the message.[2]

Then in $p \leq d-2$ rounds and s communication cost, Alice and Bob can completely simulate $\mathcal{A}$ and compute an α-approximation $\sigma \in \mathbb{R}_{\geq 0}$ of $\text{dist}_G(v_{1,1}, v_{2d,1})$ with success probability at least $2/3$. Recall that if σ is an α-approximation of $\text{dist}_G(v_{1,1}, v_{2d,1})$, then

$$\text{dist}_G(v_{1,1}, v_{2d,1}) \leq \sigma \leq \alpha \cdot \text{dist}_G(v_{1,1}, v_{2d,1}).$$

The player who receives the last message does the following:

- outputs $b = 1$ if $\sigma \leq \alpha \cdot (2d - 1)$, and
- outputs $b = 0$ if $\sigma > \alpha \cdot (2d - 1)$.

Correctness. If $b = 1$, then with probability at least $2/3$, algorithm $\mathcal{A}$ succeeds, so $\sigma \leq \alpha \cdot (2d-1)$ by Observation 9.3.3, and our protocol outputs $b = 1$. Then our protocol succeeds with probability at least $2/3$ when $b = 1$.

If $b = 0$, then by Claim 9.3.4, $\text{dist}_G(v_{1,1}, v_{2d,1}) \leq \alpha \cdot 2d$ with probability at most

$$\begin{aligned}
\Pr[\text{dist}_G(v_{1,1}, v_{2d,1}) \leq \alpha \cdot 2d] &\leq \frac{(2t)^{2\alpha d}}{w} \\
&= \frac{2^{2\alpha d \cdot (1+\log t)}}{w} \\
&\leq \frac{2^{2\alpha d \cdot \left(\frac{3}{2} \cdot \log t\right)}}{w} && \text{(since } \log t \geq 2\text{)}
\end{aligned}$$

[2]Notice that the order of edges in the stream are not the same in different passes in our simulation. If it is required by the algorithm that the order of edges is the same across all the passes, this can be done in $2p$ rounds. Asymptotically all the results remain the same.

$$
\begin{aligned}
&= \frac{2^{2\alpha d \cdot \left(\frac{3}{2} \cdot \frac{1}{4\alpha d(p+2)} \cdot \log n\right)}}{w} && \left(\text{since } t = n^{\frac{1}{4\alpha(p+2)}}\right) \\
&\leq \frac{2^{3/4 \log n}}{w} && (\text{since } p \geq 1) \\
&= 2dn^{-1/4} && (\text{since } w = n/(2d)) \\
&\leq 2\log(n) \cdot n^{-1/4} && (\text{since } d \leq \log n)
\end{aligned}
$$

Then by the union bound, with probability at least $2/3 - 2\log(n) \cdot n^{-1/4}$ algorithm $\mathcal{A}$ succeeds and $\sigma \geq \text{dist}_G(v_{1,1}, v_{2d,1}) > \alpha \cdot 2d$. Thus when $b = 0$, our protocol succeeds with probability at least $2/3 - 2\log(n) \cdot n^{-1/4}$.

Since $\texttt{OR-PPC}_{t,d,w}$ is distributionally defined so that $\Pr[b = 1] = \Pr[b = 0] = \frac{1}{2}$, our protocol succeeds with probability at least

$$\frac{1}{2} \cdot 2/3 + \frac{1}{2} \cdot (2/3 - 2\log(n) \cdot n^{-1/4}) \geq 2/3 - \log(n) \cdot n^{-1/4},$$

which completes the proof. ∎

We have completed Item (i) in our lower bound framework, by translating semi-streaming algorithms for approximate s-t shortest path into protocols for OR-PPC. We finish the proof of Theorem 13 in the next subsection.

9.3.0.2 Finishing the Proof of Theorem 13

In the next step of our lower bound proof, we translate protocols for OR-PPC into protocols for PPC. This reduction, summarized in Proposition 9.3.6, uses a direct sum argument combined with message compression techniques. We defer the proof of Proposition 9.3.6 the full version [AHS26], as it uses standard direct sum [BBCR10], message compression techniques [HJMR07, RY20] and information odometer techniques [Ass17, GJPW18].

Proposition 9.3.6. *For any $t, d, w, s \geq 1$, for any $0 < \delta < 1$, given any protocol π_{OR} for $\texttt{OR-PPC}_{t,d,w}$ that uses communication at most s bits, and has probability of error at most δ, there is a protocol $\pi_{\texttt{PPC}}$ for $\texttt{PPC}_{d,w}$ that has communication at most $c_{\text{OR}} \cdot (d-2) \cdot (s/t+1)$ bits for some absolute constant $c_{\text{OR}} \geq 1$, and has probability of error at most $\delta + 1/200$.*

The final step in our proof of Theorem 13 is a lower bound on the communication cost of the paired pointer chasing problem. This lower bound is stated in Lemma 9.3.7 and proved in Section 9.3.0.3.

Lemma 9.3.7. *For any integers $d, w \geq 2$, any deterministic protocol π for $\texttt{PPC}_{d,w}$ that achieves a success probability of at least 0.95 over the distribution $\mu_{\texttt{PPC}}$, must have cost $\Omega(w/d^5 - \log w)$.*

We can now put everything together and prove Theorem 13, which is restated below.

Proof. Let $\alpha \geq 1$, and let p be an integer such that $p \leq \log(n)/(4\alpha) - 2$. Let $\mathcal{A}$ be a streaming algorithm (deterministic or randomized) on unweighted graphs that, with success

probability at least 0.99, can output an α-approximation to the distance between a fixed pair of vertices using at most s space and at most p passes.

Then by Lemma 9.3.5, there is a protocol π for the $\texttt{OR-PPC}_{t,d,w}$ problem with parameters

$$t = n^{\frac{1}{4\alpha(p+2)}}, \quad d = p+2, \quad \text{and} \quad w = n/(2(p+2)),$$

such that π has success probability at least $0.99 - \log(n) \cdot n^{-1/4} \geq 0.98$ and communication s.

By applying Proposition 9.3.6, this implies there is a protocol $\pi_{\texttt{PPC}}$ for $\texttt{PPC}_{d,w}$ with success probability at least 0.95 and communication cost at most

$$c_{\text{OR}} \cdot (d-2) \cdot (s/t+1) \leq c_{\text{OR}} \cdot d \cdot (s/t+1)$$

bits, for some absolute constant $c_{\text{OR}} \geq 1$.

Finally, Lemma 9.3.7 implies that the communication cost of $\pi_{\texttt{PPC}}$ is at least $\Omega(w/d^5 - \log w)$. Combining our upper and lower bounds on the cost of $\pi_{\texttt{PPC}}$ gives

$$c_{\text{OR}} \cdot d \cdot (s/t+1) = \Omega(w/d^5 - \log w).$$

Solving for s and plugging in the values of t, d, and w, we obtain

$$s = \Omega\left(\frac{tw}{d^6} - \frac{t}{d}\log(w) - t\right) = \Omega\left(\frac{n^{1+1/(4\alpha(p+2))}}{2(p+2)^7} - n^{1/(4\alpha(p+2))}\log(n)\right) = \Omega\left(\frac{n^{1+1/(4\alpha(p+2))}}{p^7}\right).$$

We have shown that any streaming algorithm that, with success probability at least 0.99, can output an α-approximation to the distance between a fixed pair of vertices using at most p passes requires at least $\Omega\left(\frac{n^{1+1/(4\alpha(p+2))}}{p^7}\right)$ space. We can extend this lower bound to hold for streaming algorithms that succeed with probability at least 2/3, by using a standard probability of success amplification argument (running the algorithm in parallel $O(1)$ times and return the median answer). ∎

9.3.0.3 Lower Bound for Paired Pointer Chasing

We prove Lemma 9.3.7, which is the lower bound for PPC in this subsection. We show that we can reduce PPC to a standard pointer chasing problem through a series of probabilistic arguments.

We start by defining the standard pointer chasing problem. This communication problem is not defined with an inherent distribution, as we will analyze it on various distributions of the input while proving Lemma 9.3.7.

Definition (Pointer Chasing)**.** *For integers $w, d \geq 2$, the pointer chasing problem, denoted by $\texttt{PC}_{d,w}$ is defined as follows.*

(*i*) *The input is a graph $G \in \mathcal{L}_{d,w}$ split between the players. Alice receives M_i for all odd $i \in [d-1]$ and Bob receives M_i for all even $i \in [d-1]$.*

(*ii*) *There is an arbitrary partition of vertex set $V_d \subset V$ into sets $W, \overline{W}$ known to both players.*

(*iii*) *The players speak for a total of $d-2$ rounds. At the end of all the rounds, the player who receives the last message has to output whether $\textsc{last}_G(v_{1,1}) \in V_d$ belongs to set W or $\overline{W}$.*

$\mathtt{PPC}_{d,w}$ problem can be viewed as an instance of $\mathtt{PC}_{d,w^2}$ naturally. We use $G^\times = (V^\times, E^\times)$ to denote the graph $G^1 \times G^2$ where G^1 and G^2 are graphs in an instance of $\mathtt{PPC}$. For $i \in [d]$, we use $V_i^\times$ to denote the partitions of $V^\times$ into d layers. For $i \in [d-1]$, we use $M_i^\times$ to denote the perfect matching between layers $V_i^\times$ and $V_{i+1}^\times$ of $V^\times$ in graph $G^\times$ (see Observation 9.3.1 for properties of the product graph $G^\times$).

Observation 9.3.8. *In any instance of $\mathtt{PPC}_{d,w}$, Alice knows all $M_i^\times$ for odd $i \in [d-1]$ and no other edges. Similarly, Bob knows $M_i^\times$ for all even $i \in [d-1]$ and no other edges.*

Proof. In Definition 9.3.0.1, for any $i \in [d-1]$, we know that $M_i^\times$ depends only on M_i^1 and M_i^2, which are the i^{th} matchings in G^1 and G^2, respectively. For even i, both of these matchings are known to Bob, and neither of them are known to Alice. Likewise, for odd i, both of the matchings are known to Alice, whereas Bob knows neither of them. ∎

Observation 9.3.9. *In any instance of $\mathtt{PPC}_{d,w}$ with graphs G^1 and G^2, the distribution is chosen so that,*

- *When $\mathsf{b} = 0$, all the matchings in instances G^1 and G^2 are chosen uniformly at random and independently of each other.*
- *When $\mathsf{b} = 1$, all the matchings in instances G^1 and G^2 are chosen uniformly at random, conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) = (v_{d,i}, v_{d,i})$ for some $i \in [w]$. Moreover, the value of $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ is uniform over the set $\{(v_{d,i}, v_{d,i}) \mid i \in [w]\} \subset V_d^\times$.*

Proof. The distribution of the matchings in G^1 and G^2 when $\mathsf{b} = 0$ are exactly as described in Definition 9.3.0.1. When $\mathsf{b} = 1$, we know that in $\mathtt{PPC}_{d,w}$, the two graphs G^1 and G^2 are chosen so that $\textsc{last}_{G^1}(v_{1,1}) = \textsc{last}_{G^2}(v_{1,1}) = v_{d,j}$ for some $j \in w$. By part (*ii*) of Observation 9.3.1, we know that $\textsc{last}_{G^\times}((1,1)) = (\textsc{last}_{G^1}(1), \textsc{last}_{G^2}(1)) = (v_{d,j}, v_{d,j})$ also. For the "Moreover" part, the distribution of the matchings is symmetric across all elements of the set $\{(v_{d,i}, v_{d,i}) \mid i \in [w]\}$, and thus $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ is uniform over this set. ∎

Now, we know that the value of b induces a distribution over $\mathcal{L}_{d,w^2}$, where the two graphs G^1 and G^2 are sampled so that $G^\times$ obeys the properties in the statement of Observation 9.3.9. By Observation 9.3.8, we know that the graph $G^\times$ is split between Alice and Bob exactly the same as if it was given as an instance of $\mathtt{PC}_{d,w^2}$. However, observe that as of now, there is **no partition** of $V_d^\times$ in $G^\times$ into two sets $W, \overline{W}$ known to both Alice and Bob, as in Definition 9.3.0.3. They have to find the **value of *b***, and not where the final pointer $\textsc{last}_{G^\times}((1,1))$ is. We will choose sets $W, \overline{W}$, so that finding the value of b is (almost) equivalent to finding where $\textsc{last}_{G^\times}((1,1))$ lies.

Let $\mu_{\mathtt{PC}}$ denote the distribution of the random variables in input of $\mathtt{PC}_{d,w^2}$ induced by distribution $\mu_{\mathtt{PPC}}$, along with that of bit b. It is sufficient to show that outputting the value of b is hard in $\mathtt{PC}_{d,w^2}$ under distribution $\mu_{\mathtt{PC}}$. We work with another distribution $\tilde{\mu}_{\mathtt{PC}}$ (again over graphs from $\mathtt{PC}_{d,w^2}$ and bit b). Distribution $\tilde{\mu}_{\mathtt{PC}}$ is "close" to $\mu_{\mathtt{PC}}$, which will be useful in our analysis.

Distribution 9. Distribution $\tilde{\mu}_{\mathtt{PC}}$:

(*i*) Let set $S = \{(v_{d,i}, v_{d,i}) \mid i \in [w]\}$. Choose set T of w elements uniformly at random from the set $V_d^{\times} \setminus S$.

(*ii*) Sample graphs G^1 and G^2 uniformly at random conditioned on $\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T$.

(*iii*) When $\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S$, fix $\mathsf{b} = 1$, and set $\mathsf{b} = 0$ otherwise.

We will show that the distribution $\mu_{\mathtt{PC}}$ and $\tilde{\mu}_{\mathtt{PC}}$ are close to each other in total variation distance.

Claim 9.3.10. *In distribution $\tilde{\mu}_{\mathtt{PC}}$, for any choice of set T, the value of $\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1}))$ is uniform over the set $S \cup T$.*

Proof. Firstly, we know that when the matchings in G^1 and G^2 are chosen uniformly at random, the values of $\textsc{last}_{G^1}(v_{1,1})$ and $\textsc{last}_{G^2}(v_{1,1})$ are uniform over the set V_d^1 and V_d^2 respectively by symmetry. Moreover, when the matchings are chosen independently of each other, the value of $\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1}))$ is uniform over the set $V_d^{\times} = V_d^1 \times V_d^2$, by part (*ii*) of Observation 9.3.1. Therefore,

$$\Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T] = \frac{|S \cup T|}{w^2}. \tag{9.1}$$

Let $u = (x, y) \in S \cup T$ be any arbitrary vertex in $V_d^{\times}$ with $x \in V_d^1$ and $y \in V_d^2$. We show that,

$$\Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) = u = (x, y) \mid \textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T] = \frac{1}{|S \cup T|}, \tag{9.2}$$

which is sufficient to prove the claim.

We have,

$$\begin{aligned}
&\Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) = u \mid \textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T] \\
&= \frac{1}{\Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T]} \cdot \Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) = u \cap \textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) \in S \cup T] \\
&\qquad\qquad \text{(by law of total probability)} \\
&= \frac{w^2}{|S \cup T|} \cdot \Pr[\textsc{last}_{G^{\times}}((v_{1,1}, v_{1,1})) = u = (x, y)] \qquad \text{(by Eq (9.1))}
\end{aligned}$$

$$= \frac{w^2}{|S \cup T|} \cdot \frac{1}{w^2} \qquad \text{(as } \textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \text{ is uniform over } V_d^\times\text{)}$$
$$= \frac{1}{|S \cup T|},$$

which completes the proof. ∎

Claim 9.3.11. *Distributions $\mu_{\mathtt{PC}}$ and $\tilde{\mu}_{\mathtt{PC}}$ are close to each other in total variation distance.*

$$\|\mu_{\mathtt{PC}} - \tilde{\mu}_{\mathtt{PC}}\|_{\text{tvd}} \leq 1/2w.$$

Proof. The random variables involved in distributions $\mu_{\mathtt{PC}}$ and $\tilde{\mu}_{\mathtt{PC}}$ are the bit b, and all the matchings in G^1, G^2 which lead to $G^\times \in \mathcal{L}_{d,w^2}$. First, we argue that the distribution of b is uniform in $\mu_{\mathtt{PC}}$ and $\tilde{\mu}_{\mathtt{PC}}$, and then we show that for each value of b, the distribution of the matchings are close to each other. Finally, we use the weak chain rule of total variation distance in Fact 4.3.8 to finish the proof.

Distribution of b: In distribution $\mu_{\mathtt{PC}}$, the value of b is uniform over $\{0,1\}$ by construction. In $\tilde{\mu}_{\mathtt{PC}}$, we know that the value of $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ is uniform over $S \cup T$ by Claim 9.3.10. As $|S| = |T| = w$ by construction, the value of bit b is chosen uniformly at random from $\{0,1\}$ in $\tilde{\mu}_{\mathtt{PC}}$.

When $\mathsf{b} = 1$: In distribution $\mu_{\mathtt{PC}}$, when $\mathsf{b} = 1$, we know that all matchings are chosen uniformly at random conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in S$. We will argue that this is true in $\tilde{\mu}_{\mathtt{PC}}$ also.

In $\tilde{\mu}_{\mathtt{PC}}$, we know that matchings are chosen uniformly random conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ lying in $S \cup T$. However, when $\mathsf{b} = 1$, we know that $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ lies only in S by definition of $\tilde{\mu}_{\mathtt{PC}}$. Hence, the matchings are chosen uniformly at random conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in S$.

When $\mathsf{b} = 0$: In distribution $\mu_{\mathtt{PC}}$, when $\mathsf{b} = 0$, all the matchings are chosen uniformly at random and independently of each other. Here, we have,

$$\Pr[\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in S \mid b = 0] = \frac{1}{w^2} \cdot |S| = \frac{1}{w}, \tag{9.3}$$

by part (ii) of Observation 9.3.1, and by the matchings being uniformly random and independent in the graphs G^1 and G^2.

In distribution $\tilde{\mu}_{\mathtt{PC}}$, conditioned on $\mathsf{b} = 0$, we know that the matchings are chosen uniformly at random and independent of each other conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in T$ for some uniformly random set T of w elements from $(V_d^1 \times V_d^2) \setminus S$. By the uniformity of T over set $V_d^\times \setminus S$, this is the same as sampling all matchings uniformly at random conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \notin S$.

We have argued that the distribution $\tilde{\mu}_{\mathtt{PC}} \mid b = 0$, is the same as distribution $\mu_{\mathtt{PC}} \mid b = 0$ conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \notin S$.

Therefore, by Lemma 4.3.6, we have,

$$\|(\mu_{\mathtt{PC}} \mid b = 0) - (\tilde{\mu}_{\mathtt{PC}} \mid b = 0)\|_{\text{tvd}} \leq \Pr[\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in S \mid b = 0] = \frac{1}{w}, \tag{9.4}$$

where the last equality follows by Eq (9.3).

Now, we can complete the proof by Fact 4.3.8.

$$\begin{aligned}
\|\mu_{\mathtt{PC}} - \tilde{\mu}_{\mathtt{PC}}\|_{\mathrm{tvd}} &\le \|\mu_{\mathtt{PC}}(\mathsf{b}) - \tilde{\mu}_{\mathtt{PC}}(\mathsf{b})\|_{\mathrm{tvd}} + \mathop{\mathbb{E}}_{\mathsf{b}=b} \|(\mu_{\mathtt{PC}} \mid \mathsf{b} = b) - (\tilde{\mu}_{\mathtt{PC}} \mid \mathsf{b} = b)\|_{\mathrm{tvd}} \\
&= 0 + \mathop{\mathbb{E}}_{\mathsf{b}=b} \|(\mu_{\mathtt{PC}} \mid \mathsf{b} = b) - (\tilde{\mu}_{\mathtt{PC}} \mid \mathsf{b} = b)\|_{\mathrm{tvd}} \\
&\qquad \text{(as } \mathsf{b} \text{ is uniform over } \{0,1\} \text{ in both } \mu_{\mathtt{PC}} \text{ and } \tilde{\mu}_{\mathtt{PC}}\text{)} \\
&= \frac{1}{2} \cdot (\|(\mu_{\mathtt{PC}} \mid b = 1) - (\tilde{\mu}_{\mathtt{PC}} \mid b = 1)\|_{\mathrm{tvd}} + \|(\mu_{\mathtt{PC}} \mid b = 0) - (\tilde{\mu}_{\mathtt{PC}} \mid b = 0)\|_{\mathrm{tvd}}) \\
&\qquad \text{(again, as } \mathsf{b} \text{ is uniform)} \\
&= \frac{1}{2} \cdot (0 + \|(\mu_{\mathtt{PC}} \mid b = 0) - (\tilde{\mu}_{\mathtt{PC}} \mid b = 0)\|_{\mathrm{tvd}}) \\
&\qquad \text{(as we have argued, distributions are same conditioned on } \mathsf{b} = 1\text{)} \\
&\le \frac{1}{2w},
\end{aligned}$$

where the last inequality follows by Eq (9.4). ∎

By Claim 9.3.11, as the distributions are close to each other, it is sufficient to prove that finding the value of b is hard under distribution $\tilde{\mu}_{\mathtt{PC}}$. We have made some progress from $\mu_{\mathtt{PC}}$ in getting closer to standard pointer chasing: now, we have two sets S and T such that when $\mathsf{b} = 0$, the final pointer is in S and when $\mathsf{b} = 1$, the pointer is in T. When the set $V_d^{\times}$ is restricted to $S \cup T$, these sets S and T take the role of W and $\overline{W}$ in Definition 9.3.0.3 of standard pointer chasing. We continue to work with distribution $\tilde{\mu}_{\mathtt{PC}}$.

The final pointer we are interested in is distributed uniformly at random by Claim 9.3.10, however the distribution of the specific matchings in $\tilde{\mu}_{\mathtt{PC}}$ are far from uniform. They are from the product of the matchings in G^1 and G^2. In the next step of this subsection, we show that we can pick subsets of the vertex set $V^{\times}$ in graph $G^{\times}$ sampled from $\tilde{\mu}_{\mathtt{PC}}$ so that these matchings become uniformly random and independent of each other.

Definition 9.3.12. *A subset $P \subseteq V_d^{\times}$ is said to be a **rook set** if for any $(x, y) \in P$, $(x, y') \notin P$ for any $y \neq y'$ and $(x', y) \notin P$ for any $x' \notin P$.*[3]

For any subset $P \subset V_d^{\times}$, for $i \in [d-1]$, we use $V_i^{\times}(P)$ to denote the set of $|P|$ elements in $V_i^{\times}$ that have a path to some vertex in P in graph $G^{\times}$. For the last layer, $V_d^{\times}(P)$ will be equal to P.

Claim 9.3.13. *For any rook set P, for any $i \in [d]$, $V_i^{\times}(P)$ is also a rook set.*

Proof. The proof is by induction on i, ranging from d and decreasing to 1. The base case is when $i = d$, and $V_d^{\times}(P) = P$. The base case holds trivially as P is a rook set.

Let us assume towards a contradiction that $V_{i-1}^{\times}(P)$ is not a rook set while $V_i^{\times}(P)$ is a rook set. There exist two elements (x, y) and (x', y) in $V_{i-1}^{\times}(P)$. We consider the two elements in $V_i^{\times}(P)$ that (x, y) and (x', y) are mapped to by the matching $M_{i-1}^{\times}$. Let these two elements

[3]The name rook set has been chosen because when visualized as a grid of $w \times w$ squares, at most one element can be chosen from each row or column, similar to a non-attacking rook placement on a chess board.

be (u_1, v_1) and (u_2, v_2). But, we know that $v_1 = v_2$, as these are the vertices from layer V_i^2 of G^2 that was used to construct $G^\times$. This contradicts our assumption that $V_i^\times(P)$ is a rook set. The case when some (x, y) and (x, y') belongs to $V_{i-1}^\times(P)$ can be analyzed similarly. ∎

Claim 9.3.14. *In distribution $\tilde{\mu}_{\mathtt{PC}}$, for any rook set P, conditioned on $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in P$, and all choices of $V_i^\times(P)$ for $1 \le i \le d$, for all $1 \le j \le d-1$, the subset of $M_j^\times$ between sets $V_j^\times(P)$ and $V_{j+1}^\times(P)$ is a uniformly random permutation independent of the choices of all other matchings, and all edges of $M_j^\times$ not incident to $V_j^\times(P)$ and $V_{j+1}^\times(P)$.*

Proof. First, we observe that $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1}))$ is uniform over the set P. This argument is the same as that of Claim 9.3.10.

We will show that the matching edges between $V_{i-1}^\times(P)$ and $V_i^\times(P)$ are uniformly random conditioned on any choice of all other matchings, and all edges of $M_{i-1}^\times$ not incident on $V_{i-1}^\times(P), V_i^\times(P)$ for any $i \in [d-1]$. Henceforth in this proof, we assume that all these other random variables are fixed to be something arbitrary.

Next, we observe that the condition of $\textsc{last}_{G^\times}((v_{1,1}, v_{1,1})) \in P$ is satisfied by the conditioning on these random variables already, as the vertex $(v_{1,1}, v_{1,1}) \in V_1^\times$ belongs to the set $V_1^\times(P)$, and hence has a path to some vertex in P. Therefore, now, in distribution $\tilde{\mu}_{\mathtt{PC}}$, all the matchings M_j^1 and M_j^2 for $j \in [d-1]$ are chosen uniformly at random and independently of each other, conditioned on the choices of $V_j^\times(P)$ for all $j \in [d]$ by definition. This implies that the matchings M_{i-1}^1 and M_{i-1}^2 are independent of the choices of all the other matchings, and the edges not incident to $V_{i-1}^\times(P), V_i^\times(P)$.

It remains to argue that when the edges of the two matchings M_{i-1}^1, M_{i-1}^2 incident on $V_{i-1}^\times(P)$, $V_i^\times(P)$ are uniformly random conditioned on the choices of the two sets $V_{i-1}^\times(P)$, $V_i^\times(P)$, the respective edges in $M_{i-1}^\times$ are uniformly random as well.

Let us take any two arbitrary vertices $(x, y) \in V_{i-1}^\times(P)$ and $(u, v) \in V_i^\times(P)$. We will show that the probability of (x, y) being mapped to (u, v) is exactly $1/\left|V_i^\times(P)\right|$, which is sufficient to prove the claim.

Let $k = \left|V_i^\times(P)\right| = \left|V_{i-1}^\times(P)\right|$. First, we analyze what the probability is that the entirety of set $V_{i-1}^\times(P)$ is mapped to $V_i^\times(P)$.

$$\begin{aligned}
&\Pr[M_{i-1}^\times \text{ maps } V_{i-1}^\times(P) \text{ to } V_i^\times(P)] \\
&= \text{Number of ways to arrange } V_i^\times(P) \\
&\qquad \cdot \text{Probability of getting a specific mapping from } V_{i-1}^\times(P) \text{ to } V_i^\times(P) \\
&= k! \cdot \left(\frac{(n-k)!}{n!}\right)^2,
\end{aligned}$$

as there are $k!$ ways of arranging set $V_i^\times(P)$, and for any specific ordering, k of the edges in both matchings M_{i-1}^1 and M_{i-1}^2 are fixed. The rest of the edges in both matchings remain fully random.

We also have,

$$\Pr[(x, y) \text{ is mapped to } (u, v) \text{ and } M_{i-1}^\times(P) \text{ maps } V_{i-1}^\times(P) \text{ to } V_i^\times(P)]$$

$$
\begin{aligned}
&= \Pr[(x,y) \text{ is mapped to } (u,v)] \\
&\qquad \cdot \Pr[M^{\times}_{i-1}(P) \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P) \mid (x,y) \text{ is mapped to } (u,v)] \\
&= \frac{1}{n} \cdot \frac{1}{n} \cdot \Pr[M^{\times}_{i-1}(P) \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P) \mid (x,y) \text{ is mapped to } (u,v)] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(as } M^{1}_{i-1} \text{ maps } x \text{ to } u \text{ and } M^{2}_{i-1} \text{ maps } y \text{ to } v) \\
&= \frac{1}{n^2} \cdot \Pr[M^{\times}_{i-1}(P) \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P) \mid (x,y) \text{ is mapped to } (u,v)] \\
&= \frac{1}{n^2} \cdot (k-1)! \cdot \left(\frac{(n-k)!}{(n-1)!}\right)^2,
\end{aligned}
$$

where for the last equality, we know that conditioned on (x,y) being mapped to (u,v), the rest of the $n-1$ edges of both M^{1}_{i-1} and M^{2}_{i-1} are fully random. The probability that the vertices of $V^{\times}_{i-1}(P) \setminus \{(x,y)\}$ are mapped to $V^{\times}_{i}(P) \setminus \{(u,v)\}$ will be the number of ways of arranging the vertices of $V^{\times}_{i}(P) \setminus \{(u,v)\}$ times the probability of a specific mapping from $V^{\times}_{i-1}(P) \setminus \{(x,y)\}$ to $V^{\times}_{i}(P) \setminus \{(u,v)\}$ being realized.

We can continue the proof as follows.

$$
\begin{aligned}
&\Pr[(x,y) \text{ is mapped to } (u,v) \mid M^{\times}_{i-1}(P) \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P)] \\
&= \frac{\Pr[(x,y) \text{ is mapped to } (u,v) \text{ and } M^{\times}_{i-1}(P) \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P)]}{\Pr[M^{\times}_{i-1} \text{ maps } V^{\times}_{i-1}(P) \text{ to } V^{\times}_{i}(P)]} \\
&= \frac{1}{n^2} \cdot (k-1)! \cdot \left(\frac{(n-k)!}{(n-1)!}\right)^2 \cdot \frac{1}{k!} \cdot \left(\frac{n!}{(n-k)!}\right)^2 \qquad \text{(from our equations above)} \\
&= \frac{1}{k} = \frac{1}{\left|V^{\times}_{i}(P)\right|},
\end{aligned}
$$

which completes the proof. ∎

We are very close to the standard pointer chasing problem over the uniform distribution with independent matchings. We will choose a rook set P which is a subset of $S \cup T$ of large size, so that conditioned on the choices of $V^{\times}_{i}(P)$ for all $i \in [d-1]$, the distribution of the matchings in $\mathtt{PC}_{d,w^2}$ is uniformly random.

We need the following standard results about size of the largest matching in random bipartite graphs. The proof of this result is given in Section A.2 for completeness.

Proposition 9.3.15. *For any large enough integer $k \geq 0$, in a random bipartite graph $G = (L \sqcup R, E)$ where $L = R = \{1, 2, \ldots, k\}$, and E is k edges chosen uniformly at random without repetition from the set*

$$E \subset L \times R \setminus \{(i,i) \mid i \in [k]\},$$

with probability at least $1 - 1/k^2$, there exists a matching in G of size at least $0.1k$.

The next claim follows almost directly from Proposition 9.3.15.

Claim 9.3.16. *In any set T of w elements chosen unifomly at random from $[w] \times [w] \setminus S$ as in Distribution 9, there is a rook set $\tilde{T} \subset T$ with $\left|\tilde{T}\right| = 0.1w$ with probability at least $1 - 1/w^2$ over the randomness of sampling set T.*

Proof. We will look at a bipartite graph with vertex set $L = R = [w]$. The randomness in sampling set T from $[w] \times [w] \setminus S$ is the same as sampling a random set of w edges from the set

$$L \times R \setminus \{(i, i) \mid i \in [w]\}.$$

We apply Proposition 9.3.15 with $k = w$. It is sufficient to show that matchings in our graph correspond to rook sets. Let $M \subset E$ be a matching in the graph G. Assume towards a contradiction that $M \subset L \times R = [w] \times [w]$ is not a rook set. Then there exists some $(x, y), (x, y') \in M$ with $x, y, y' \in [w]$ and $y \neq y'$. However, the vertex $x \in L$ now has two edges incident on it, namely (x, y) and (x, y'), contradicting the fact that M is a matching.

The case when some $(x, y), (x', y) \in M$ with $x, x', y \in [w]$ and $x \neq x'$ is handled similarly, where vertex $y \in R$ will have more than one edge incident on it in M. ∎

In the next claim, we show that we can add to $\tilde{T}$ from set S and create a larger rook set.

Claim 9.3.17. *Let $\tilde{T} \subset T$ be the set from Claim 9.3.16. It is possible to choose $\tilde{S} \subset S$ with $\left|\tilde{S}\right| = 0.1w$ such that $\tilde{S} \cup \tilde{T}$ is a rook set of size $0.2w$.*

Proof. We know that S is a rook set, as $S = \{(i, i) \mid i \in [w]\}$. Let

$$S' = \{(i, i) \mid (i, x) \in \tilde{T} \text{ or } (y, i) \in \tilde{T}, \text{ for } x, y \in [w]\}.$$

We observe that $\tilde{T} \cup (S \setminus S')$ is a rook set as all the elements $(x, x), (y, y)$ which interfere with some $(x, y) \in \tilde{T}$ have been removed. Moreover, the size of S' is at most $0.2w$, as for each element in $\tilde{T}$, at most two elements are added to S'. Hence we can choose $\tilde{S}$ as any arbitrary subset of $S \setminus S'$ of size $0.1w$. ∎

Lemma 9.3.18. *For large integers $d, w \geq 100$, there exists a choice of set T such that the following is true:*

(i) T has a rook set $\tilde{T}$ of size $0.1w$.

(ii) A set $\tilde{S} \subset S$ can be chosen such that $\left|\tilde{S}\right| = 0.1w$, and $\tilde{S} \cup \tilde{T}$ is a rook set of size $0.2w$.

(iii) Any deterministic protocol π which outputs the value of b under distribution $\mu_{\mathtt{PC}}$ with probability at least 0.95 must output whether $\textsc{last}_{G^\times}((1,1)) \in \tilde{S}$ or $\textsc{last}_{G^\times}((1,1)) \in \tilde{T}$, in distribution $\tilde{\mu}_{\mathtt{PC}}$ conditioned on choice of T, $\tilde{S}, \tilde{T}$ and $V_i^\times(\tilde{S} \cup \tilde{T})$ for all $i \in [d-1]$ and event $\textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T}$ with probability at least 0.7.

(iv) For $i, j \in [d-1]$, the subset of $M_i^\times$ between vertices of $V_i^\times(\tilde{S} \cup \tilde{T})$ and $V_{i+1}^\times(\tilde{S} \cup \tilde{T})$ is uniformly random and independent of other $M_j^\times$ for $j \neq i$.

Proof. Let π be a deterministic protocol that finds the value of b with probability at least $2/3$ when the input distribution is $\mu_{\mathtt{PC}}$. First, we have,

$$\begin{aligned}
&\Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}}] \\
&\geq \Pr[\pi \text{ succeeds on } \mu_{\mathtt{PC}}] - \|\mu_{\mathtt{PC}} - \tilde{\mu}_{\mathtt{PC}}\|_{\text{tvd}} && \text{(by Fact 4.3.5)} \\
&\geq 0.95 - 1/2w. && \text{(by Claim 9.3.11)}
\end{aligned}$$

We have argued that protocol π outputs the value of b with probability at least $0.95-1/2w$ when the input distribution is $\tilde{\mu}_{\mathtt{PC}}$. We can also charge the probability that the random set T has a large rook set or not to the probability of success.

$$\begin{aligned}
&\Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}} \mid \mathsf{T} \text{ has a large rook set }] && (9.5) \\
&\geq \Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}}] - \Pr[\mathsf{T} \text{ does not have a large rook set }] \\
& && \text{(by law of total probability)} \\
&\geq 0.95 - 1/2w - 1/w^2 && \text{(by Claim 9.3.16)} \\
&\geq 0.95 - 1/w, && (9.6)
\end{aligned}$$

where the last inequality holds for large w. We know that the success of protocol π under distribution $\tilde{\mu}_{\mathtt{PC}}$ conditioned on T having a large rook set is on average over the randomness of variable T. Therefore, there must exist some specific choice of set T such that conditioned on $\mathsf{T} = T$, the protocol π succeeds with probability at least $0.95 - 1/w$. We fix T to be this set for the statement of the lemma.

We know that we can pick $\tilde{T} \subset T$ such that the statement of Claim 9.3.16 holds for set $\tilde{T}$, as we have conditioned on T having a large rook set. This satisfies part (i) of the lemma statement.

We choose set $\tilde{S}$ from Claim 9.3.17, based on our choice of $\tilde{T}$. Part (ii) of the lemma statement is satisfied by Claim 9.3.17. For the rest of the proof, in distribution $\tilde{\mu}_{\mathtt{PC}}$, we assume that the choice of $T, \tilde{T}$ and $\tilde{S}$ are fixed as we have described.

For part (iii), we begin by analyzing the probability that $\textsc{last}_{G^\times}((1,1))$ belongs to $\tilde{S} \cup \tilde{T}$ in distribution $\tilde{\mu}_{\mathtt{PC}} \mid T, \tilde{S}, \tilde{T}$. By Claim 9.3.10, we have,

$$\Pr[\textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T}] = \frac{\left|\tilde{S} \cup \tilde{T}\right|}{|S \cup T|} = 0.2. \tag{9.7}$$

We can write,

$$\begin{aligned}
&\Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}} \mid T, \tilde{S}, \tilde{T}] \\
&\leq \Pr[\textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T} \mid T, \tilde{S}, \tilde{T}] \cdot \Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}} \mid \tilde{S}, \tilde{T}, T, \textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T}] \\
&\qquad + \Pr[\textsc{last}_{G^\times}((1,1)) \notin \tilde{S} \cup \tilde{T} \mid T, \tilde{T}, \tilde{S}] && \text{(by law of total probability)} \\
&= 0.2 \cdot \Pr[\textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T} \mid S, \tilde{T}, \tilde{S}] + 0.8. && \text{(by Eq (9.7))}
\end{aligned}$$

Hence,

$$\Pr[\pi \text{ succeeds on } \tilde{\mu}_{\mathtt{PC}} \mid \tilde{S}, \tilde{T}, T, \textsc{last}_{G^\times}((1,1)) \in \tilde{S} \cup \tilde{T}] \\ \geq (1/0.2) \cdot (0.95 - 1/w - 0.8) \geq 0.7,$$

by our lower bound on the probability in Eq (9.6) and for $w \geq 100$. This probability is taken on average over choices of $V_i^\times(\tilde{S} \cup \tilde{T})$ for $i \in [d-1]$, so there must exist a choice of these sets $V_i^\times(\tilde{S} \cup \tilde{T})$ for all $i \in [d-1]$, so that conditioned on these choices, protocol π still succeeds with probability 0.7. These choices are fixed for part (iii) of the lemma statement.

We know from definition of distribution $\tilde{\mu}_{\mathtt{PC}}$ that $\mathsf{b} = 0$ if and only if $\textsc{last}_{G^\times}((1,1)) \in T$ and $\mathsf{b} = 1$ if and only if $\textsc{last}_{G^\times}((1,1)) \in S$. This proves part (iii) of the lemma, as success of protocol π, when it finds the value of b correctly, also identifies whether $\textsc{last}_{G^\times}((1,1))$ belongs to $\tilde{S}$ or $\tilde{T}$.

Part (iv) of the lemma follows directly from $\tilde{S} \cup \tilde{T}$ being a rook set and Claim 9.3.14. ∎

We have thus obtained a protocol that finds whether the final pointer $\textsc{last}_{G^\times}((1,1))$ belongs to $\tilde{S}$ or $\tilde{T}$ when the matchings are chosen uniformly at random and independently. We are done with the proof of Lemma 9.3.7 if we can prove a lower bound for $\mathtt{PC}_{d,w}$ under the uniform distribution.

Let $\nu_{\mathtt{PC}}$ be the input distribution to $\mathtt{PC}$ where all the matchings of given G are chosen uniformly at random and independently of each other. When the input distribution is $\nu_{\mathtt{PC}}$, we have the following.

Lemma 9.3.19. *For any integers $k, \ell \geq 2$, any deterministic protocol which outputs the answer for $\mathtt{PC}$ under uniform distribution $\nu_{\mathtt{PC}}$ with probability of success at least 0.7 must have communication at least $\Omega(\ell/k^5 - \log \ell)$.*

The difference between standard pointer chasing and our version is that the functions are permutations, and therefore the independence between the values taken by each element of the domain is lost. This version of pointer chasing has appeared in prior works before, [AKSY20, AN21], although not explicitly in the format we want, so a proof can be found in the full version [AHS26] for completeness. The techniques are largely borrowed from [AN21] in how they handle permutations.

We can easily prove Lemma 9.3.7 using Lemma 9.3.18 and Lemma 9.3.19.

Proof of Lemma 9.3.7. We know from part (iii) that protocol π succeeds on the following instance of $\mathtt{PC}_{d,0.2w}$:

- The vertex layers are $V_i^\times(\tilde{S} \cup \tilde{T})$ for $i \in [d]$.
- The final pointer belongs to set $V_d^\times(\tilde{S} \cup \tilde{T}) = \tilde{S} \cup \tilde{T}$.
- The vertex partition in the final layer is $\tilde{S}$ and $\tilde{T}$.
- In each layer, the edges are perfect matchings $M_i^\times$ from $V_i^\times$ to $V_{i+1}^\times$ for $i \in [d-1]$.

The matchings are chosen uniformly at random and independently from part (iv) of Lemma 9.3.18. This is the same as distribution ν for $\mathtt{PC}_{k,\ell}$ with $k = d$ and $\ell = 0.2w$. By Lemma 9.3.19, we get directly that protocol π must use communication at least $\Omega(\ell/k^5 - \log \ell) = \Omega(w/d^5 - \log w)$.
∎

10 Streaming Set Cover

10.1 Introduction

Set cover is a standard optimization problem with a wide range of applications [GW97, Mih99, GKK+08, SG09, CKT10]. There is a universe $\mathcal{U}$ of size n, and there is a family $\mathcal{S}$ containing m subsets $S_1, \ldots, S_m \subseteq \mathcal{U}$. The aim is to find a collection $\mathcal{C} \subseteq \mathcal{S}$ such that the union of all the subsets in collection $\mathcal{C}$ covers the entire universe $\mathcal{U}$, while minimizing the size of the collection $\mathcal{C}$. There exist multiple algorithms that achieve an approximation ratio of $\ln n$ [Chv79, Joh74, Lov75], and we also know that this ratio is the best possible in polynomial time if P $\neq$ NP [Fei98, DS14].

In recent years, there has also been extensive interest in studying set cover in the streaming model. Here, the sets arrive one by one in some arbitrary order in a stream. The algorithm is allowed to make one or few passes over the stream before outputting a solution. Storing all the sets explicitly to process the set cover instance needs $\Theta(mn)$ space, but the goal is to find a solution in $o(mn)$ space. As is standard in the streaming model, we primarily focus on the space complexity of the algorithms and do not necessarily bound their runtime. The study of set cover in streaming was initiated by [SG09], and since then has been extensively explored in many works [DIMV14, ER16, CW16, HIMV16, AKL16, Ass17, IMR+17, BEM17, KK22, KKA23].

The work in this area can be broadly classified as having two main focuses: using as little space as possible, or getting as good an approximation as possible.

Low Space. The goal in the low-space regime is to use $\widetilde{O}(n)$ space[1] and minimize the approximation ratio and number of passes as much as possible.[2] [SG09] initiated this line of study and obtained an $O(\log n)$-approximation in $O(\log n)$-passes. [ER16] considered single-pass algorithms and showed a tight bound of $\Theta(\sqrt{n})$ on the approximation ratio. Finally, [CW16] showed a smooth trade-off between the results of [ER16] and [SG09]: for any $p \geq 1$, they gave an algorithm that obtains an $\widetilde{O}(n^{1/(p+1)})$-approximation in p-passes and showed that obtaining $\widetilde{\Omega}(n^{1/(p+1)})$ approximation requires $\Omega(n^c)$ space for some constant $c > 1$, fully settling the complexity of the streaming set cover problem in the low space regime.

Low Approximation. Another line of work—which is the main focus of our work—aims to match the performance of non-streaming algorithms in terms of approximation ratio while using sublinear space, although not necessarily as restrictive as $\widetilde{O}(n)$ space. Typically, these results either use polynomial time to get $O(\log n)$-approximations or exponential time to get $O(1)$-approximations. Here, we often assume that m is some (large) polynomial in n.

The first result in this regime—although not studied directly for streaming algorithms—is the work of [Nis02] who proved that communication complexity of getting a better than a

[1] Throughout this paper, $\widetilde{\Omega}$ and $\widetilde{O}$ hide logarithmic factors in m, n.

[2] Space of $O(n)$ is necessary to even certify whether the input set cover instance is feasible.

$(\log n/2)$-approximation to set cover is $\Omega(m)$ bits. This translates to a lower bound of $\Omega(m/p)$ space for p-pass algorithms and same approximation ratio using the standard connection between streaming algorithms and communication protocols (see, e.g. [AMS96]).

Algorithmically, [DIMV14] initiated this line of study and obtained an $O(p)$-approximation in p-passes and $\widetilde{O}(mn^{1/O(\log(p))})$-space. This space bound was later improved by [HIMV16] to $\widetilde{O}(mn^{1/p})$ space. For polynomial-time algorithms, this result led to an $O(p \log n)$-approximation which was improved to $O(\log n)$-approximation in [BEM17] and even $(1+o(1)) \cdot \ln n$ approximation in [IMR+17] at a cost of increasing the number of passes by a constant factor. This constitutes the state-of-the-art algorithms for this problem.

The tradeoff curve obtained by algorithms of [HIMV16, BEM17, IMR+17] namely, $mn^{1/p}$ space and p passes, is a familiar one in the streaming model. Based on this, [HIMV16] conjectured that:

> *"... this tradeoff might be tight, at least for 'low enough' approximation factors".*

In the same work, [HIMV16] provided an evidence in favor of this conjecture: finding an *exact* set cover, i.e., a 1-approximation, requires $\tilde{\Omega}(mn^{1/2p})$ space in p-passes. It turns out however that this evidence is not that strong due to the *three-way* interplay between the resource parameters of the problem, space, pass, and approximation. Specifically, [Ass17] showed that finding an exact set cover in p passes requires $\Omega(mn/p)$ space—and not just $\Omega(mn^{1/p})$ proven by [HIMV16]—and more generally $\Omega(mn^{1/\alpha}/p)$ space for α-approximation when $\alpha = o(\log n)$. The bounds of [Ass17] identifies the optimal *space-approximation* tradeoff for this problem when we *ignore* the number of passes (as long as it is polylog(n)); but it does not address the original conjecture (only weakens the evidence of [HIMV16]).

Focusing on the original goal of low space algorithms—matching the performance of non-streaming algorithms, namely, a $\log n$ or even a polylog(n)-approximation—and the conjecture of [HIMV16], we do not yet have any conclusive answer. On one hand, the algorithmic progress on this question has halted for nearly a decade now. On the other hand, even considering all known lower bounds [Nis02, ER16, DIMV14, AKL16, CW16, HIMV16, Ass17], it is still entirely plausible that there is even a two-pass algorithm with $O(m)$ space and $O(\log n)$-approximation!

10.1.0.1 Our Contributions

We settle the conjecture of [HIMV16] by proving that any $\log n$ or even polylog(n)-approximation algorithm for streaming set cover requires $\widetilde{\Omega}(mn^{1/p})$ space in p-passes. In fact, we prove a much more general result that establishes a (nearly) *optimal* three-way space-pass-approximation tradeoff:

Theorem 8 (restated). *For any $p, \alpha, m, n \geq 1$, any p-pass algorithm that given a set cover instance over universe $[n]$ with m input sets outputs an α-approximation to the optimal set cover* size *(not necessarily even finding the sets) with probability at least $2/3$ requires space*

$$\tilde{\Omega}\left(\frac{m}{\alpha} \cdot \left(\frac{n}{\alpha}\right)^{1/p}\right).$$

The lower bound holds as long as $p \ll \dfrac{\log n}{\log\log n}$ *and* $p \le \alpha \ll \left(\dfrac{1}{p} \cdot (\dfrac{n}{\log m})^{1/(p+1)}\right)$.

Several remarks are in order (see also Tables 10.1 and 10.2),

- The tradeoff in Theorem 8 is nearly optimal for all involved parameters in the light of existing results, in particular, a simple modification of the algorithms in [HIMV16, AKL16].
- The ranges of parameters in Theorem 8 are all necessary and nearly optimal. For any $p = \Omega(\log n)$, we can obtain an $O(\log n)$ approximation in $\widetilde{O}(n)$ space and $O(\log n)$ passes [SG09, CW16]. When $\alpha < p$, the space requirement "jumps" to $\Omega(mn^{1/\alpha}/p)$ by the lower bound of [Ass17]. When $\alpha \ge n^{1/(p+1)}$, the space requirement "drops" to $\widetilde{O}(n)$ by the result of [CW16].
- Theorem 8 generalizes the *single-pass* lower bounds of [AKL16] who proved an $\Omega(mn/\alpha^2)$ space lower bound (matching our tradeoff for $p = 1$) to all choices of $p \ll \log n$.
- Theorem 8 also generalizes the *multi-pass* lower bounds of [CW16] by proving that $\Omega(m)$ space is needed for $(n/\log n)^{1/(p+1)}$ approximation, improving upon their $\Omega(m^{1/p})$ space lower bound (at a cost of a $(\log n)^{1/(p+1)}$ loss in our approximation ratio).

Citation	Space	Passes	Approximation	Range of Parameters
Low Space Regime				
[ER16]	$\Omega(m)$	1	$\sqrt{n}$	
[CW16]	$\widetilde{\Omega}(m^{1/p})$	p	$n^{1/(p+1)}$	$p \ll \frac{\log n}{\log\log n}$
Low Approximation Regime				
[AKL16]	$\Omega(mn/\alpha^2)$	1	α	$\alpha \ll \sqrt{n}$
[Nis02]	$\Omega(m/p)$	p	$(\log n)/2$	
[HIMV16]	$\Omega(mn^{1/p})$	p	1	$p \ll \frac{\log n}{\log\log n}$
[Ass17]	$\Omega(mn^{1/\alpha}/p)$	p	α	$\alpha \ll \log n$
This work	$\widetilde{\Omega}(mn^{1/p})$	p	polylog(n)	$p \ll \frac{\log n}{\log\log n}$
This work	$\widetilde{\Omega}\left(\frac{m}{\alpha} \cdot (\frac{n}{\alpha})^{1/p}\right)$	p	α	$p \ll \frac{\log n}{\log\log n}$, $p \le \alpha \ll n^{1/(p+1)}$

Table 10.1: A summary of prior lower bounds for streaming set cover and our work.

Citation	Space	Passes	Approximation
[CW16]	$\widetilde{O}(n)$	p	$n^{1/(p+1)}$
[HIMV16]	$\widetilde{O}(mn^{1/p})$	p	p
[IMR+17]	$\widetilde{O}(mn^{1/p})$	$O(p)$	$\ln n$
[HIMV16] + [AKL16]	$\widetilde{O}\left(\frac{m}{\alpha}\cdot\left(\frac{n}{\alpha}\right)^{1/p}\right)$	p	α

Table 10.2: The state-of-the-art multi-pass algorithms for streaming set cover. The final line is not stated explicitly in prior work but follows from a combination of [HIMV16, AKL16] and proves the optimality of our lower bound. The upper bound on α in our lower bound is also optimal in light of the algorithm in the first line.

10.1.0.2 Our Techniques

The closest work to us is the previous streaming set cover lower bound of [CW16] in the low space regime. To prove a p-pass lower bound, they start with "many" $(p-1)$-pass hard instances on different universes with a "small" overlap between them. The stream consists of these sets plus a single set at the end that covers all elements except for the universe of one randomly chosen *special* $(p-1)$-pass hard instance. The small overlap between the sets implies that getting a good approximation to the entire instance requires one to solve the special sub-instance also. The space lower bound is proven by arguing that the number of embedded $(p-1)$-pass instances is so much larger than the memory of the algorithm, that it is unlikely in the first pass, the algorithm remembers anything useful about the special one; thus, it needs to solve a hard $(p-1)$-pass instance in the remaining $(p-1)$ passes which, by induction, is not possible. This argument is formalized in [CW16] via a reduction from the *tree pointer jumping* problem in communication complexity [CCM08] and introducing an extremal family of small overlapping sets, named *edifices*, through well-structured algebraic geometry constructs [Bue79].

The arguments in [CW16] are heavily tailored to the low space regime: as argued above, the *number* of underlying $(p-1)$-pass instances needs to be more than the memory of the algorithm, which leads to an exponential growth in the number of inputs sets across the passes. In terms of m, this only leads to an $\Omega(m^{1/p})$ space lower bound, which is way below the $\omega(m)$ bounds one needs in the low approximation regime. Similar to [CW16], the techniques in other graph streaming lower bound results [AKNS24a, ABK+25][3] rely on using *separate* $(p-1)$-pass hard sub-instances (in terms of the sets in those instances) and this makes it inapplicable to prove Theorem 8.

Our approach. We take a different approach when proving Theorem 8. For simplicity, let us focus on the case when $\alpha \approx \log n$. Our p-pass hard instances are also constructed

[3]Streaming set cover is equivalent to dominating set in graph streaming. [CW16] gives an $\Omega(n^{1/p})$ space lower bound for n-vertex graphs whereas one ideally would like an $\Omega(n^{1+1/p})$-space lower bound in graph streams.

recursively from $(p-1)$-pass hard instances, but not so directly as in [CW16]. A p-pass hard instance consists of a *single* $(p-1)$-pass hard instance with an *unknown* universe; instead, it will be revealed to the algorithm at the end of the first pass—by our construction of the instance—as essentially a *random subset* of size $\approx n^{1-1/p}$ from the original universe (once the universe is revealed, the $(p-1)$-pass instance simply becomes the intersection of the original sets with this new universe).

Unlike in [CW16] (and [AKNS24a, ABK+25]), there is no hope in proving that the algorithm does not learn *any* information about the underlying $(p-1)$-pass hard instance (given it is just one instance). But, we show that the algorithm does not learn anything *useful* about this underlying instance. This is achieved through a careful construction of the input instances as sets of size $\approx n/\log n$ that are *marginally* uniformly random, but have a mild *correlation* in a way that each set "points to" another one; the union of these two sets covers (considerably) more elements than two independent random sets. This correlation is then used to argue that to get a good approximation, we should start from some set—whose identity is forced to us at the end of the stream—and then continue picking the sets "pointed to" by the previous one. At the same time, this correlation is low enough that until the algorithm has not identified the universe of the underlying $(p-1)$-hard instance, its chance of revealing useful information about the *right* set of pointers is negligible[4].

To formalize our result, we use the existing lower bounds for the *pointer chasing* problem introduced by [PS84], which has since became a staple in proving multi-pass streaming lower bounds of similar nature, e.g., in [GO16, AKSY20, CKP+21a, AGL+24]. Our reduction from pointer chasing is fairly direct and we rely on elementary properties of random sets to carry out the analysis.

We conclude this part by remarking that the dynamics of our hard instances, namely, reducing the problem to another instance on $\approx n^{1-1/p}$ elements after the first pass, is exactly what streaming set cover algorithms in [HIMV16, BEM17, Ass17, IMR+17] achieve through their *element sampling technique*. Indeed, our lower bound construction is directly inspired by these algorithms.

10.2 Preliminaries

For the cartesian product of two sets $[c] \times [d]$, we use (i, j) to denote the elements in the set for $i \in [c]$ and $j \in [d]$.

Pointer chasing. We now formally define the pointer chasing problem used for the result; see also Figure 10.1. This is the definition in the multi-party setting that has been used in multiple results in the past, e.g., [GO16, AN21], although many variants exist.

Definition 10.2.1. *For any integers $p, a, b \geq 1$, the pointer chasing problem $\mathsf{PC}_{p,a,b}$ is defined as:*

- *There are $p+1$ players $P^1, P^2, \ldots, P^{p+1}$.*

[4] To compare this with [CW16], for them one only needs to reveal $\log n$ bits about a $(p-1)$-pass instance in the first pass to make it no longer hard in the remaining $(p-1)$ passes; but there are so many such instances that the algorithm cannot reveal $\log n$ bits for each of them. On the other hand, to break our underlying $(p-1)$-pass instance *before* its universe of size $n^{1-1/p}$ is revealed, one still needs to reveal $\Omega(mn^{1/p})$ information.

- *P^1 has a balanced Boolean function $f^1 : [a] \times [b] \to \{0,1\}$.*[5]
- *For each $1 < i \leq (p+1)$, P^i has a function $f^i : [a] \times [b] \to [a] \times [b]$.*
- *In one round, they talk in the order of $P^1, P^2, \ldots, P^{p+1}$ on a blackboard that everyone can see. They talk for a total of p rounds.*
- *For each $i \in [p+1]$, we define the **special pointer** $z_i \in [a] \times [b]$ recursively as:*

$$z_{p+1} = (1,1), \text{ and for } 1 \leq i < (p+1),\ z_i = f^{i+1}(z_{i+1}).$$

- *At the end of all the rounds, the last player P^{p+1} has to output the value of $f^1(z_1)$ which is a Boolean value, based on their input and the contents of the blackboard.*

See Figure 10.1 for an illustration.

We know the following lower bound on the communication complexity of $\mathsf{PC}_{p,a,b}$.

Proposition 10.2.2 ([AN21])**.** *For any $p \geq 1$, any randomized protocol π that solves $\mathsf{PC}_{p,a,b}$ with probability at least $3/5$ has the total length of messages on the board to be at least*

$$\Omega\left(\frac{a \cdot b}{p^5}\right).$$

The work of [AN21] proves a lower bound for pointer chasing when the functions given to each player are permutations, but this is a special case of our problem, and hence the lower bound holds for our version of pointer chasing also.

10.3 The Lower Bound

In this section, we prove Theorem 8 in its entirety. We employ the impossibility of the pointer chasing problem stated in Proposition 10.2.2. We start with a protocol for pointer chasing that uses an algorithm that approximately estimates the set cover. We describe the protocol in three sequential parts. Finally, we prove the correctness of our protocol, and use Proposition 10.2.2 to prove the required space lower bound for set cover.

10.3.0.1 Protocol for Pointer Chasing

In this section, we see how to create a protocol π for $\mathsf{PC}_{p,a,b}$ using an algorithm $\mathcal{A}$ for α-approximation of set cover in p passes over a universe $\mathcal{U}$ of size n and totally m sets. The parameters a and b are chosen based on m, n, α and p as follows.

Parameters:

$$a = \frac{m-1}{p} \qquad \text{and} \qquad b = \frac{1}{p+1} \cdot \left(\frac{n}{10(p+1)\log m}\right)^{1/p} \cdot \frac{1}{\alpha^{(p+1)/p}}. \tag{10.1}$$

[5] By balanced, we mean number of inputs mapping to 0 and number of inputs mapping to 1 are equal.

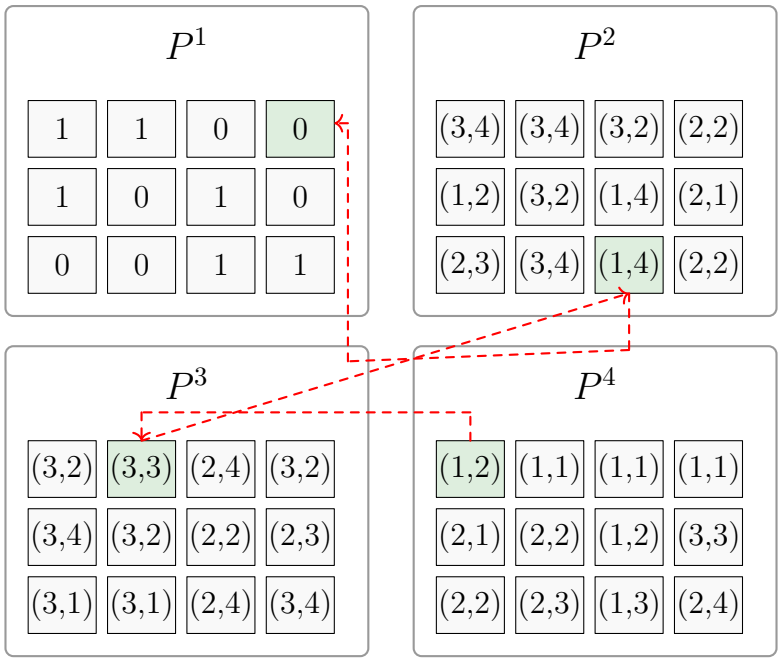


Figure 10.1: An illustration of pointer chasing for $p = 3, a = 3$ and $b = 4$. Each function is represented by a 3×4 grid containing the corresponding values. The special pointers of each player are highlighted in green. The red dashed arrows represent how the pointers move. The answer to the pointer chasing instance is 0.

Our reduction crucially uses public randomness to create the sets that the set cover algorithm is run on, and does not use any private randomness. We break the protocol into three simple parts.

We want to create a sets for each player P^i for $i \in [p]$ so that each of the a sets corresponds to b many function outputs held by P^i. Since only one function output of P^{p+1} matters, P^{p+1} only creates one set. For P^i with $i \in [p]$, the $a \cdot b$ function values are grouped together in the natural way: P^i creates one set S^i_j for each $j \in [a]$ that depends only on $f^i(j, k)$ for all $k \in [b]$. To continue, we first create many auxiliary sets called the **U-common family** with public randomness as follows.

Construction of the U-common family:

The first part of the protocol for pointer chasing involves the creation of many auxiliary sets using public randomness, and we describe this next.

Part (1) **Protocol** π **for** $\mathsf{PC}_{p,a,b}$**:**
Each player P^i for $i \in [p]$ samples the following using public randomness:

(i) For each $j \in [a]$, sample a set U^i_j uniformly at random of size $n/(10\alpha \cdot (p+1))$ from $[n]$, independently of all the other sets.

(*ii*) Randomly partition U_j^i into b sets, each of size $n/(10\alpha \cdot b \cdot (p+1))$ using public randomness, called $U_{j,k}^i$ for $k \in [b]$.

Refer to Figure 10.2 for an illustration.

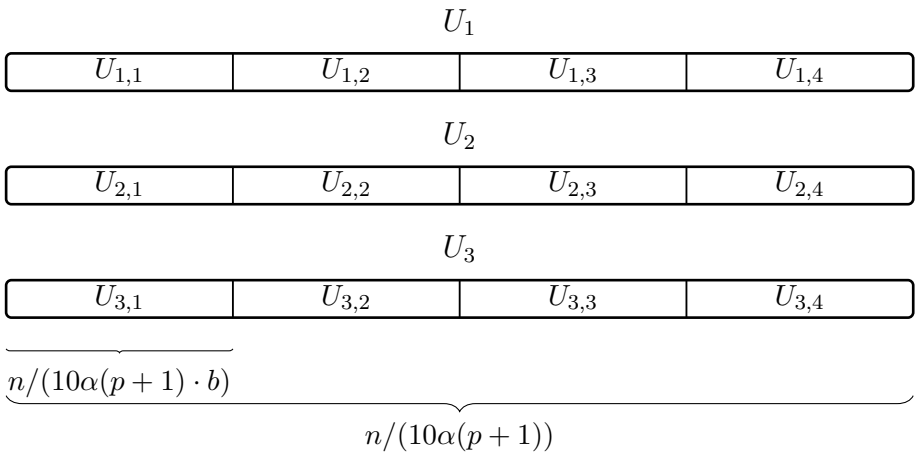


Figure 10.2: Creation of U-common family for each player: this is repeated for all $P^1, P^2, P^3, \ldots, P^p$. The illustration is when $a = 3$ and $b = 4$. There are three sets U_1, U_2 and U_3 each of size $n/(10\alpha \cdot (p+1))$ chosen uniformly at random. These are split into four sets, each of size $n/(10\alpha \cdot b \cdot (p+1)) = n/(40\alpha \cdot (p+1))$. Although the sets U_1, U_2 and U_3 are shown as disjoint, they may have common elements as they are sampled independently. However, for each $i \in [3]$, the sets $U_{i,j}$ for $j \in [4]$ are disjoint from each other.

We will prove some simple properties of the sets in the U-common family. Mainly, we want to look at the intersection of any p sets with one set each picked from the first p players. Although the below statement applies to the sets corresponding to the special pointers in Definition 10.2.1, it holds for any collection of p sets with the aforementioned property.

Claim 10.3.1. *The set $\bigcap_{i\in[p]} U_{z_i}^i$ has size at least $4\alpha \cdot (p+1) \cdot \log(m)$ with high probability over the randomness of the choices of all the sets $U_{j,k}^i$ for $i \in [p+1], j \in [a], k \in [b]$.*

Proof. Each element is chosen to be in set $U_{z_i}^i$ with probablity exactly $1/(\alpha \cdot b \cdot (p+1))$ as the set contains $n/(\alpha \cdot b \cdot (p+1))$ elements chosen uniformly at random. Moreover, the choices of $U_{z_i}^i$ and $U_{z_{i'}}^{i'}$ for $i \neq i'$ are independent of each other.

Hence, for any element e, the probability that it belongs to $U_{z_i}^i$ for all $i \in [p]$ is:

$$\Pr[e \in \cap_{i\in[p]} U_{z_i}^i] = \left(\frac{1}{\alpha \cdot b \cdot (p+1)}\right)^p = \frac{1}{\alpha^p} \cdot \left(\frac{10(p+1)\log m}{n}\right) \cdot \alpha^{p+1} = \frac{1}{n} \cdot (10\alpha \cdot (p+1) \cdot \log m),$$

by the value of b from Eq (10.1).

Let T denote the intersection of all the sets $U^i_{z_i}$ for $i \in [p]$. For any element $e \in \mathcal{U}$, let X_e denote the indicator random variable for whether element e is in T. We know that the X_e random variables are negatively correlated, as $U^i_{z_i}$ consists of exactly $n/(10\alpha \cdot b \cdot (p+1))$ elements. Using Proposition 4.2.3, we get,

$$\begin{aligned}
\Pr[|T| < 4\alpha(p+1) \cdot \log m] &= \Pr[\sum_e X_e < 4\alpha(p+1) \cdot (\log m)] \\
&= \Pr[\sum_e X_e < (4/10)\,\mathbb{E}[\sum_e X_e]] \quad \text{(by linearity of expectation)} \\
&\leq \exp(-0.18 \cdot 10\alpha \cdot (p+1) \log m) \quad \text{(by Proposition 4.2.3)} \\
&\leq m^{-\alpha \cdot (p+1)},
\end{aligned}$$

which proves the claim, as $\alpha \geq 1$ and for large m. ∎

Creation of Set Cover Instance:

The next part of the protocol is to create the sets that the algorithm $\mathcal{A}$ is run on. Each players only use the U-common sets and their own pointers in the pointer chasing instance to create the set cover instance. This step is performed deterministically by all the players. The sets will be collectively called the **S-private family**.

Part (2) Protocol π for $\mathsf{PC}_{p,a,b}$:
Input: The U-common sets from Part (1) of π, and $\mathsf{PC}_{p,a,b}$ instance.
Output: Set cover instance.

(*i*) The last player P^{p+1} constructs one input set S^{p+1} as follows:

$$S^{p+1} := [n] \setminus U^p_{f^{p+1}(1,1)}.$$

(*ii*) Each other player P^i for $1 < i \leq p$ constructs a sets for $j \in [a]$ as follows:

$$S^i_j := \bigcup_{k \in [b]} U^i_{j,k} \setminus U^{i-1}_{f^i(j,k)}.$$

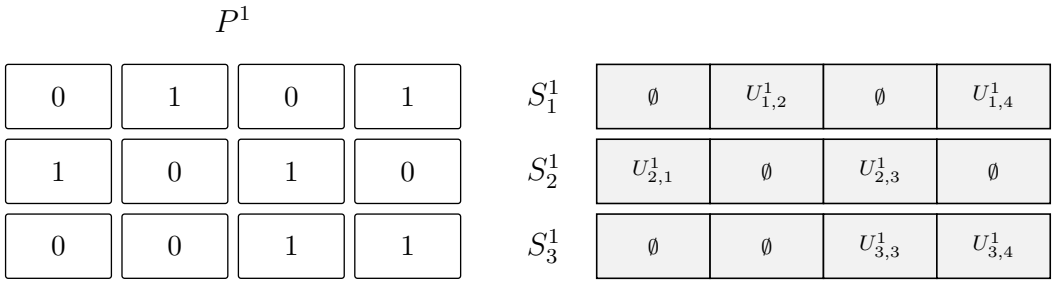
(*iii*) The first player P^1 constructs a input sets S^1_j for $j \in [a]$ as follows:

$$S^1_j := \bigcup_{\substack{k \in [b]; \\ f^1(j,k)=1}} U^1_{j,k}.$$

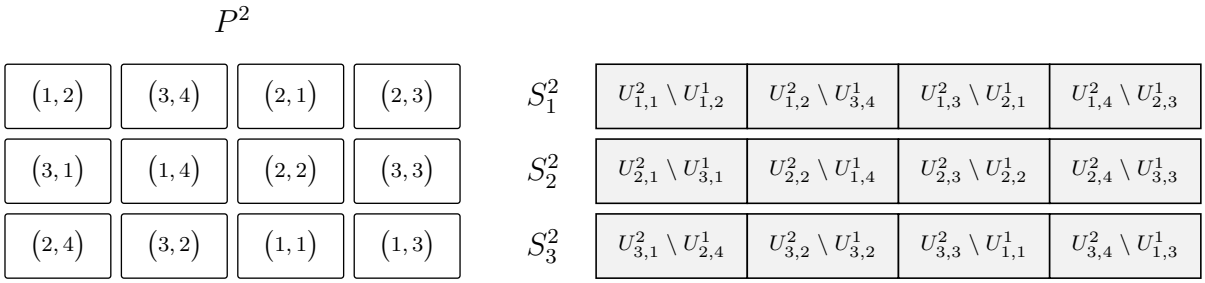
Refer to Figure 10.3 for an illustration.

Again, we prove some simple properties about the set cover instance.

Observation 10.3.2. *There are exactly m sets created by all the $p+1$ players together.*



(a) The sets for P^1.



(b) The sets for P^2.

Figure 10.3: An illustration of the S-private family for players P^1 and P^2 with $a = 3, b = 4$. Each S_j^i for $i \in [2], j \in [3]$ is a union of all the sets to its right.

Proof. Each player P^i for $i \in [p]$ creates a sets and the last player P^{p+1} creates one set. Together, these are $a \cdot p + 1 = m$ sets by Eq (10.1). ∎

The main lemma of this subsection follows. This states a gap in the values of the optimal set cover based on the answer of the pointer chasing instance.

Lemma 10.3.3. *When the answer to the pointer chasing instance is* 1*, there are* $(p+1)$ *sets among* m *sets that cover the entire universe* $[n]$*. Otherwise, with high probability, any collection of* $\alpha \cdot (p+1)$ *sets among the* m *sets does not cover the universe* $[n]$*.*

Before we prove Lemma 10.3.3, we establish some notation. For $i \in [p]$, we use $S_\star^i$ to denote the set S_j^i such that $z_i = (j, k)$ for some $k \in [b]$. These are the sets in the S-private family that depend on the special pointers given to the first p players. We also use $S_\star^{p+1}$ to denote the set S^{p+1}.

First, let us see what happens when the answer to the pointer chasing instance is 1. The following fairly direct claim states that the size of the set cover instance is small, and this happens just by picking all $S_\star^i$ for $i \in [p+1]$.

Claim 10.3.4. *When the answer to the pointer chasing problem is* 1*, the set cover instance constructed in protocol* π*, has a set cover of size at most* $(p+1)$*.*

Proof. We know that the answer to the pointer chasing instance is $f^1(z_1)$. We will show that :

$$\bigcup_{i \in [p+1]} S_\star^i = [n],$$

which proves the claim.

We have that $S_\star^{p+1} = [n] \setminus U_{z_p}^p$ by construction so we need to show that $U_{z_p}^p$ is covered by the remaining sets. We will show by induction on i for $i \in [p]$, that when $f^1(z_1) = 1$, the sets $S_\star^1, S_\star^2, \ldots, S_\star^i$ cover all the elements in $U_{z_i}^i$, which will complete the proof. This will ensure that all elements in $U_{z_p}^p$ will be covered by the union of $S_\star^1, \ldots, S_\star^p$ when $i = p$.

For the base case, we know directly that $S_\star^1 = U_{z_1}^1$, as $f^1(z_1) = 1$. Assume towards the induction hypothesis that $S_\star^1, \ldots, S_\star^{i-1}$ covers all of $U_{z_{i-1}}^{i-1}$. Then, the set $S_\star^i$ contains as a subset the set $U_{z_i}^i \setminus U_{z_{i-1}}^{i-1}$, and hence the union of $S_\star^1, \ldots, S_\star^i$ covers all the remaining elements of the set $U_{z_i}^i$ by construction. ∎

Now, we analyze what happens when the answer to the pointer chasing instance is 0. The intersection of all $U_{z_i}^i$ for $i \in [p]$, which are the sets corersponding to special pointers, will be large by Claim 10.3.1, and we will show that less than $\alpha \cdot (p+1)$ sets from the S-private family will not be able to cover all these elements.

Claim 10.3.5. *When the answer to pointer chasing is* 0*, with high probability no collection of* $\alpha \cdot (p+1)$ *sets* $T_1, T_2, \ldots, T_{\alpha\cdot(p+1)}$ *among the* m *sets in S-private family covers the entire universe* $[n]$*.*

Proof. We will show specifically that the elements of $\bigcap_{i\in[p]} U_{z_i}^i$ will not be covered by any collection of $\alpha \cdot (p+1)$ sets from the S-private family. Let T be the set $\bigcap_{i\in[p]} U_{z_i}^i$. For each $i \in [p+1]$, let $j_i^\star \in [a]$ be the index such that $z_i = (j_i^\star, k)$ for some $k \in [b]$, i.e., $j^\star$ corresponds to the first part of the special pointer. Notice also that $S_\star^i = S_{j_i^\star}^i$ for each $i \in [p]$.

We argue that $S_\star^1$ does not contain any element of set T first using the fact that $T \subseteq U_{z_1}^1$. Let $z_1 = (j_1^\star, k_1)$, and observe that $U_{j^\star,k'}^1$ is disjoint from $U_{j_1^\star,k_1}^1$ for any $k' \neq k_1, k' \in [b]$ by construction. Moreover, the set $U_{z_1}^1 = U_{j_1^\star,k_1}^1$ is explicitly excluded from $S_{j_1^\star}^1$ as the answer to the pointer chasing instance is 0. Hence, $S_\star^1 = S_{j_1^\star}^1$ is a subset of $U_{j_1^\star}^1 \setminus U_{z_1}^1$, and does not contain any element of $U_{z_1}^1$. Similarly, we can argue that $S_\star^i$ for any $i \in [p]$ does not contain any element of T as $T \subset U_{z_i}^i$. Again, as $T \subset U_{z_p}^p$, the set $S_\star^{p+1}$ does not cover any element of T either.

It remains to argue about any collection of $\alpha \cdot (p+1)$ sets from the S-private family excluding all $S_\star^i$ for $i \in [p+1]$. To do this, we condition on the event that the set T has at least $4\alpha(p+1) \cdot \log m$ elements. This event happens with high probability by Claim 10.3.1. For the rest of this proof, we do not use any randomness associated with any $U_{j_i^\star}^i$ for any $i \in [p]$. This randomness is only used to condition on the size of the set T.

We will prove the stronger statement that any collection of $\alpha \cdot (p+1)$ sets U_j^i for $i \in [p], j \in [a] \setminus \{j_i^\star\}$ does not cover T, as $S_j^i \subseteq U_j^i$ for all $i \in [p], j \in [a]$. We know that each U_j^i is a set of $n/10\alpha \cdot (p+1)$ elements chosen at random, independently of each other and of T. We will use Chernoff bounds to argue that the probability that they cover the entirety of set T is small. Fix any choice of $\alpha(p+1)$ distinct sets U_j^i for $i \in [p], j \in [a] \setminus \{j_i^\star\}$.

Choose any $4\alpha(p+1) \cdot (\log m)$ elements of T. These many distinct elements exist in T because of our earlier conditioning. Let $X_{e,i}$ denote the indicator random variable for whether e is covered by the i^{th} set for some e in the subset of $4\alpha(p+1) \cdot (\log m)$ elements of

T and, $i \in [\alpha \cdot (p+1)]$. We have that:

$$\mathbb{E}[X_{e,i}] = \frac{1}{10\alpha \cdot (p+1)},$$

as the i^{th} set chooses $n/(10\alpha \cdot (p+1))$ elements at random independently of T. The expectation of the sum of these random variables is:

$$\mathbb{E}[X] = \mathbb{E}[\sum_{e,i} X_{e,i}] = 4\alpha(p+1) \cdot \log m \cdot \alpha \cdot (p+1) \cdot \frac{1}{10\alpha(p+1)} = 4\alpha \cdot (\log m) \cdot (p+1)/10, \tag{10.2}$$

by linearity of expectation. We also have,

$$\begin{aligned}
&\Pr[T \text{ is covered by the collection}] \\
&\leq \Pr[X \geq 4\alpha(p+1) \cdot \log m] && \text{(as this subset of elements of } T \text{ must be covered)} \\
&\leq \Pr[X > 10 \cdot \mathbb{E}[X]] && \text{(by Eq (10.2))} \\
&\leq \exp(-2\alpha \cdot (p+1) \cdot (\log m)). \\
& && \text{(by Proposition 4.2.3, and as the variables are negatively correlated)}
\end{aligned}$$

There are at most $\binom{m}{\alpha(p+1)}$ many $\alpha(p+1)$-size collection from the m sets we are considering. A union bound over all such collections gives that with probability at most $m^{-\alpha(p+1)}$, the set T is covered by some α-size collection. For large m and $\alpha \geq 1$, this probability is small, and thus the proof of the claim is complete. ∎

The proof of Lemma 10.3.3 now follows directly from Claim 10.3.5 and Claim 10.3.4.

Running the streaming algorithm $\mathcal{A}$:

The final part in the construction of π is to actually run $\mathcal{A}$ and output an answer. This part is straightforward.

Part (3) **of protocol π:**
Input: Pointer Chasing instance $\mathsf{PC}_{p,a,b}$ and a p-pass α-approximation algorithm $\mathcal{A}$ for set cover over a universe $[n]$, taking m input sets.
Output: Answer to pointer chasing instance.

(*i*) The players run $\mathcal{A}$ on the $m = a \cdot (p-1) + 1$ input sets S^p and S^i_j for $i \in [p], j \in [a]$ over the universe $[n]$ by sending messages to the blackboard. (See Claim 10.3.6 for how to run $\mathcal{A}$.)

(*ii*) If $\mathcal{A}$ returns a value $\leq \alpha \cdot (p+1)$ as an estimate of the size of the optimal set cover, P^p outputs that the value of the pointer chasing instance is 1. Otherwise, P^p outputs 0.

First, we show that the players can run $\mathcal{A}$ effectively.

Claim 10.3.6. *Algorithm $\mathcal{A}$ that uses p-passes and s space can be run by the $(p+1)$ players using total communcation $(p+1)\cdot p\cdot s$ total communication and p rounds.*

Proof. Using the sets $U^i_{j,k}$ sampled with public randomness for $i \in [p], j \in [a], k \in [b]$, we have:

- P^i knows all the sets S^i_j for $i \in [p], j \in [a]$ as they depend only on the publicly chosen U-common family, and the pointers given to P^i.
- P^{p+1} knows the set S^{p+1} as it depends on the publicly sampled U-common family, and the value of $f^p(1,1)$, that P^p knows.

Hence, collectively the players can run one pass of $\mathcal{A}$ starting from P^1 who inserts the sets S^1_j for $j \in [a]$, and sends the memory state of $\mathcal{A}$ to the blackboard in s bits. This is followed by P^2 who inserts S^2_j for all $j \in [a]$, all the way till P^{p+1} who inserts S^{p+1}. The total communication used to run one pass of $\mathcal{A}$ is $(p+1)\cdot s$ bits, and can be done in one round.

To run all the p-passes, $p\cdot(p+1)\cdot s$ bits of messages are used totally. ∎

Next, we argue the correctness of protocol π.

Claim 10.3.7. *If algorithm $\mathcal{A}$ succeeds with probability at least $2/3$, protocol π for $\mathsf{PC}_{p,a,b}$ succeeds with probability at least $3/5$.*

Proof. When the answer to pointer chasing is 1, by Lemma 10.3.3, we know that the optimal set cover is of size at most $(p+1)$. Hence, if $\mathcal{A}$ is correct, it outputs an estimate of size at most $\alpha\cdot(p+1)$, and π outputs the correct answer as well. When the answer is 0, by Lemma 10.3.3, we know that with high probability, the set cover size is more than $\alpha\cdot(p+1)$. Thus, if $\mathcal{A}$ succeeds, the estimate of the size of the set cover will be more than $\alpha\cdot(p+1)$. Protocol π succeeds with probability at least $3/5$ for large values of m, n and $\alpha \geq 1$. ∎

This concludes our description of the protocol. As we have also proved the necessary properties along the way, proving Theorem 8 now is simple, and is covered in the next subsection.

10.3.0.2 Finishing the Proof of Theorem 8

We can complete the proof of Theorem 8 using the impossibility result for pointer chasing stated in Proposition 10.2.2, and our proof of the correctness of protocol π in Claim 10.3.7.

Proof of Theorem 8. Firstly, let us examine if parameters a, b as specified in Eq (10.1) can be chosen. We have $p \ll m$, so we can choose value of a as required. We also have $p\cdot\alpha = o(n/\log m)^{1/(p+1)}$, so parameter b can also be set to the required large value. Thus, pointer chasing instances of the input size specified in Eq (10.1) can be constructed.

We know algorithm $\mathcal{A}$ uses p-passes and s-space to provide an α-approximate estimate of the solution to a set cover instance over universe $[n]$ with m sets, and succeed with probability at least $2/3$. We know that protocol π uses $(p+1)p\cdot s$ bits of communication

from Claim 10.3.6, and from Claim 10.3.7, we know it succeeds with probability at least $3/5$ for large m and $\alpha \geq 1$.

Therefore, by Proposition 10.2.2 and choice of parameters in Eq (10.1), we know that:

$$s \geq \frac{1}{(p+1)p} \cdot \Omega(\frac{ab}{p^5}) = \Omega(1/p^7) \cdot \Omega\left(\frac{(m-1)}{p} \cdot \left(\frac{n}{\alpha}\right)^{1/(p)} \cdot \frac{1}{\alpha(p+1)} \cdot \frac{1}{(10 \log m \cdot (p+1))^{1/p}}\right).$$

By simplifying the above term, we know that $\mathcal{A}$ uses space at least:

$$\Omega\left(\frac{1}{p^{10} \cdot \log m} \cdot \frac{m}{\alpha} \cdot \left(\frac{n}{\alpha}\right)^{1/p}\right),$$

which finishes the proof. ∎

11 Future Directions

In this chapter we give some open questions and further avenues of research.

11.1 Immediate Questions

We present some immediate questions left open by the results in this dissertation.

Exact Space Bounds for Each Pass. While our results in Theorem 1 and Theorem 4 give optimal asymptotic lower bounds on the number of passes in the semi-streaming setting with $\widetilde{O}(n)$ space, we do not know tight bounds for any specific number of passes.

The lower bound for MIS in Chapter 5 gives a tradeoff between the number of passes and the space used by the algorithm. It proves that in p-passes, any streaming algorithm requires $\Omega(n^{1+1/(2^p-1)+o(1)})$ space. This translates to a lower bound of $\Omega(n^2)$ for one pass algorithms [ACK19b, CDK19]. For $p = 2$, however, the lower bound is $\Omega(n^{4/3-o(1)})$, whereas the state-of-the-art algorithm that uses two passes over the edges runs in $O(n\sqrt{n})$ space [ACG+15]. While our lower bound is asymptotically optimal with respect to semi-streaming space, for any specific number of passes the correct answer is unknown.

Better Bounds for Approximate Matchings in Semi-streaming. For a $(1-\varepsilon)$-approximation of maximum matching, the lower bound we present in Theorem 3 is $\Omega(\log(1/\varepsilon))$, whereas the current upper bounds are $O(1/\varepsilon^2)$ [ALT21], or $O(\log n/\varepsilon)$ [AJJ+22] for bipartite graphs. The key bottleneck in getting a better lower bound is that, in each successive pass, the number of vertices which participate in a special hidden matching decreases by a constant factor. For any $p \geq 1$ passes, there are a set of $\approx (n/3^p)$ set of vertices among which a large induced matching must be found, to get a large approximation. This decay seems unavoidable with the existing combinatorial constructions.

Maximal Matchings in Dynamic Streams. The lower bound in Theorem 4 holds for any constant approximation even on bipartite graphs. Thus, as maximal matchings are a 2-approximation to maximum matchings, the lower bound of $\Omega(\log\log n)$ rounds remains. The current upper bound to find a maximal matching remains $O(\log n)$ passes [AGM12].

The lower bound construction for p-passes in Theorem 4 forces the algorithm to solve *a large fraction of* $(n^{1/c})$-many smaller hard instances for $(p-1)$-passes where $c > 1$ is a constant. Adding maximality will ensure that the algorithm solves *all* of these instances[1], but the polynomial factor loss in the sizes of the smaller instances remains. Hence, this framework may not lead to stronger lower bounds for maximal matchings.

Logarithmic Lower Bound for Triangle Detection. In triangle detection, as we state in Remark B.4.3, the realistic target for a lower bound is $\Omega(\log n)$ due to the connections to circuit complexity. There is still an exponential gap between our lower bound of $\Omega(\log\log n)$

[1] This is similar to Theorem 1, where an MIS of all smaller subinstances must be found.

and this target. Unlike in graph streaming lower bounds, constructing a hard instance that may exhibit an $\Omega(\log n)$ lower bound is simple.

In our construction, we pad the $(r-1)$-round instance with $\text{poly}(n_{r-1})$ many vertices to ensure the round elimination argument goes through. Even if we only pad the $(r-1)$-round instance with $d_r = (n_r)^{1/\Theta(r)}$ vertices, the instance may remain hard due to the local view of vertices. This will lead to a lower bound of:

$$s_r(n_r) = \Omega(n_1) = \Omega((n_r)^{1/r})$$

on the bandwidth, proving an $\Omega(\log n)$ lower bound on the number of rounds in the CONGEST model.

There is one caveat, however: they may be able to talk about the smaller instance to each other in the first round. The invariant in round elimination that we have used, where only $o(1)$ extra information about the $(r-1)$-round instance is conveyed will not be true anymore. Nonetheless, even though the first round can reveal some non-trivial information about the smaller $(r-1)$-round instance, solving the instance completely may require large communication in r-rounds. These instances will need drastically different techniques to analyze them.

Chaining other Communication Complexity Problems. In the Chained Index problem, k different instances of Index, all with the same answer were "chained" together. Bob of the first instance also plays the role of Alice of the second instance, and Bob of the second instances takes on the role of Alice of the third instance and so on.

Any two-party communication problem can similarly be chained together. For instance, if multiple set disjointness instances are sampled from the same distribution such that they have the same answer[2], we do not know if beating the $\Omega(n)$ lower bound of [KS92, Raz92, BJKS02] is possible. Our analysis crucially uses the fact that any additional change in the entropy of the answer must have proportional message length, but uses ad hoc techniques specific to Index.

The Chain problem found applications in proving lower bounds for graph problems and streaming submodular maximization (see Section 8.1). Recently, chaining set disjointness was used to prove a lower bound on coloring graphs with low chromatic number in a single pass [ASY26]. The techniques they use only get $\Omega(n/\text{poly}(k))$ lower bounds for chaining k-many instances of set disjointness over a universe of size n.

Closing Gaps for Approximate Shortest Paths. There is an immediate open problem left from Theorem 7. Our upper bound (not covered in this dissertation, can be found in the full version [AHS26]) uses $O(\log^2 n)$ passes in semi-streaming $\widetilde{O}(n)$ space, and our lower bound is $\tilde{\Omega}(\log n)$ passes.

Our approach to prove the lower bound relies on the framework of [FKM+04, GO16] where for p-passes $n^{\Theta(1/p)}$ many instances of pointer chasing overlap. For $p = \omega(\log n / \log\log n)$, these constructions do not give any meaningful results.

[2]This is different from a direct sum version of the problem, where the instances are independent and may have different answers, and we want to solve all of them. Here, solving any one instance suffices to find the answer.

11.2 Broader Directions

In this section, we give some overarching directions that the tools from our dissertation can explore.

11.2.1 Other Related Models.

There are models closely related to the ones we discussed where there are many open directions regarding proving lower bounds.

Random Order Streams. In the graph streaming model, the edges are assumed to be coming in any arbitrary order. All our lower bounds crucially exploit this adversarial ordering of the edges. This assumption is not always necessary, for instance, the edges may be arriving in a stochastic manner from some distribution. Theoretically, random order streams is a well-studied model where the edges are assumed to be given in a uniformly random permutation. [CCM08,PS18,CKKP22,AB20,Ber20,CKKP22,AS23b,HW24] In fact, we know that many problems become significantly easier as the randomness of the order can be used to construct algorithms. For instance, we know that an approximation ratio better an (0.59) can be achieved in a single pass in semi-streaming [AB21, Ber20, HW24] whereas this is provably impossible in adversarial streams [Kap13, Kap21].

Pointer chasing type arguments have been used to prove lower bounds for random order streams also [AS23b]. In the communication complexity arguments in this dissertation, while constructing the lower bounds, we had control over the partition of the edges between the players. Simulating streaming algorithms that receive a random order of the edges is infeasible when the players also do not receive random subsets of the edges. [CCM08] uses the robust communication framework, where the players get random subsets of the edges, to prove lower bounds for a variety of problems such as exact shortest path and connectivity in the random order streaming model.

Exploring the power of randomness over multiple passes is an interesting question. If the algorithm receives the edges in a fresh random order in each pass over the edges, communication complexity, even with the robust communication framework of [CCM08], cannot be used to simulate random order streams. This is because the edge partitions may change over the different passes. As far as we know, there are no ways to model multiple random order streams to prove lower bounds currently in the literature.

Congested Clique. In the CONGEST model we discussed, the vertices are only allowed to communicate over the edges of the underlying graph. In the more powerful CONGESTED CLIQUE model, the vertices know their immediate neighborhood, but they can communicate to any other vertex in the graph even if an edge does not exist.

Triangle detection runs in $O(n^{1-2/\omega})$ where ω is the exponent of matrix multiplication. This is because algebraic methods can be implemented in the congested clique model, see [CKK+15]. The current value of ω is 2.371339 [ADV+25]. However, [DKO14] showed that proving any super-constant lower bound for a problem with a Boolean output in the congested clique model leads to breaking circuit complexity barriers.[3]

[3]In fact, [FGKO18] adapts the framework of [DKO14] with the expander decomposition results of [CPSZ21] to prove a same kind of result for triangle detection in the CONGEST model. More details

With the circuit complexity barriers in mind, hoping for a tight lower bound of $\Omega(n^{1-2/\omega})$ is unrealistic, but we do not know if *any* lower bound, even for one or two rounds can be proven.

11.2.2 Lower Bounds against Edge Sampling

Algorithmic approaches in the streaming model can be broadly classified into two.

(*i*) Vertex Sampling approaches: the algorithm samples a small subset of vertices (say $s \leq n$) at random, and stores information about these vertices. If $s = O(\sqrt{n})$, for instance, all the edges between these vertices can be stored in semi-streaming memory. This is the approach used by [ACG+15] to find a maximal independent set and by [HIMV16, Ass17] to find a set cover. Our matching upper bound of Theorem 4 also uses vertex sampling approaches.

(*ii*) Edge Sampling approaches: the algorithm samples $O(n \cdot \text{poly} \log n)$ edges at random and stores them in memory to process the graph. Our upper bound corresponding to Theorem 7 uses this approach. It is also used by [Ass24, AJJ+22] which give algorithms for approximate maximum matchings.

Our lower bounds inherently partition the graph into $(r-1)$-round hard instances on a smaller set of vertices, and we are able to get tight lower bounds for algorithms that use the vertex sampling approach as well. Both our lower bounds which are not tight (Theorems 3 and 7) for streaming algorithms have state-of-the-art algorithms that employ edge sampling approaches. Proving tight lower bounds for algorithms where edge sampling performs better than vertex sampling would need new techniques.

11.2.3 Structure of Hard Instances: Random versus Combinatorial

In the round elimination arguments of Theorems 1 and 4, to embed the smaller instances, we used combinatorial constructions that ensured that the smaller instances do not interfere with each other. In the case of Theorem 1, this was performed with the DUP graphs of Proposition 5.4.5, in Theorem 3 we use the RS graphs from [GKK12], and in Theorem 4 we use the one-round lower bound of [DK20]. In contrast to all these combinatorial structures, Theorem 7 and Theorem 8 uses a reduction from pointer chasing, where, as we mentioned in Section 2.2.2, the final instances look random.

Theorem 8 is in fact optimal in all three parameters of space, number of passes and the approximation factor, and uses random looking sets to carefully choose subsets of the universe. We did not need combinatorial structure to embed multiple hard instances from $(p-1)$-passes because there was only one underlying instance. It would be interesting to explore what other problems exhibit this property, where one smaller instance suffices to prove optimal lower bounds, and random constructions can be employed.

11.2.4 Lower Bounds for Problems with Minimal Locality

In distributed computing, we know multiple problems with $\Theta(\sqrt{n} + D)$-round complexity where D is the diameter of the graph, such as minimum spanning tree [KP95, SHK+12], global

about their result can be found in Chapter B.

minimum cut [DEMN21, SHK+12], shortest path approximations [BFKL21, BKKL17, Elk04] and approximate maximum flow [GKK+18, SHK+12]. The dependence on the diameter is easy to see even in the LOCAL model without any bandwidth constraints: at least some information about the entire graph must be propagated before any algorithm can arrive at an answer. The $\Omega(\sqrt{n})$ lower bound holds because of bandwidth constraints in the CONGEST model.

The set disjointness framework has led to poly(n) lower bounds for other problems which are easy in the LOCAL model. The hardness of these problems in CONGEST is purely due to the bandwidth limitations, and one prime example is detecting p-cliques for constant p. [CK18] proves a lower bound of $\tilde{\Omega}(\sqrt{n})$ for this problem, and the state-of-the-art upper bound is $\tilde{O}(n^{1-2/p})$ [EFF+22]. [CK18] also shows that there is a protocol in two-party communication for detecing p-cliques such that no lower bound larger than $\tilde{\Omega}(\sqrt{n})$ can be proven.

Our techniques with round elimination to prove Theorem 5 may not be amenable to prove poly(n)-round lower bounds as the sizes of the hard instances grow in each round. The other results which prove $\Omega(\text{poly}(n))$-lower bounds on the number of passes in the streaming model from [ACK19a, AGL+24] are only applicable to problems that have a high locality, namely lexicographically first maximal independent set and degeneracy. Currently, as far as we know, there are no techniques that could be useful in proving an $\omega(\sqrt{n})$ lower bound for detecting p-cliques.

11.3 Conclusion

Obtaining efficient algorithms and matching lower bounds has been the holy grail of theoretical computer science, and in the setting of streaming and distributed computing, such results can be achieved unconditionally through information-theoretic arguments. This dissertation proves such optimal multi-pass streaming lower bounds, resolving the complexity of several problems. Beyond streaming, it makes a first step towards getting multi-round distributed computing lower bounds with new techniques.

We return to the guiding objective stated in Chapter 1:

> **Aim:** To develop a toolkit to prove optimal lower bounds for graph problems with more stringent models of communication.

The results of this dissertation make concrete progress toward this aim. While we now understand some problems quite well, several others remain unresolved as outlined in this chapter where new techniques and ideas may be needed.

References

[AB16] Amir Abboud and Greg Bodwin. The 4/3 additive spanner exponent is tight. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 351–361. ACM, 2016. 64, 71, 73

[AB19] Sepehr Assadi and Aaron Bernstein. Towards a unified theory of sparsification for matching problems. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 11:1–11:20, 2019. 53

[AB20] Sepehr Assadi and Soheil Behnezhad. On the robust communication complexity of bipartite matching with application to random-order streams. Manuscript, 2020. 266

[AB21] Sepehr Assadi and Soheil Behnezhad. Beating two-thirds for random-order streaming matching. *CoRR*, abs/2102.07011. To appear in ICALP 2021, 2021. 266

[ABB+19] Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1616–1635, 2019. 106

[ABJ+25] Amir Azarmehr, Soheil Behnezhad, Rajesh Jayaram, Jakub Lacki, Vahab Mirrokni, and Peilin Zhong. Massively parallel minimum spanning tree in general metric spaces. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 143–174. SIAM, 2025. 4

[ABK+25] Sepehr Assadi, Soheil Behnezhad, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. Settling the pass complexity of approximate matchings in dynamic graph streams. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 864–904. SIAM, 2025. iv, 5, 8, 163, 167, 169, 253, 254

[ABKL23] Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. On regularity lemma and barriers in streaming and dynamic matching. In *STOC '23: 55th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2023. 3, 106

[Abl93] Farid M. Ablayev. Lower bounds for one-way probabilistic communication complexity. In *Automata, Languages and Programming, 20nd International Colloquium, ICALP93, Lund, Sweden, July 5-9, 1993, Proceedings*, pages 241–252, 1993. 11, 60

[Abl96] Farid Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 157(2):139–159, 1996. 13, 14, 15

[ACG+15] Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2237–2246, 2015. 5, 6, 58, 61, 264, 267

[ACGS23] Sepehr Assadi, Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Coloring in graph streams via deterministic and adversarially robust algorithms. In Floris Geerts, Hung Q. Ngo, and Stavros Sintos, editors, *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2023, Seattle, WA, USA, June 18-23, 2023*, pages 141–153. ACM, 2023. 3

[ACK16] Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-linear lower bounds for distributed distance computations, even in sparse networks. In Cyril Gavoille and David Ilcinkas, editors, *Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, volume 9888 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2016. 24

[ACK19a] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial pass lower bounds for graph streaming algorithms. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 265–276, 2019. 3, 15, 18, 19, 58, 162, 268

[ACK19b] Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786, 2019. 5, 32, 58, 59, 60, 264

[ACKL20] Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Christoph Lenzen. Fooling views: a new lower bound technique for distributed computations under congestion. *Distributed Comput.*, 33(6):545–559, 2020. 4, 8, 26, 161, 162, 171

[ACKP21] Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Ami Paz. Smaller cuts, higher lower bounds. *ACM Trans. Algorithms*, 17(4):30:1–30:40, 2021. 18, 25

[ACN05] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In Harold N. Gabow and Ronald Fagin,

editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 684–693. ACM, 2005. 6

[AD16] Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 477–486, 2016. 4

[AD21] Sepehr Assadi and Aditi Dudeja. A simple semi-streaming algorithm for global minimum cuts. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 172–180. SIAM, 2021. 6

[ADV+25] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2005–2039. SIAM, 2025. 164, 266

[AG11] Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, pages 526–538, 2011. 106, 108

[AG15] Kook Jin Ahn and Sudipto Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 202–211, 2015. 8, 108

[AG18] Kook Jin Ahn and Sudipto Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. *ACM Trans. Parallel Comput.*, 4(4):17:1–17:40, 2018. 3, 6, 106

[AGL+24] Sepehr Assadi, Prantar Ghosh, Bruno Loff, Parth Mittal, and Sagnik Mukhopadhyay. Polynomial pass semi-streaming lower bounds for k-cores and degeneracy. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, Ann Arbor, MI, USA, July 22-25, 2024*, volume 300 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. 19, 58, 254, 268

[AGM12] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 459–467, 2012. 6, 7, 8, 108, 264

[AHS26] Sepehr Assadi, Gary Hoppenworth, and Janani Sundaresan. Better bounds for semi-streaming single-source shortest paths. In Kasper Green Larsen and Barna Saha, editors, *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026, Vancouver, BC, Canada, January 11-14, 2026*, pages 5093–5116. SIAM, 2026. iv, 10, 238, 248, 265

[AJJ+22] Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 627–669. SIAM, 2022. 3, 6, 10, 106, 108, 227, 228, 264, 267

[AJX26] Sepehr Assadi, Max Jiang, and Mars Xiang. Semi-streaming matching in a single pass: A new framework for lower bounds via blueprints. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC)*, 2026. To appear. 7

[AKL16] Sepehr Assadi, Sanjeev Khanna, and Yang Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 698–711, 2016. 48, 216, 250, 251, 252, 253

[AKL17] Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742, 2017. 7, 106, 108

[AKLY16] Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016. 8, 106, 108

[AKM22] Sepehr Assadi, Pankaj Kumar, and Parth Mittal. Brooks' theorem in graph streams: a single-pass semi-streaming algorithm for Δ-coloring. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 234–247, New York, NY, USA, 2022. Association for Computing Machinery. 3

[AKNS24a] Sepehr Assadi, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. O(log log n) passes is optimal for semi-streaming maximal independent set. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 847–858. ACM, 2024. iv, 6, 162, 163, 167, 169, 253, 254

[AKNS24b] Sepehr Assadi, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. $\wr(\log \log n)$ passes are optimal for semistreaming maximal independent set. *SIAM Journal on Computing*, 0(0):STOC24–151–STOC24–206, 2024. 6

[AKNS26] Sepehr Assadi, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. Multipass lower bounds for streaming matchings, 2026. In preparation. iv

[AKO18] Mohamad Ahmadi, Fabian Kuhn, and Rotem Oshman. Distributed approximate maximum matching in the CONGEST model. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 6:1–6:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 3

[AKSY20] Sepehr Assadi, Gillat Kol, Raghuvansh R. Saxena, and Huacheng Yu. Multipass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 354–364. IEEE, 2020. 15, 106, 107, 214, 248, 254

[AKZ22] Sepehr Assadi, Gillat Kol, and Zhijun Zhang. Rounds vs communication tradeoffs for maximal independent sets. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1193–1204. IEEE, 2022. 27, 66, 67, 162, 163, 167, 169, 179

[AKZ24] Sepehr Assadi, Gillat Kol, and Zhijun Zhang. Optimal multi-pass lower bounds for MST in dynamic streams. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 835–846. ACM, 2024. 41

[Alo02] Noga Alon. Testing subgraphs in large graphs. *Random Struct. Algorithms*, 21(3-4):359–370, 2002. 53

[ALT21] Sepehr Assadi, S. Cliff Liu, and Robert E. Tarjan. An auction algorithm for bipartite matching in streaming and massively parallel computation models. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 165–171. SIAM, 2021. 6, 106, 264

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29. ACM, 1996. 68, 123, 251

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. 14

[AMS12] Noga Alon, Ankur Moitra, and Benny Sudakov. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1079–1090, 2012. 53

[AN21] Sepehr Assadi and Vishvajeet N. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 612–625. ACM, 2021. 15, 106, 107, 248, 254, 255

[ANRW15] Noga Alon, Noam Nisan, Ran Raz, and Omri Weinstein. Welfare maximization with limited interaction. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1499–1512, 2015. 27, 43, 66, 67, 162, 163, 167, 169, 179

[AR20] Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 342–353. IEEE, 2020. 4, 20, 21, 34, 60, 106, 107, 162, 227, 229

[AS06] Noga Alon and Asaf Shapira. A characterization of easily testable induced subgraphs. *Combinatorics, Probability & Computing*, 15(6):791–805, 2006. 53

[AS22] Sepehr Assadi and Vihan Shah. An asymptotically optimal algorithm for maximum matching in dynamic streams. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 9:1–9:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 6, 106, 108

[AS23a] Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 909–932. IEEE, 2023. iv, 7, 61, 214, 227

[AS23b] Sepehr Assadi and Janani Sundaresan. (noisy) gap cycle counting strikes back: Random order streaming lower bounds for connected components and beyond. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 183–195. ACM, 2023. v, 106, 266

[AS25] Sepehr Assadi and Janani Sundaresan. Distributed triangle detection is hard in few rounds. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 755–766, 2025. iv, 9, 202, 203

[AS26] Sepehr Assadi and Janani Sundaresan. Settling the pass complexity of streaming set cover, 2026. To appear in STOC 2026. iv, 11

[Ass17] Sepehr Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts, editors, *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 321–335. ACM, 2017. 10, 11, 238, 250, 251, 252, 254, 267

[Ass20] S. Assadi. Lecture notes on sublinear algorithms. https://sepehr.assadi.info/courses/cs514-s20/lec8.pdf, 2020. 214

[Ass22] Sepehr Assadi. A two-pass (conditional) lower bound for semi-streaming maximum matching. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 708–742. SIAM, 2022. 7, 21, 34, 60, 106

[Ass23] Sepehr Assadi. Recent advances in multi-pass graph streaming lower bounds. *SIGACT News*, 54(3):48–75, September 2023. 6, 14, 61, 106, 162

[Ass24] Sepehr Assadi. A simple $(1-\varepsilon)$-approximation semi-streaming algorithm for maximum (weighted) matching. In Merav Parter and Seth Pettie, editors, *2024 Symposium on Simplicity in Algorithms, SOSA 2024, Alexandria, VA, USA, January 8-10, 2024*, pages 337–354. SIAM, 2024. 108, 267

[ASW19] Sepehr Assadi, Xiaorui Sun, and Omri Weinstein. Massively parallel algorithms for finding well-connected components in sparse graphs. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, page 461–470, New York, NY, USA, 2019. Association for Computing Machinery. 315

[ASY26] Sepehr Assadi, Janani Sundaresan, and Helia Yazdanyar. Coloring graphs with few colors in the streaming model. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 6065–6132, 2026. v, 265

[AW14] Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 434–443, 2014. 4

[AW22] Sepehr Assadi and Chen Wang. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 6

[AY05] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005. 2

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, 5-8 June 2010*, pages 67–76, 2010. 42, 49, 66, 67, 68, 69, 123, 124, 229, 238

[BBH+19] Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 481–497. IEEE Computer Society, 2019. 5

[BBH+21] Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. *J. ACM*, 68(5):39:1–39:30, 2021. 23

[BBK+25] Alkida Balliu, Sebastian Brandt, Fabian Kuhn, Dennis Olivetti, and Joonatan Saarhelo. Towards fully automatic distributed lower bounds. In Dariusz R. Kowalski, editor, *39th International Symposium on Distributed Computing, DISC 2025, Berlin, Germany, October 27-31, 2025*, volume 356 of *LIPIcs*, pages 13:1–13:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. 23

[BBKO22a] Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed delta-coloring plays hide-and-seek. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 464–477. ACM, 2022. 3, 23

[BBKO22b] Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed edge coloring in time polylogarithmic in Δ. In Alessia Milani and Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 15–25. ACM, 2022. 3

[BBM11] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 210–220. IEEE Computer Society, 2011. 4

[BBO20] Alkida Balliu, Sebastian Brandt, and Dennis Olivetti. Distributed lower bounds for ruling sets. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 365–376. IEEE, 2020. 23

[BC17] Suman K. Bera and Amit Chakrabarti. Towards Tighter Space Bounds for Counting Triangles and Other Substructures in Graph Streams. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects*

of Computer Science (STACS 2017), volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 18

[BCD+19] Nir Bachrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. Hardness of distributed optimization. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 238–247. ACM, 2019. 25, 26

[BCGS17] Reuven Bar-Yehuda, Keren Censor-Hillel, Mohsen Ghaffari, and Gregory Schwartzman. Distributed approximation of maximum independent set and maximum matching. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 165–174. ACM, 2017. 3

[BCMT22] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. Almost 3-approximate correlation clustering in constant rounds. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 720–731. IEEE, 2022. 5, 6

[BCMT23] Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. Single-pass streaming algorithms for correlation clustering. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 819–849. SIAM, 2023. 6

[BDH+19] Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Cliff Stein, and Madhu Sudan. Fully dynamic maximal independent set with polylogarithmic update time. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 382–405. IEEE Computer Society, 2019. 6

[BDL21] Aaron Bernstein, Aditi Dudeja, and Zachary Langley. A framework for dynamic matching in weighted graphs. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 668–681. ACM, 2021. 106, 108

[Beh46] Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences of the United States of America*, 32(12):331, 1946. 34

[Beh21] Soheil Behnezhad. Time-optimal sublinear algorithms for matching and vertex cover. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 873–884. IEEE, 2021. 6

[BEM17] MohammadHossein Bateni, Hossein Esfandiari, and Vahab Mirrokni. Almost optimal streaming algorithms for coverage problems. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '17, page 13–23, New York, NY, USA, 2017. Association for Computing Machinery. 10, 250, 251, 254

[BEPS12] Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 321–330. IEEE Computer Society, 2012. 3

[Ber20] Aaron Bernstein. Improved bounds for matching in random-order streams. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 12:1–12:13, 2020. 106, 266

[BFH+16] Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. A lower bound for the distributed lovász local lemma. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 479–488. ACM, 2016. 23

[BFKL21] Ruben Becker, Sebastian Forster, Andreas Karrenbauer, and Christoph Lenzen. Near-optimal approximate shortest paths and transshipment in distributed and streaming models. *SIAM Journal on Computing*, 50(3):815–856, 2021. 3, 10, 17, 227, 228, 268

[BFS12] Guy E. Blelloch, Jeremy T. Fineman, and Julian Shun. Greedy sequential maximal independent set and matching are parallel on average. In Guy E. Blelloch and Maurice Herlihy, editors, *24th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '12, Pittsburgh, PA, USA, June 25-27, 2012*, pages 308–317. ACM, 2012. 6

[BG14] Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2014. 35, 67, 69

[BGW20] Mark Braverman, Sumegha Garg, and David P. Woodruff. The coin problem with applications to data streams. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 318–329. IEEE, 2020. 51

[BH22] Greg Bodwin and Gary Hoppenworth. New additive spanner lower bounds by an unlayered obstacle product. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 778–788. IEEE, 2022. 64

[BHP12] Andrew Berns, James Hegeman, and Sriram V. Pemmaraju. Super-fast distributed algorithms for metric facility location. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 428–439. Springer, 2012. 6

[BI15] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless seth is false). In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 51–58, New York, NY, USA, 2015. Association for Computing Machinery. 3

[BJKS02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 209–218. IEEE Computer Society, 2002. 14, 18, 19, 49, 229, 265

[BKK+23] Alkida Balliu, Janne H. Korhonen, Fabian Kuhn, Henrik Lievonen, Dennis Olivetti, Shreyas Pai, Ami Paz, Joel Rybicki, Stefan Schmid, Jan Studený, Jukka Suomela, and Jara Uitto. Sinkless orientation made simple. In Telikepalli Kavitha and Kurt Mehlhorn, editors, *2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023*, pages 175–191. SIAM, 2023. 23

[BKKL17] Ruben Becker, Andreas Karrenbauer, Sebastian Krinninger, and Christoph Lenzen. Near-optimal approximate shortest paths and transshipment in distributed and streaming models. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 7:1–7:16, 2017. 268

[BKM20] Philipp Bamberger, Fabian Kuhn, and Yannic Maus. Efficient deterministic distributed coloring with small bandwidth. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 243–252. ACM, 2020. 3

[BKO22] Sujoy Bhore, Fabian Klute, and Jelle J. Oostveen. On streaming algorithms for geometric independent set and clique. In Parinya Chalermsook and Bundit Laekhanukit, editors, *Approximation and Online Algorithms*, pages 211–224, Cham, 2022. Springer International Publishing. 212

[BLM93] Yitzhak Birk, Nathan Linial, and Roy Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory*, 39(1):186–191, 1993. 53

[BLW76] Norman L. Biggs, E. Keith Lloyd, and Robin J. Wilson. *Graph Theory, 1736–1936*. Clarendon Press, Oxford, 1976. 2

[BMdV26] Joakim Blikstad, Yannic Maus, and Tijn de Vos. Deterministic edge coloring with few colors in congest, 2026. 3

[BN19] Aaron Bernstein and Danupon Nanongkai. Distributed exact weighted all-pairs shortest paths in randomized near-linear time. *SIAM J. Comput.*, 52(on):STOC19–112–STOC19–127, 2019. 3, 25

[BOV13] Vladimir Braverman, Rafail Ostrovsky, and Dan Vilenchik. How hard is counting triangles in the streaming model? In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming*, pages 244–254, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 15, 18

[BRWY13a] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct product via round-preserving compression. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 2013. 67

[BRWY13b] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013*, pages 746–755, 2013. 67

[BS15] Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, September 14-16, 2015, Proceedings*, pages 263–274, 2015. 106

[Bue79] Francis Buekenhout. Diagrams for geometries and groups. *Journal of Combinatorial Theory, Series A*, 27(2):121–151, 1979. 253

[BvdBE+22] Joakim Blikstad, Jan van den Brand, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Nearly optimal communication and query complexity of bipartite matching. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1174–1185. IEEE, 2022. 18

[BW15] Mark Braverman and Omri Weinstein. An interactive information odometer and applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 341–350. ACM, 2015. 67

[CCE+16] Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, January 10-12, 2016*, pages 1326–1344, 2016. 106, 108

[CCEW23] Xiuge Chen, Rajesh Chitnis, Patrick Eades, and Anthony Wirth. Sublinear-space streaming algorithms for estimating graph parameters on sparse graphs. In Pat Morin and Subhash Suri, editors, *Algorithms and Data Structures - 18th International Symposium, WADS 2023, Montreal, QC, Canada, July 31 - August 2, 2023, Proceedings*, volume 14079 of *Lecture Notes in Computer Science*, pages 247–261. Springer, 2023. 15

[CCGL21] Keren Censor-Hillel, Yi-Jun Chang, François Le Gall, and Dean Leitersdorf. Tight distributed listing of cliques. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2878–2891. SIAM, 2021. 25, 164

[CCHM15] Rajesh Hemant Chitnis, Graham Cormode, Mohammad Taghi Hajiaghayi, and Morteza Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1234–1251. SIAM, 2015. 15, 106, 108

[CCKM13] Amit Chakrabarti, Graham Cormode, Ranganath Kondapally, and Andrew McGregor. Information cost tradeoffs for augmented index and streaming language recognition. *SIAM Journal on Computing*, 42(1):61–83, 2013. 41, 214

[CCM08] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 641–650, 2008. 65, 253, 266

[CCZ25] Shiri Chechik, Hongyi Chen, and Tianyi Zhang. Improved streaming edge coloring. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, Aarhus, Denmark, July 8-11, 2025*, volume 334 of *LIPIcs*, pages 48:1–48:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. 3

[CDK14] Flavio Chierichetti, Nilesh N. Dalvi, and Ravi Kumar. Correlation clustering in mapreduce. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 641–650. ACM, 2014. 6

[CDK19] Graham Cormode, Jacques Dark, and Christian Konrad. Independent Sets in Vertex-Arrival Streams. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 3, 5, 9, 15, 32, 58, 59, 60, 212, 213, 264

[CE06] Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discret. Math.*, 20(2):463–501, 2006. 64

[Cen22] Keren Censor-Hillel. Distributed subgraph finding: Progress and challenges. *CoRR*, abs/2203.06597, 2022. 25, 161, 162, 164

[CF13] David Conlonl and Jacob Fox. Graph removal lemmas. *Surveys in combinatorics 2013*, 409:1, 2013. 53

[CFHT20] Yi-Jun Chang, Martin Farach-Colton, Tsan-sheng Hsu, and Meng-Tsung Tsai. Streaming complexity of spanning tree computation. In *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, pages 34:1–34:19, 2020. 17

[CGQ15] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 318–330. Springer, 2015. 106

[CHKP17] Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and Near-Quadratic Lower Bounds for the CONGEST Model. In Andréa Richa, editor, *31st International Symposium on Distributed Computing (DISC 2017)*, volume 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 4, 25

[CHS24] Yi-Jun Chang, Shang-En Huang, and Hsin-Hao Su. Deterministic expander routing: Faster and more versatile. In Ran Gelles, Dennis Olivetti, and Petr Kuznetsov, editors, *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing, PODC 2024, Nantes, France, June 17-21, 2024*, pages 194–204. ACM, 2024. 161, 164

[Chv79] Vasek Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235, 1979. 250

[CJMM17] Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, and S. Muthukrishnan. The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms, ESA 2017, Vienna, Austria, September 4-6, 2017*, volume 87 of *LIPIcs*, pages 29:1–29:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 15, 106

[CK14] Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: Matchings, matroids, and more. In Jon Lee and Jens Vygen, editors, *Integer Programming and Combinatorial Optimization - 17th International Conference, IPCO 2014, Bonn, Germany, June 23-25, 2014. Proceedings*, volume 8494 of *Lecture Notes in Computer Science*, pages 210–221. Springer, 2014. 106

[CK18] Artur Czumaj and Christian Konrad. Detecting cliques in CONGEST networks. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 4, 25, 164, 268

[CKK+15] Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 143–152. ACM, 2015. 164, 266

[CKKP22] Ashish Chiplunkar, John Kallaugher, Michael Kapralov, and Eric Price. Factorial lower bounds for (almost) random order streams. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 486–497, 2022. 266

[CKL+24] Melanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. A $(3+\varepsilon)$-approximate correlation clustering algorithm in dynamic streams. In David Woodruff, editor, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA 2024*. SIAM, 2024. 6

[CKP+21a] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Almost optimal super-constant-pass streaming lower bounds for reachability. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 570–583. ACM, 2021. 4, 21, 60, 61, 106, 107, 162, 227, 229, 254

[CKP+21b] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Near-optimal two-pass streaming algorithm for sampling

random walks over directed graphs. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, volume 198 of *LIPIcs*, pages 52:1–52:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 17, 34

[CKP+23] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Towards multi-pass streaming lower bounds for optimal approximation of max-cut. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 878–924. SIAM, 2023. 15

[CKPU23] Mélanie Cambus, Fabian Kuhn, Shreyas Pai, and Jara Uitto. Time and space optimal massively parallel algorithm for the 2-ruling set problem. *CoRR*, abs/2306.00432, 2023. 6

[CKS03] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117, 2003. 25

[CKT10] Flavio Chierichetti, Ravi Kumar, and Andrew Tomkins. Max-cover in map-reduce. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 231–240, 2010. 250

[CLJ+14] Rayan Chikhi, Antoine Limasset, Shaun Jackman, Jared T. Simpson, and Paul Medvedev. On the representation of de bruijn graphs. In Roded Sharan, editor, *Research in Computational Molecular Biology*, pages 35–55, Cham, 2014. Springer International Publishing. 2

[CLM+21] Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation clustering in constant many parallel rounds. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2069–2078. PMLR, 2021. 6

[CLV22] Keren Censor-Hillel, Dean Leitersdorf, and David Vulakh. Deterministic near-optimal distributed listing of cliques. In Alessia Milani and Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 271–280. ACM, 2022. 161, 164

[CMGS25] Daoyuan Chen, Simon Meierhans, Maximilian Probst Gutenberg, and Thatchaphol Saranurak. Parallel and distributed expander decomposition:

Simple, fast, and near-optimal. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1705–1719. SIAM, 2025. 9, 163, 311

[CPSZ21] Yi-Jun Chang, Seth Pettie, Thatchaphol Saranurak, and Hengjie Zhang. Near-optimal distributed triangle enumeration via expander decompositions. *J. ACM*, 68(3):21:1–21:36, 2021. 8, 161, 164, 266, 312, 313

[CPZ19] Yi-Jun Chang, Seth Pettie, and Hengjie Zhang. Distributed triangle detection via expander decomposition. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 821–840. SIAM, 2019. 8, 161

[CS14] Michael Crouch and Daniel S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014*, pages 96–104, 2014. 106

[CS19] Yi-Jun Chang and Thatchaphol Saranurak. Improved distributed expander decomposition and nearly optimal triangle enumeration. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 66–73. ACM, 2019. 8, 161, 164

[CS20] Yi-Jun Chang and Thatchaphol Saranurak. Deterministic distributed expander decomposition and routing with applications in distributed derandomization. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 377–388. IEEE, 2020. 161, 164

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 270–278. IEEE Computer Society, 2001. 35, 42, 49, 68, 123, 229

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. 53, 54

[CV86] R Cole and U Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 206–219, New York, NY, USA, 1986. Association for Computing Machinery. 22

[CW16] Amit Chakrabarti and Anthony Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1365–1373, 2016. 16, 17, 250, 251, 252, 253, 254

[DDK23] Jacques Dark, Adithya Diddapur, and Christian Konrad. Interval Selection in Data Streams: Weighted Intervals and the Insertion-Deletion Setting. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 216

[DEMN21] Michal Dory, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Distributed weighted min-cut in nearly-optimal time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1144–1153, New York, NY, USA, 2021. Association for Computing Machinery. 25, 268

[DIMV14] Erik D. Demaine, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian. On streaming and communication complexity of the set cover problem. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 484–498, 2014. 250, 251

[DJS96] Carsten Damm, Stasys Jukna, and Jirí Sgall. Some bounds on multiparty communication complexity of pointer jumping. In Claude Puech and Rüdiger Reischuk, editors, *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings*, volume 1046 of *Lecture Notes in Computer Science*, pages 643–654. Springer, 1996. 9, 213

[DK20] Jacques Dark and Christian Konrad. Optimal lower bounds for matching and vertex cover in dynamic graph streams. *CoRR*, abs/2005.11116, 2020. To appear in CCC 2020. xvi, 6, 8, 40, 41, 43, 58, 106, 108, 267

[DKO14] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 367–376. ACM, 2014. 4, 25, 43, 162, 164, 266

[DKRM25] Mohammad Mehdi Daliri Khomami, Alireza Rezvanian, and Mohammad Reza Meybodi. Maximum independent set in multiplex social networks and its application to influence maximization. *Scientific Reports*, 15(1):16322, May 2025. 5

[DP09] Devdatt P Dubhashi and Alessandro Panconesi. **Concentration of measure for the analysis of randomized algorithms**. Cambridge University Press, 2009. 53

[DS05] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005. 3

[DS14] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014. 250

[EFF+22] Talya Eden, Nimrod Fiat, Orr Fischer, Fabian Kuhn, and Rotem Oshman. Sublinear-time distributed algorithms for detecting small cliques and even cycles. *Distributed Comput.*, 35(3):207–234, 2022. 8, 9, 161, 163, 164, 268, 310

[EGK20] Yuval Efron, Ofer Grossman, and Seri Khoury. Beyond alice and bob: Improved inapproximability for maximum independent set in CONGEST. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 511–520. ACM, 2020. 25

[EHL+15] Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, January 4-6, 2015*, pages 1217–1233, 2015. 106

[EHM16] Hossein Esfandiari, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh. Finding large matchings in semi-streaming. In Carlotta Domeniconi, Francesco Gullo, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, editors, *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain*, pages 608–614. IEEE Computer Society, 2016. 106

[EKMS12] Sebastian Eggert, Lasse Kliemann, Peter Munstermann, and Anand Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1-2):490–508, 2012. 6, 106

[Elk04] Michael Elkin. Unconditional lower bounds on the time-approximation trade-offs for the distributed minimum spanning tree problem. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 331–340, New York, NY, USA, 2004. Association for Computing Machinery. 268

[Elk17] Michael Elkin. Distributed exact shortest paths in sublinear time. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 757–770. ACM, 2017. 3

[EN18] Michael Elkin and Ofer Neiman. Efficient algorithms for constructing very sparse spanners and emulators. *ACM Transactions on Algorithms (TALG)*, 15(1):1–29, 2018. 227

[EN19] Michael Elkin and Ofer Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. *SIAM Journal on Computing*, 48(4):1436–1480, 2019. 10, 227

[EP04] Michael Elkin and David Peleg. $(1+\epsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004. 10, 227

[ER16] Yuval Emek and Adi Rosén. Semi-streaming set cover. *ACM Trans. Algorithms. Conference version in* ICALP 2014., 13(1):6:1–6:22, 2016. 250, 251, 252

[ET22] Michael Elkin and Chhaya Trehan. $(1+\varepsilon)$-approximate shortest paths in dynamic streams. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 51:1–51:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 3, 227, 228

[EZ04] Michael Elkin and Jian Zhang. Efficient algorithms for constructing $(1+\varepsilon,\beta)$-spanners in the distributed and streaming models. In Soma Chaudhuri and Shay Kutten, editors, *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004*, pages 160–168. ACM, 2004. 10, 227

[Fac26] Facebook. https://en.wikipedia.org/wiki/Facebook, 2026. Accessed: 1 April 2026. 2

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. 250

[FFK21] Salwa Faour, Marc Fuchs, and Fabian Kuhn. Distributed CONGEST approximation of weighted vertex covers and matchings. In Quentin Bramas, Vincent Gramoli, and Alessia Milani, editors, *25th International Conference on Principles of Distributed Systems, OPODIS 2021, Strasbourg, France, December 13-15, 2021*, volume 217 of *LIPIcs*, pages 17:1–17:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 3

[FFSZ22] Ashkan Norouzi Fard, Moran Feldman, Ola Svensson, and Rico Zenklusen. Submodular maximization subject to matroid intersection on the fly. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, pages 52:1–52:14, 2022. 212

[FGKO18] Orr Fischer, Tzlil Gonen, Fabian Kuhn, and Rotem Oshman. Possibilities and impossibilities for distributed subgraph detection. In Christian Scheideler

and Jeremy T. Fineman, editors, *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*, pages 153–162. ACM, 2018. 4, 8, 24, 26, 44, 45, 46, 161, 162, 171, 266

[FGL+91] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost np-complete. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 2–12, 1991. 3

[FHM+20] Alireza Farhadi, Mohammad Taghi Hajiaghayi, Tung Mai, Anup Rao, and Ryan A. Rossi. Approximate maximum matching in random streams. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1773–1785, 2020. 106

[FHM23] Manuela Fischer, Magnús M. Halldórsson, and Yannic Maus. Fast distributed brooks' theorem. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2567–2588, 2023. 3

[FHS15] Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *arXiv preprint arXiv:1512.07852*, 2015. 7, 53, 107

[FHW12] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012. 24

[FKM+04] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 531–543. Springer, 2004. 2, 10, 12, 227, 265

[FKM+05] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the streaming model: the value of space. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 745–754. SIAM, 2005. 4, 6, 10, 15, 17, 20, 48, 106, 227, 229

[FKM+08] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008. 3, 4, 15, 16, 17, 18, 19

[FLN+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002. 7, 52, 53, 107

[FMU22] Manuela Fischer, Slobodan Mitrović, and Jara Uitto. Deterministic $(1+\epsilon)$-approximate maximum matching with poly$(1/\epsilon)$ passes in the semi-streaming model and beyond. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 248–260, New York, NY, USA, 2022. Association for Computing Machinery. 3, 6, 106

[FN18a] Manuela Fischer and Andreas Noever. Tight analysis of parallel randomized greedy MIS. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2152–2160. SIAM, 2018. 6

[FN18b] Sebastian Forster and Danupon Nanongkai. A faster distributed single-source shortest paths algorithm. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 686–697. IEEE Computer Society, 2018. 3

[FNFSZ20] Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 1363–1374, New York, NY, USA, 2020. Association for Computing Machinery. 9, 212, 213, 214

[Fox11] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174(1):561–579, 2011. 53

[GGK+18] Mohsen Ghaffari, Themis Gouleakis, Christian Konrad, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved massively parallel computation algorithms for mis, matching, and vertex cover. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, July 23-27, 2018*, pages 129–138, 2018. 5, 58, 61

[GJPW18] Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication versus partition number. *TOCT*, 10(1):4:1–4:20, 2018. 238

[GK21] Mohsen Ghaffari and Fabian Kuhn. Deterministic distributed vertex coloring: Simpler, faster, and without network decomposition. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1009–1020. IEEE, 2021. 3

[GKK+08] Lukasz Golab, Howard J. Karloff, Flip Korn, Divesh Srivastava, and Bei Yu. On generating near-optimal tableaux for conditional functional dependencies. *Proc. VLDB Endow.*, 1(1):376–390, 2008. 250

[GKK12] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485, 2012. xvi, 4, 6, 7, 19, 21, 37, 38, 39, 52, 53, 61, 106, 267

[GKK+18] Mohsen Ghaffari, Andreas Karrenbauer, Fabian Kuhn, Christoph Lenzen, and Boaz Patt-Shamir. Near-optimal distributed maximum flow. *SIAM Journal on Computing*, 47(6):2078–2117, 2018. 268

[GKMS19] Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 491–500, 2019. 106, 108

[GKP98] Juan A. Garay, Shay Kutten, and David Peleg. A sublinear time distributed algorithm for minimum-weight spanning trees. *SIAM J. Comput.*, 27(1):302–316, 1998. 25

[GKS17] Mohsen Ghaffari, Fabian Kuhn, and Hsin-Hao Su. Distributed mst and routing in almost mixing time. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, page 131–140, New York, NY, USA, 2017. Association for Computing Machinery. 312, 313

[GKSS23] Varun Gupta, Ravishankar Krishnaswamy, Sai Sandeep, and Janani Sundaresan. Look before, before you leap: Online vector load balancing with few reassignments. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, MIT, Cambridge, Massachusetts, USA, January 10-13, 2023*, LIPIcs, pages 65:1–65:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. v

[GKW21] Alexander Golovnev, Alexander S. Kulikov, and R. Ryan Williams. Circuit Depth Reductions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 318

[GL18] Mohsen Ghaffari and Jason Li. New Distributed Algorithms in Almost Mixing Time via Transformations from Parallel Algorithms. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing (DISC 2018)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 313

[GM08] Sudipto Guha and Andrew McGregor. Tight lower bounds for multi-pass stream computation via pass elimination. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, pages 760–772, 2008. 51

[GO95] Anka Gajentaan and Mark H Overmars. On a class of o(n2) problems in computational geometry. *Computational Geometry*, 5(3):165–185, 1995. 3

[GO13] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013. 10

[GO16] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *Algorithmica*, 76(3):654–683, 2016. 4, 10, 16, 17, 19, 48, 49, 106, 162, 227, 229, 254, 265

[Gow01] WT Gowers. Some unsolved problems in additive/combinatorial number theory. *preprint*, 2001. 53

[GT24] Mohsen Ghaffari and Anton Trygub. A near-optimal low-energy deterministic distributed SSSP with ramifications on congestion and APSP. In Ran Gelles, Dennis Olivetti, and Petr Kuznetsov, editors, *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing, PODC 2024, Nantes, France, June 17-21, 2024*, pages 401–411. ACM, 2024. 3

[GW97] Tal Grossman and Avishai Wool. Computational experience with approximation algorithms for the set covering problem. *European Journal of Operational Research*, 101(1):81–92, August 1997. 250

[Has86] J Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 6–20, New York, NY, USA, 1986. Association for Computing Machinery. 3

[Hes03] William Hesse. Directed graphs requiring large numbers of shortcuts. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA*, pages 665–669. ACM/SIAM, 2003. 64

[HGG09] Jonathan Huang, Carlos Guestrin, and Leonidas Guibas. Fourier theoretic probabilistic inference over permutations. *J. Mach. Learn. Res.*, 10:997–1070, jun 2009. 107

[HIMV16] Sariel Har-Peled, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian. Towards tight bounds for the streaming set cover problem. In *Proceedings of the 35th*

ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '16, page 371–383, New York, NY, USA, 2016. Association for Computing Machinery. 10, 17, 250, 251, 252, 253, 254, 267

[HJMR07] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 10–23. IEEE Computer Society, 2007. 35, 42, 67, 69, 238

[HKMT21] Magnús M. Halldórsson, Fabian Kuhn, Yannic Maus, and Tigran Tonoyan. Efficient randomized distributed coloring in CONGEST. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1180–1193. ACM, 2021. 3

[HKN16] Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. A deterministic almost-tight distributed algorithm for approximating single-source shortest paths. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 489–498, 2016. 3, 10, 227

[HKNS15] Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 21–30, New York, NY, USA, 2015. Association for Computing Machinery. 4

[HP19] Zengfeng Huang and Pan Peng. Dynamic graph stream algorithms in o(n) space. *Algorithmica*, 81(5):1965–1987, 2019. 15

[HP21] Shang-En Huang and Seth Pettie. Lower bounds on sparse spanners, emulators, and diameter-reducing shortcuts. *SIAM J. Discret. Math.*, 35(3):2129–2144, 2021. 64

[HPZZ21] Dawei Huang, Seth Pettie, Yixiang Zhang, and Zhijun Zhang. The communication complexity of set intersection and multiple equality testing. *SIAM Journal on Computing*, 50(2):674–717, 2021. 161

[HS23] Shang-En Huang and Hsin-Hao Su. (1-ϵ)-approximate maximum weighted matching in poly($1/\epsilon$, log n) time in the distributed and parallel settings. *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing*, 2023. 3

[HSSW12] Magnús M. Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium,*

ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I, volume 7391 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2012. 18

[HW24] Diba Hashemi and Weronika Wrzos-Kaminska. Weighted matching in the random-order streaming and robust communication models. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, London School of Economics, London, UK, August 28-30, 2024*, volume 317 of *LIPIcs*, pages 16:1–16:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. 3, 266

[HZ23] Alireza Haqi and Hamid Zarrabi-Zadeh. Almost optimal massively parallel algorithms for k-center clustering and diversity maximization. In Kunal Agrawal and Julian Shun, editors, *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2023, Orlando, FL, USA, June 17-19, 2023*, pages 239–247. ACM, 2023. 6

[IG17] Taisuke Izumi and François Le Gall. Triangle finding and listing in CONGEST networks. In Elad Michael Schiller and Alexander A. Schwarzmann, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 381–389. ACM, 2017. 8, 24, 161, 164

[IKY24] Taisuke Izumi, Naoki Kitamura, and Yutaro Yamaguchi. A nearly linear-time distributed algorithm for exact maximum matching. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4062–4082. SIAM, 2024. 3

[IMR+17] Piotr Indyk, Sepideh Mahabadi, Ronitt Rubinfeld, Jonathan Ullman, Ali Vakilian, and Anak Yodpinyanee. Fractional set cover in the streaming model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 12:1–12:20, 2017. 10, 250, 251, 253, 254

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. 3

[Joh74] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974. 250

[JPY16] Rahul Jain, Attila Pereszlényi, and Penghui Yao. A direct product theorem for two-party bounded-round public-coin communication complexity. *Algorithmica*, 76(3):720–748, 12 2016. A preliminary version of this article has appeared in the Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012. 42, 124

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003. 16, 17, 39

[JRS09] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A property of quantum relative entropy with an application to privacy in quantum communication. *J. ACM*, 56(6), sep 2009. 65, 215

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989. 311

[JX22] Ce Jin and Yinzhan Xu. Tight dynamic problem lower bounds from generalized bmm and omv. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1515–1528, New York, NY, USA, 2022. Association for Computing Machinery. 4

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1):40–56, 1982. 3

[Kap12] Mikhail Kapralov. Better bounds for matchings in the streaming model. In *ACM-SIAM Symposium on Discrete Algorithms*, 2012. 212

[Kap13] Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013. 6, 7, 37, 106, 266

[Kap21] Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1874–1893. SIAM, 2021. 4, 6, 7, 37, 106, 266

[Kho05] Subhash Khot. On the unique games conjecture. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, page 3, USA, 2005. IEEE Computer Society. 3

[KK19] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating max-cut. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 277–288, New York, NY, USA, 2019. Association for Computing Machinery. 4

[KK22] Sanjeev Khanna and Christian Konrad. Optimal Bounds for Dominating Set in Graph Streams. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz In-*

ternational Proceedings in Informatics (LIPIcs), pages 93:1–93:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 250

[KKA23] Sanjeev Khanna, Christian Konrad, and Cezar-Mihail Alexandru. Set cover in the one-pass edge-arrival streaming model. In Floris Geerts, Hung Q. Ngo, and Stavros Sintos, editors, *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2023, Seattle, WA, USA, June 18-23, 2023*, pages 127–139. ACM, 2023. 250

[KKS14] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014. 106

[KKS15] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1263–1282. SIAM, 2015. 15, 214

[KKTY21] Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. Towards tight bounds for spectral sparsification of hypergraphs. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 598–611. ACM, 2021. 53

[KL80] Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, page 302–309, New York, NY, USA, 1980. Association for Computing Machinery. 3

[KMM12] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 231–242, 2012. 6, 106

[KMNT20] Michael Kapralov, Slobodan Mitrovic, Ashkan Norouzi-Fard, and Jakab Tardos. Space efficient approximation to maximum matching size from uniform edge samples. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1753–1772, 2020. 106

[KMT+22] Michael Kapralov, Amulya Musipatla, Jakab Tardos, David P. Woodruff, and Samson Zhou. Noisy Boolean Hidden Matching with Applications. In Mark

Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 91:1–91:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 214

[KMVV13] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '13, page 1–10, New York, NY, USA, 2013. Association for Computing Machinery. 3, 58

[KMW16] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2):17:1–17:44, 2016. 4, 5, 23

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997. 67, 123, 217

[KN21] Christian Konrad and Kheeran K. Naidu. On two-pass streaming algorithms for maximum bipartite matching. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 19:1–19:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 7, 106

[KN24] Christian Konrad and Kheeran K. Naidu. An unconditional lower bound for two-pass streaming algorithms for maximum matching approximation. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024*. SIAM, 2024. xvi, 7, 21, 34, 37, 38, 39, 61, 65, 106, 109

[KNR95] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, page 596–605, New York, NY, USA, 1995. Association for Computing Machinery. 11, 13, 14, 15, 60

[KNS23] Christian Konrad, Kheeran K. Naidu, and Arun Steward. Maximum matching via maximal matching queries. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 41:1–41:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. 106

[Kon15] Christian Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015. 106, 108

[Kon18a] Christian Konrad. MIS in the congested clique model in o(log log Δ) rounds. *CoRR*, abs/1802.07647, 2018. 5

[Kon18b] Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 74:1–74:16, 2018. 3, 6, 58, 106

[KP95] Shay Kutten and David Peleg. Fast distributed construction of k-dominating sets and applications. In *Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '95, page 238–251, New York, NY, USA, 1995. Association for Computing Machinery. 25, 267

[KPRR19] Christian Konrad, Sriram V. Pemmaraju, Talal Riaz, and Peter Robinson. The complexity of symmetry breaking in massive graphs. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary*, volume 146 of *LIPIcs*, pages 26:1–26:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 6

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995. 4

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, 1992. 14, 18, 19, 265

[KS25] Seri Khoury and Aaron Schild. Round elimination via self-reduction: Closing gaps for distributed maximal matching. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 2292–2305. IEEE, 2025. 5, 23

[KT17] Sagar Kale and Sumedh Tirodkar. Maximum matching in two, three, and a few more passes over graph streams. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 15:1–15:21, 2017. 6, 106

[KW20] Young Kun-Ko and Omri Weinstein. An adaptive step toward the multiphase conjecture. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 752–761. IEEE, 2020. 4

[Lin92] Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992. 4, 22, 161

[LMSV11] Silvio Lattanzi, Benjamin Moseley, Siddharth Suri, and Sergei Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 85–94, 2011. 58, 108

[LNW14] Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183, 2014. 51

[Lov75] László Lovász. Three short proofs in graph theory. *Journal of Combinatorial Theory, Series B*, 19(3):269–271, 1975. 250

[LP15] Christoph Lenzen and Boaz Patt-Shamir. Fast partial distance estimation and applications. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 153–162. ACM, 2015. 25

[LPP15] Zvi Lotker, Boaz Patt-Shamir, and Seth Pettie. Improved distributed approximate matching. *J. ACM*, 62(5):38:1–38:17, 2015. 3

[LPPP05] Zvi Lotker, Boaz Patt-Shamir, Elan Pavlov, and David Peleg. Minimum-weight spanning tree construction in $O(\log \log n)$ communication rounds. *SIAM J. Comput.*, 35(1):120–131, 2005. 164

[LS08] Troy Lee and Adi Shraibman. Disjointness is hard in the multi-party number-on-the-forehead model. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 81–91, 2008. 43

[LSZ20] S. Cliff Liu, Zhao Song, and Hengjie Zhang. Breaking the n-pass barrier: A streaming algorithm for maximum weight bipartite matching. *CoRR*, abs/2009.06106, 2020. 6, 106

[Lub85] Michael Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 1–10, 1985. 5, 58

[LW21] Roie Levin and David Wajc. Streaming submodular matching meets the primal-dual method. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1914–1933. SIAM, 2021. 106

[LWWX22] Kevin Lu, Virginia Vassilevska Williams, Nicole Wein, and Zixuan Xu. Better lower bounds for shortcut sets and additive spanners via an improved alternation product. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3311–3331. SIAM, 2022. 64

[McG05] Andrew McGregor. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop*

on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings, pages 170–181, 2005. 6, 21, 106, 108

[McG14] Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, 2014. 14

[MDQ+18] Vikram S. Mailthody, Ketan Date, Zaid Qureshi, Carl Pearson, Rakesh Nagi, Jinjun Xiong, and Wen-mei Hwu. Collaborative (cpu + gpu) algorithms for triangle counting and truss decomposition. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7, 2018. 2

[Mih99] Milena Mihail. Set cover with requirements and costs evolving over time. In Dorit S. Hochbaum, Klaus Jansen, José D. P. Rolim, and Alistair Sinclair, editors, *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*, pages 63–72, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. 250

[MKPV21] Tiago Monteiro, Xenia Klimentova, João Pedro Pedroso, and Ana Viana. A comparison of matching algorithms for kidney exchange programs addressing waiting time. *Central Eur. J. Oper. Res.*, 29(2):539–552, 2021. 6

[MNSW95] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In Frank Thomson Leighton and Allan Borodin, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 103–111. ACM, 1995. xv, 11, 15, 27, 28, 43, 44, 59, 61, 167

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998. 4

[MS77] F. Jessie MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977. 217

[MV16] Andrew McGregor and Sofya Vorotnikova. Planar matching in streams revisited. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016*, pages 17:1–17:12, 2016. 106

[MV18] Andrew McGregor and Sofya Vorotnikova. A simple, space-efficient, streaming algorithm for matchings in low arboricity graphs. In *1st Symposium on Simplicity in Algorithms, SOSA 2018, January 7-10, 2018*, pages 14:1–14:4, 2018. 106

[MYHZ24] Guangxu Yang Mi-Ying Huang, Xinyu Mao and Jiapeng Zhang. Breaking square-root loss barriers via min-entropy. *Electron. Colloquium Comput. Complex.*, pages TR24–067, 2024. 213, 216

[MYZ25] Xinyu Mao, Guangxu Yang, and Jiapeng Zhang. Gadgetless Lifting Beats Round Elimination: Improved Lower Bounds for Pointer Chasing. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:14, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 47, 50

[Nan14] Danupon Nanongkai. Distributed approximation algorithms for weighted shortest paths. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 565–573. ACM, 2014. 3, 24, 25

[Nao91] Moni Naor. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM J. Discret. Math.*, 4(3):409–412, 1991. 23

[Nis02] Noam Nisan. The communication complexity of approximate set packing and covering. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 868–875, 2002. 250, 251, 252

[NW91] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 419–429, 1991. 14, 16, 17

[NW93] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM Journal on Computing*, 22(1):211–219, 1993. 47, 50

[Pel00] David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000. 2, 52, 161

[PR00] David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. *SIAM J. Comput.*, 30(5):1427–1442, 2000. 24, 25

[PRS18] Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. On the distributed complexity of large-scale graph computations. In Christian Scheideler and Jeremy T. Fineman, editors, *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*, pages 405–414. ACM, 2018. 164

[PS84] Christos H. Papadimitriou and Michael Sipser. Communication complexity. *J. Comput. Syst. Sci.*, 28(2):260–269, 1984. 11, 14, 15, 254

[PS97] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997. 53

[PS17] Ami Paz and Gregory Schwartzman. A $(2 + \varepsilon)$-approximation for maximum weight matching in the semi-streaming model. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2153–2161, 2017. 106

[PS18] Pan Peng and Christian Sohler. Estimating graph parameters from random order streams. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2449–2466, 2018. 3, 266

[Raz87] A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, Apr 1987. 3

[Raz92] Alexander A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992. 14, 18, 19, 265

[RS78] Imre Z Ruzsa and Endre Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978. 19, 21, 34, 52, 64, 69, 71

[RVW18] Tim Roughgarden, Sergei Vassilvitskii, and Joshua R. Wang. Shuffles and circuits (on lower bounds for modern parallel computation). *J. ACM*, 65(6), November 2018. 4

[RXC+21] Jing Ren, Feng Xia, Xiangtai Chen, Jiaying Liu, Mingliang Hou, Ahsan Shehzad, Nargiz Sultanova, and Xiangjie Kong. Matching algorithms: Fundamentals, applications and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(3):332–350, 2021. 6

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. 67, 69, 123, 217, 238

[Sag11] Mert Saglam. Tight bounds for data stream algorithms and communication problems. Master's thesis, Simon Fraser University, 2011. 48, 216

[SG09] Barna Saha and Lise Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, Sparks, Nevada, USA*, pages 697–708, 2009. 250, 252

[SGB23] Stevan Stanovic, Benoit Gaüzère, and Luc Brun. Maximal independent sets for pooling in graph neural networks. In Mario Vento, Pasquale Foggia, Donatello Conte, and Vincenzo Carletti, editors, *Graph-Based Representations in Pattern Recognition - 13th IAPR-TC-15 International Workshop, GbRPR 2023, Vietri sul Mare, Italy, September 6-8, 2023, Proceedings*, volume 14121 of *Lecture Notes in Computer Science*, pages 113–124. Springer, 2023. 5

[She13] Alexander A. Sherstov. Communication lower bounds using directional derivatives. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 921–930, New York, NY, USA, 2013. Association for Computing Machinery. 43

[SHK+12] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.*, 41(5):1235–1265, 2012. 4, 24, 25, 267, 268

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 77–82, New York, NY, USA, 1987. Association for Computing Machinery. 3

[Sud09] Madhu Sudan. Probabilistically checkable proofs. *Commun. ACM*, 52(3):76–84, March 2009. 3

[Sun25] Janani Sundaresan. Optimal communication complexity of chained index. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference, ITCS 2025, Columbia University, New York, NY, USA, January 7-10, 2025*, volume 325 of *LIPIcs*, pages 89:1–89:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. iv, 9, 213

[Suo20] Jukka Suomela. Using round elimination to understand locality. *SIGACT News*, 51(3):63–81, 2020. 4, 23, 163

[SW15] Xiaoming Sun and David P. Woodruff. Tight Bounds for Graph Problems in Insertion Streams. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 435–448, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 15, 17

[Tir18] Sumedh Tirodkar. Deterministic algorithms for maximum matching on general graphs in the semi-streaming model. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 39:1–39:16, 2018. 6, 106, 108

[TV06] Terence Tao and Van H Vu. *Additive combinatorics*, volume 105. Cambridge University Press, 2006. 53

[Vel24] Nate Veldt. Growing a random maximal independent set produces a 2-approximate vertex cover. In Merav Parter and Seth Pettie, editors, *2024 Symposium on Simplicity in Algorithms, SOSA 2024, Alexandria, VA, USA, January 8-10, 2024*, pages 355–362. SIAM, 2024. 5

[Vit01] Jeffrey Scott Vitter. External memory algorithms and data structures: dealing with massive data. *ACM Comput. Surv.*, 33(2):209–271, June 2001. 2

[VY11] Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 11–25, 2011. 15

[Wei15] Omri Weinstein. Information complexity and the quest for interactive compression. *SIGACT News*, 46(2):41–64, 2015. 68, 123

[Wil14] Ryan Williams. Nonuniform acc circuit lower bounds. *J. ACM*, 61(1), January 2014. 3

[WW18] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5), August 2018. 3

[WXX23] Virginia Vassilevska Williams, Yinzhan Xu, and Zixuan Xu. Simpler and higher lower bounds for shortcut sets. *CoRR*, abs/2310.12051, 2023. 64

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227, 1977. 29

[Yeh16] Amir Yehudayoff. Pointer chasing via triangular discrimination. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:151, 2016. 14

[Yeh20] Amir Yehudayoff. Pointer chasing via triangular discrimination. *Combinatorics, Probability and Computing*, 29(4):485–494, 2020. 15, 16, 17, 50

[YV18] Grigory Yaroslavtsev and Adithya Vadapalli. Massively parallel algorithms and hardness for single-linkage clustering under ℓ_p distances. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5600–5609. PMLR, 10–15 Jul 2018. 4

[YYI09] Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. An improved constant-time approximation algorithm for maximum matchings. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 225–234. ACM, 2009. 5, 6

Part IV

Appendices

A Missing Proofs

A.1 Round Elimination for Index: Proof of Claim 3.1.3

Proof of Claim 3.1.3. Distribution of M, i are the same in $\mathcal{D}$ and $\mathcal{D}'$ as Alice and Bob use public randomness to sample them exactly from $\mathcal{D}$, and i is independent of M. However, the distribution of x conditioned on M and i differs in both cases.

In distribution $\mathcal{D}$, the entirety of x is sampled conditioned on M and i. In distribution $\mathcal{D}'$, $\mathsf{x}(\mathsf{i})$ is sampled independent of M and i. Nonetheless, we show that these distributions are close to each other. Let u_2 denote the uniform random variable from $\{0,1\}$.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \|\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) - \mathcal{D}'(\mathsf{x}(\mathsf{i}))\|_{\mathrm{tvd}} \\
&= \mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \|\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) - \mathsf{u}_2\|_{\mathrm{tvd}} \\
&\qquad\qquad \text{(as in ECHO, } b \text{ is chosen uniformly at random from } \{0,1\}) \\
&\le \frac{1}{2} \cdot \mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \sqrt{\mathbb{D}(\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) \mid\mid \mathsf{u}_2)} \qquad \text{(by Fact 4.3.10 Pinskers inequality)} \\
&\le \frac{1}{2} \cdot \sqrt{\mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \mathbb{D}(\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) \mid\mid \mathsf{u}_2)} \qquad \text{(by Jensen's inequality)} \\
&= \frac{1}{2} \cdot \sqrt{\mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{M},\mathsf{i})}. \qquad \text{(by Fact 4.3.4, and as } \mathsf{x}(\mathsf{i}) \text{ is uniform over } \{0,1\} \text{ in } \mathcal{D})
\end{aligned}
$$

We can prove that the mutual information between $\mathsf{x}(\mathsf{i})$ and M,i is small directly.

$$
\begin{aligned}
\mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{M},\mathsf{i}) &= \mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{i}) + \mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{M} \mid \mathsf{i}) \\
&\qquad \text{(by Fact 4.3.1-(7), chain rule of mutual information)} \\
&= 0 + \mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{M} \mid \mathsf{i}) \qquad \text{(as } \mathsf{x}(\mathsf{i}) \text{ is independent of } \mathsf{i}) \\
&= \frac{1}{n} \cdot \sum_{\mathsf{i}=i} \mathbb{I}(\mathsf{x}(\mathsf{i}) ; \mathsf{M} \mid \mathsf{i} = i) \qquad \text{(by definition of conditional mutual information)} \\
&= \frac{1}{n} \cdot \sum_{\mathsf{i}=i} \mathbb{I}(\mathsf{x}(i) ; \mathsf{M} \mid \mathsf{i} = i) \\
&= \frac{1}{n} \cdot \sum_{\mathsf{i}=i} \mathbb{I}(\mathsf{x}(i) ; \mathsf{M}) \qquad \text{(as } \mathsf{i} \text{ is independent of the distribution of } \mathsf{x},\mathsf{M}) \\
&\le \frac{1}{n} \cdot \sum_{\mathsf{i}=i} \mathbb{I}(\mathsf{x}(i) ; \mathsf{M} \mid \mathsf{x}(<i)) \qquad \text{(by Proposition 4.3.2, as } \mathsf{x}(i) \perp \mathsf{x}(<i)) \\
&= \frac{1}{n} \cdot \mathbb{I}(x ; \mathsf{M}) \qquad \text{(by Fact 4.3.1-(7), chain rule of mutual information)}
\end{aligned}
$$

$$\leq \frac{1}{n} \cdot \mathbb{H}(rM) \qquad \text{(as entropy is positive)}$$
$$\leq \frac{s}{n}. \qquad \text{(as the length of messages is upper bounded by } n/100\text{)}$$

This directly gives that:

$$\mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \|\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) - \mathcal{D}'(\mathsf{x}(\mathsf{i}))\|_{\text{tvd}} \leq \frac{1}{2} \cdot \sqrt{\frac{s}{n}}.$$

To prove the claim:

$$\begin{aligned}
&\|\mathcal{D}(\mathsf{x},\mathsf{M},\mathsf{i}) - \mathcal{D}'(\mathsf{x},\mathsf{M},\mathsf{i})\|_{\text{tvd}} \\
&\leq \|\mathcal{D}(\mathsf{M},\mathsf{i}) - \mathcal{D}'(\mathsf{M},\mathsf{i})\|_{\text{tvd}} \\
&\quad + \mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i})\sim\mathcal{D}} \|\mathcal{D}(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i}) - \mathcal{D}'(\mathsf{x}(\mathsf{i}))\|_{\text{tvd}} \\
&\quad + \mathop{\mathbb{E}}_{(\mathsf{M},\mathsf{i},\mathsf{x}(\mathsf{i}))\sim\mathcal{D}} \|\mathcal{D}(\mathsf{x}(-\mathsf{i}) \mid \mathsf{M},\mathsf{i},\mathsf{x}(\mathsf{i})) - \mathcal{D}'(\mathsf{x}(\mathsf{i}) \mid \mathsf{M},\mathsf{i},\mathsf{x}(\mathsf{i}))\|_{\text{tvd}} \qquad \text{(by Fact 4.3.8)} \\
&\leq 0 + \frac{1}{2} \cdot \sqrt{\frac{s}{n}} + 0 \qquad \text{(by our bound, and as the other distributions are the same)} \\
&\leq \frac{1}{2} \cdot \sqrt{\frac{s}{n}}.
\end{aligned}$$

This completes the proof. ∎

A.2 Matchings in Random Bipartite Graphs: Proof of Proposition 9.3.15

In this subsection, we prove Proposition 9.3.15.

Proposition 9.3.15 (restated). *For any large enough integer $k \geq 0$, in a random bipartite graph $G = (L \sqcup R, E)$ where $L = R = \{1, 2, \ldots, k\}$, and E is k edges chosen uniformly at random without repetition from the set*

$$E \subset L \times R \setminus \{(i,i) \mid i \in [k]\},$$

with probability at least $1 - 1/k^2$, there exists a matching in G of size at least $0.1k$.

Proof. We analyze the probability that the maximum matching size is at most $k/10$. As we are working with bipartite graphs, by Konig's theorem, we know that the size of a vertex cover is also $k/10$ when the matching is of size $k/10$. For any specific set $A \subset L \cup R$, we can bound the probability that A is a vertex cover of G. Then, we do a union bound for all possible choices of A.

For any set $A \subseteq L \cup R$, with $|A| \leq ck$, for some constant $0 < c < 1$, we have,

$$\begin{aligned}
&\Pr[A \text{ is a vertex cover}] \\
&= \Pr[(L \setminus A) \times (R \setminus A) \cap E = \Phi] \quad \text{(as there are no edges with both end points out of } A\text{)}
\end{aligned}$$

$$\leq \binom{k^2 - |L \setminus A| \cdot |R \setminus A|}{k} \cdot \frac{1}{\binom{k^2-k}{k}} \quad \text{(no vertex pairs are chosen from } (L \setminus A) \times (R \setminus A))$$

$$\leq \binom{k^2(1-(1-c)^2)}{k} \cdot \frac{1}{\binom{k^2-k}{k}} \quad \text{(as } |L \setminus A| \geq (1-c)k, |R \setminus A| \geq (1-c)k)$$

$$\leq \left(e \cdot (2c - c^2) \cdot k\right)^k \cdot \left(\frac{k}{k^2-k}\right)^k \quad \text{(as } (p/q)^q \leq \tbinom{p}{q} \leq (ep/q)^q \text{ for any } p, q \geq 1)$$

$$\leq (1.1e \cdot (2c - c^2))^k. \quad \text{(as } k^2/(k^2-k) \leq 1.1 \text{ for } k > 11)$$

The total number of choices for set A is,

$$\binom{2k}{1} + \binom{2k}{2} + \ldots + \binom{2k}{ck} \leq ck \cdot \binom{2k}{ck} \leq ck \cdot (2e/c)^{ck},$$

again, as $\binom{2k}{ck} \leq (e \cdot 2k/ck)^{ck}$.

We continue with the union bound.

$$\begin{aligned}
&\Pr[\text{Largest matching in } G \text{ has} \leq 0.1k \text{ edges}] \\
&= \Pr[\text{There exists vertex cover of size } 0.1k] \\
&\leq \sum_{A:|A| \leq 0.1k = ck} \Pr[A \text{ is a vertex cover}] \\
&\leq ck \cdot ((2e/c)^c \cdot 1.1e \cdot (2c - c^2))^k && \text{(by our bounds from earlier)} \\
&\leq 0.1k \cdot (0.85)^k && \text{(for choice of small constant } c = 0.1) \\
&\leq 1/k^2. && \text{(for large values of } k)
\end{aligned}$$

This completes the proof. ∎

A.3 Extending Proposition 5.4.5 to all Graph Sizes

Recall that in the proof of Proposition 5.4.5, the graph G constructed only exists for all integers $\ell \geq 1$, which does not include all large enough integers n as in the statement of the proposition.

To fix this, we simply take the largest possible G with fewer than n vertices and pad each layer with extra vertices to have exactly n vertices overall as required. This padding does not change the parameters p and q. As such, we show that for any choice of large enough number of vertices n there exists a choice of n' that is consistent with some integer ℓ such that G exists and $n \geq n' \geq n/2^{\Theta(\sqrt{\log n})}$. This implies a negligible loss on the bounds of p and q when G is padded.

To that end, in the construction of G, we have that $d^2 = \log \frac{n}{k+1} = d \cdot \log((k+2) \cdot \ell)$ and thus

$$\log(k\ell) \leq d_\ell \leq \log(3k\ell), \text{ and} \tag{A.1}$$

$$\log(k(\ell+1)) \leq d_{\ell+1} \leq \log(3k(\ell+1)) \leq \log(6k\ell) \tag{A.2}$$

where $d_\ell := \log((k+2)\cdot\ell)$ refers to the choice of d that depends only on integer ℓ (since k is fixed). Let n be as in the statement of the proposition where $n = b\cdot(k+1)$ for some large enough integer b since the required DUP graph is a $(k+1)$-layered graph. Fix a choice of ℓ such that

$$b' = ((k+2)\cdot\ell)^{d_\ell} \leq b \leq ((k+2)\cdot(\ell+1))^{d_{\ell+1}} \tag{A.3}$$

where b' also defines the integer $n' = b'\cdot(k+1)$. Then, using the bounds in Eq (A.1), Eq (A.2), and Eq (A.3), we have that

$$\begin{aligned}\frac{n}{n'} = \frac{b}{b'} &\leq \frac{((k+2)\cdot(\ell+1))^{\log(6k\ell)}}{((k+2)\cdot\ell)^{\log(k\ell)}} \leq \left(1+\frac{1}{\ell}\right)^{\log(k\ell)}\cdot((k+2)\cdot(\ell+1))^{\log 6} \\ &\leq \exp\left(\frac{\log(k\ell)}{\ell}\right)\cdot(6k\ell)^3 \leq 2^{\Theta(\log(k\ell))}.\end{aligned}$$

Finally, since $(k\ell)^{\log(k\ell)} \leq ((k+2)\cdot\ell)^{d_\ell} = b' \leq b \leq n$, we have that

$$\log(k\ell) = \sqrt{\log(k\ell)^{\log(k\ell)}} \leq \sqrt{\log n}.$$

This concludes the proof for all graph sizes.

B CONGEST Lower Bounds and Circuit Complexity Barriers

This appendix is a note on the barrier proven by [EFF+22] for proving lower bounds on the number of rounds required to solve triangle detection in CONGEST, and how they may break barriers in circuit complexity. We optimize the parameters a bit further to improve the result from $n^{\Omega(1)}$ lower bound breaking barriers in circuit complexity to $n^{\Omega(1/\log\log n)}$ lower bound breaking the same barriers in a weaker sense, but the framework of these reductions are exactly the same as [EFF+22], and we claim no novelty for these ideas.

Another purpose of this appendix is to keep track of all parameters involved explicitly which leads to the following observation. Even though, strictly speaking, proving an $n^{o(1/\log\log n)}$ lower bound for triangle detection does not break circuit complexity barriers *currently*, it is *plausible* that even proving an $\Omega(\log^c n)$ lower bound for some large constant $c > 1$ could break circuit complexity barriers *assuming* CONGEST algorithms for expander routing can be made to run in $\operatorname{poly}\log(n)$ rounds (see Remark B.4.3)[1].

B.1 Background on Expander Decomposition and Routing

In this section, we state the prior results on expander decomposition and routing in graphs with high conductance in the CONGEST model. We start with some basic definitions and results.

Definitions and terminology. In any graph $G = (V, E)$, we use m to denote the number of edges and n to denote the *total* number of vertices.[2] We use $\deg(v)$ to denote the degree of v.

For any subset $S \subset V$, we use $vol(S)$ to denote $\sum_{v\in S} \deg(v)$. The **conductance of a cut** S and $V \setminus S$ is defined as:

$$\Phi(S) = \frac{|\{(u,v) \mid (u,v) \in E, u \in S, v \in V \setminus S\}|}{\min\{vol(S), vol(V \setminus S)\}}.$$

For the special case $S = \emptyset$, we set $\Phi(S) = 0$. The **conductance of a graph** G is defined as,

$$\Phi = \min_{S\subset V, 0<|S|<n} \Phi(S).$$

A **uniform lazy random walk** in a graph starting from a vertex $s \in V$ performs the following step iteratively: with probability $1/2$, stay at the current vertex; otherwise, pick a uniformly random neighbor of the current vertex and move to that neighbor. In any

[1] To our knowledge, currently, no CONGEST lower bounds rule out such a possibility.

[2] In the other parts of this paper, n referred to the number of vertices in one layer of a tripartite graph. Throughout this appendix, we are not working with tripartite graphs and use n to denote the total number of vertices.

connected graph G, the stationary distribution of a lazy random walk is $\pi(u) = \deg(u)/2m$ for all $u \in V$.

Let $p_t^s(v)$ denote the probability that after $t \geq 0$ steps, a lazy random walk starting from vertex $s \in V$ lands at v for each $v \in V$. Then the **mixing time**, denoted by $\tau_{mix}(G)$ of graph G is defined as the minimum t such that for all $s, v \in V$,

$$|p_t^s(v) - \pi(v)| \leq \pi(v)/n.$$

The following relation between mixing time and conductance is well known [JS89]:

$$\Theta\left(\frac{1}{\Phi}\right) \leq \tau_{mix}(G) \leq \Theta\left(\frac{\log n}{\Phi^2}\right). \tag{B.1}$$

It is also known that the diameter D of any graph is at most the mixing time of the graph.

B.2 Results on Expander Decomposition in CONGEST

An expander is any graph which has large conductance, usually at least $1/\text{polylog}(n)$. We now define the standard expander decomposition, where the vertex set of the input graph has to be partitioned into clusters that form expanders.

Definition B.2.1 (Expander Decomposition)**.** *Given a graph $G = (V, E)$, and parameters $\varepsilon \in (0, 1)$ and $\phi \in (0, 1)$, an expander decomposition of G is a partition of V into sets $V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell$ for some integer $\ell > 0$ such that,*

1. *The subgraph of G induced by each partition V_i for $i \in [\ell]$ is an expander with conductance at least ϕ.*
2. *The number of edges $(u, v) \in E$ such that $u \in V_i$ and $v \in V_j$ for $i \neq j \in [\ell]$, also called* **inter-cluster edges***, is at most $\varepsilon \cdot |E|$.*

Each partition V_i for $i \in [\ell]$ is termed a **cluster** *of the decomposition.*

In the CONGEST model, we want to perform expander decomposition and require that at the end of the decomposition, each node knows the partition that it belongs to.

Definition B.2.2 (Expander Decomposition in CONGEST)**.** *Given a graph $G = (V, E)$ in the CONGEST model, given parameters $\varepsilon, \phi \in (0, 1)$, the expander decomposition in CONGEST problem is to find an expander decomposition $V_1 \sqcup \ldots \sqcup V_\ell$ with parameters ϕ and ε such that each vertex $v \in V$ knows which partition V_i for $i \in [\ell]$ it belongs to.*

We use $T_{\text{exp}}(\varepsilon, \phi, n)$ to denote the minimum number of CONGEST rounds that such a decomposition can be implemented in for all n-vertex graphs.

The following result is known for expander decomposition in CONGEST.

Proposition B.2.3 ([CMGS25, Theorem 1.2])**.** *There is an algorithm in the CONGEST model that solves Definition B.2.2; there exists some constants $c_{\text{COND}}, c_{\text{EXP}} > 1$ such that, for any $\phi \in (0, 1)$,*

$$T_{\text{exp}}(\phi \cdot \log^{c_{\text{COND}}}(n), \phi, n) \leq (\log n)^{c_{\text{EXP}}} \cdot \phi^{-4}.$$

The algorithm is randomized and succeeds with high probability.

It is easy to see that once Definition B.2.2 is solved, each vertex can also identify the number of vertices and the edges inside the cluster that it belongs to.

Claim B.2.4. *In any graph G, after Definition B.2.2 is solved with parameters ε, ϕ, in an additional $O(\log n \cdot \phi^{-2})$ rounds, each vertex $v \in V$ can also know:*

1. *For each $u \in N(v)$, which cluster V_i for $i \in [\ell]$ that u belongs to.*
2. *If $v \in V_i$, the size of cluster V_i, and the total number of edges inside cluster V_i.*
3. *If $v \in V_i$, each vertex also gets an identity from $[|V_i|]$ that is distinct from the identities of all other vertices in V_i, and v knows the identities of all its neighbors that are also present in V_i.*

Proof. To communicate the cluster that each neighbor of v belongs to, one additional round is sufficient, as each vertex $u \in V$ can broadcast to all its neighbors the identity of the cluster u belongs to in $\log n$ bits.

Let $v \in V_i$, let G_i be the subgraph of G induced by V_i, and let D be the diameter of G_i. The vertices can compute a rooted spanning tree of G_i in $O(D)$ time by performing a breadth-first search from each node, and using the identities of the nodes in V for tie-breaking in choosing which breadth-first tree becomes the spanning tree. Once such a spanning tree is computed, the total number of vertices and edges inside G_i can be collected through the edges of the tree in $O(D)$ many rounds. The required identities of vertices in V_i can also be generated by propagating new identities from the root vertex of the spanning tree in $O(D)$ many rounds.

We know that the diameter of the graph G_i is at most $O(\log n \cdot \phi^{-2})$, by Eq (B.1) and the properties of the expander decomposition, which proves the claim. ∎

B.3 Results on Routing Algorithms in CONGEST

We cover the results on routing demand instances from [GKS17], following the presentation of these results in [CPSZ21].

Definition B.3.1 (Routing in CONGEST)**.** *In a graph $G = (V, E)$ in the CONGEST model, an instance of the routing problem is defined as:*

1. *There are a source-destination pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_a, t_a)$ for some $a \in \mathbb{N}$. Each source s_i has to send a message of length at most $O(\log n)$ to vertex t_i for $i \in [a]$. Each vertex $v \in V$ knows all vertices u such that (v, u) is one of the source-destination pairs, and the message v should send to u.*[3]
2. *Each vertex $v \in V$ is the source of at most $\deg(v)$ messages and is the destination of at most $\deg(v)$ messages.*

[3]The vertices need not know any other information about the source-destination pairs.

There may be multiple instances of the routing problem that need to be routed sequentially.

We use a tuple $(P_{\text{rou}}(n), Q_{\text{rou}}(n))$ to denote the runtime of an algorithm in CONGEST that on n-vertex graphs, uses $P_{\text{rou}}(n) \cdot \tau_{mix}$ rounds of preprocessing steps, and after these steps, can route each instance of the routing problem in $Q_{\text{rou}}(n) \cdot \tau_{mix}$ time, where τ_{mix} is the mixing time of the underlying graph G.

Let us state one existing routing algorithm that we employ. This way of presenting the results of [GKS17] can be found in Section 2.1 of [CPSZ21].

Proposition B.3.2 (cf. Lemma 3.2, 3.3 and 3.4 of [GKS17])**.** *There exists constants $c_{\text{P}}, c_{\text{Q}} > 1$ such that, for any integer $k \geq 1$, there exists a routing algorithm with:*

$$P_{\text{rou}}(n) \leq k \cdot n^{4/k} \cdot (\log n)^{c_{\text{P}} \cdot k} \cdot O(1),$$
$$Q_{\text{rou}}(n) \leq (\log n)^{c_{\text{Q}} \cdot k} \cdot O(1),$$

that succeeds with high probability. Note that there is also a τ_{mix} factor in the total number of rounds based on the definitions in Definition B.3.1.

Another routing algorithm is from [GL18], which is better than the ones we state from [GKS17] for a certain ranges of number of instances of routing, although it does not admit a tradeoff in preprocessing and routing requests.

To conclude our background section, we summarize the problems we consider and the functions corresponding to their runtimes. This may help the reader keep track of these functions while perusing the next subsection.

Function	Definition
$T_{\text{exp}}(\varepsilon, \phi, n)$	The minimum number of rounds taken by a CONGEST algorithm that finds an expander decomposition with parameters ε, ϕ of the input n-vertex graph.
$P_{\text{rou}}(n)$	In the routing problem, $P(n) \cdot \tau_{mix}$ is the total number of rounds required for preprocessing.
$Q_{\text{rou}}(n)$	In the routing problem, after pre-processing, $Q(n) \cdot \tau_{mix}$ is the total number of rounds required to handle one instance of the routing demand problem.

B.4 From Circuits to CONGEST Algorithms

In this section, we will show how CONGEST lower bounds can also imply circuit lower bounds for an explicit function. For $n, m \geq 1$, let $f_{n,m} : \{0,1\}^{2m \log n} \to \{0,1\}$ denote the function that takes as input a representation of a graph $G = (V, E)$ where each edge is given by $2 \log n$ bits denoting the identities of the two end points of the edge in $\log n$ bits each. The function outputs 1 if the input graph has a triangle, and outputs 0 otherwise.

We show that we can simulate a circuit that computes $f_{n,m}$ in the CONGEST model.

Lemma B.4.1. *Given a family of circuits $\mathcal{C} = \{C\}_{n,m\geq 1}$ that compute function $f_{n,m}$ with constant fan-in and fan-out, depth $d(n,m)$ and total number of gates at most $m \cdot s(n,m)$, for any graph $G = (V,E)$, for any parameters $\varepsilon \in (0,1/3)$ and $\phi \in (0,1)$, there is an algorithm in CONGEST that solves triangle detection in*

$$\left(T_{\exp}(\varepsilon,\phi,n) + O(\log n \cdot \phi^{-2}) \cdot (P_{\text{rou}}(n) + Q_{\text{rou}}(n) \cdot s(n,m) \cdot d(n,m))\right) \cdot O\left(\frac{\log n}{\log(1/3\varepsilon)}\right)$$

many rounds.

Proof. First, we perform expander decomposition on graph G with parameters ε, ϕ in $T_{\exp}(\varepsilon,\phi,n)$ many rounds. Each cluster performs the following procedures for simulating circuits in parallel.

Choosing the circuit. We use the result in Claim B.2.4 to make sure each vertex in any cluster V_i knows all the information listed in the statement of Claim B.2.4. We use $new(v)$ to denote the identities from Claim B.2.4.

In each cluster V_i, a vertex $v \in V_i$ is said to be a **good node** if the number of neighbors of v inside cluster V_i is at least as many as the number of neighbors of v outside cluster V_i. All other vertices are called **bad nodes**.

We let graph G_i be the subgraph of G with the union of all the edges incident on the good nodes of V_i (including ones which go out of V_i), and all the edges between two vertices in V_i. Using an argument similar to the one in Claim B.2.4, the total number of these edges can be known to all vertices in V_i in $O(\log n \cdot \phi^{-2})$ rounds. Let m_i be the total number of these edges.

Each cluster V_i simulates the circuit C_{n,m_i}, and each vertex in cluster V_i knows the value of n and m_i. Hence, all the vertices in the cluster know which circuit to simulate.

Assignment of gates to vertices. Each gate g of the circuit C_{n,m_i} is assigned to one of the nodes in V_i that is responsible for computing the output of gate g, as well as sending this output to all the gates in the next layer of the circuit that g feeds into. The assignment of gates is as follows:

(*i*) Each gate g of the input layer corresponds to some edge (u,v) of graph G. If both $u, v \in V_i$, all these $2\log n$ gates are assigned to either u or v, lexicographically breaking ties based on $new(u)$ and $new(v)$. Moreover, which gate among the input layer corresponds to which edge can be fixed by arguments similar to the proof of Claim B.2.4: the root of the spanning tree can communicate to each of its neighbors a partition of the input gates corresponding to the number of edges in each of the subtrees, and this can be done recursively. This will take $O(\log(n) \cdot \phi^{-2})$ rounds, also based on the upper bound on the diameter of the cluster.

(*ii*) Each gate in any layer which is not the input layer, is assigned to a vertex in V_i so that the total number of gates in any layer given to any vertex v is at most $2 \cdot s(n,m_i) \cdot \deg_i(v)$, where $\deg_i(v)$ is the total number of neighbors of v inside the cluster as follows:

(a) The gates assigned to each vertex v form $2 \cdot s(n, m_i)$ many disjoint contiguous segments in layer j.

(b) There is a partition of $[m_i]$ into $|V_i|$ many contiguous segments, and each segment is given to a vertex $v \in V_i$. Let $[g_{v,1}, g_{v,2}]$ be the segment given to vertex v, and we require that $g_{v,2} - g_{v,1} + 1 = \deg_i(v)$.

(c) All the gates numbered $(x-1) \cdot (m_i) + y$ is assigned to vertex v for each $x \in [2 \cdot s(n, m_i)]$, $g_{v,1} \leq y \leq g_{v,2}$ if such a gate exists.

The total number of gates in each layer is at most $m_i \cdot s(n, m_i)$, so, all the gates are assigned to some vertex. Each vertex v can also be made to know the integers $g_{v,1}$ and $g_{v,2}$ with arguments similar to assigning gates in the input layer: the root picks a contiguous segment of length equal to its degree for itself starting from 1, then partitions the rest of the gates among its subtrees based on the number of edges. This step is performed recursively in each subtree. Totally in $O(\log(n) \cdot \phi^{-2})$ rounds, each vertex v knows $g_{v,1}$ and $g_{v,2}$ and hence, all the gates that are assigned to it.

Next, we show how the vertices can communicate to each other the output of the gates they are assigned to.

Routing gate outputs. The vertices simulate the circuit layer by layer starting from the input layer. We prove that this simulation is possible by induction. For the base case, we know that the value of all the gates in the input layer is known to the vertices to which the gates are assigned to trivially. For the induction hypothesis, let us assume that the output of all the gates in layer $j-1$ is known to the vertices that they are assigned to. We will show that the output of all these gates in layer $j-1$, once found, can be routed to the vertices that are assigned the gates in layer j that they feed into.

Only for the purpose of routing the output of gates in each layer, we assume that each vertex $v \in V_i$ has $\deg_i(v)$ many identities: all integers from $g_{v,1}$ to $g_{v,2}$ (including $g_{v,1}$ and $g_{v,2}$). The identities of each vertex can be specified by $O(\log m_i)$ bits $g_{v,1}, g_{v,2} \leq m_i$. The vertices can send all their identities to their neighbors in 1 additional round.

The vertices use the preprocessing routine for the routing algorithm, which can be run in $P_{\text{rou}}(n)$ rounds. Next, for each gate g, the vertex v that g is assigned to needs to send the output of g to $O(1)$ many vertices, which are assigned the gates in layer j that g feeds into. The identities of these vertices are known to v as it knows circuit C_{n,m_i} (if the gate number is g, $(g-1) \mod m_i + 1$ is the identity of the vertex that g is assigned to) and the output of each gate is also only 1 bit. The total number of gates assigned to each vertex v is at most $s(n, m_i) \cdot 2 \cdot \deg_i(v)$, this can be viewed as $O(s(n, m_i))$ many instances of the routing problem, each of which takes $Q_{\text{rou}}(n) \cdot O(\log n \cdot \phi^{-2})$.[4]

[4]Here, we assign multiple identites to each vertex, to have a total of $\deg_i(v)$ many identites each, and we claim the routing algorithm can still be run correctly. This can be easily seen by performing the routing algorithm on another graph with m_i vertices: a replacement product of the original graph in cluster V_i with a family of $O(1)$-regular low-degree expanders of conductance $\Omega(1)$. Each vertex in the new graph will have a unique identity from $[m_i]$, and any round of a CONGEST algorithm can be simulated in the original graph in cluster V_i. Refer to Section 4 and Proposition 4.2 of [ASW19] for details on replacement products and the fact that the mixing time increases by at most an $O(1)$ factor, respectively.

This routing process is repeated till the last layer of the circuit, and if a triangle is found in graph G_i the vertex with the output gate of the circuit outputs the existence of the triangle, and the algorithm terminates.

Recursion. If there is no triangle found in any of the clusters, all the edges (u, v) where u, v are two good nodes inside the same cluster, i.e., $u, v \in V_i$ for some $i \in [\ell]$ are removed. (Note the definition of a good node is with respect to the cluster it belongs to.) First, we argue that we remove at least $(1 - 3\varepsilon) \cdot m$ edges.

We know by the parameters on the expander decomposition that the total number of inter-cluster edges is at most $\varepsilon \cdot m$. We will argue that each edge (u, v) with $u, v \in V_i$ that is not removed can be charged to an inter-cluster edge, so that each inter-cluster edge receives at most two charges. This will prove that the total number of edges left is at most $3 \cdot \varepsilon \cdot m$. Let $(u_1, v), (u_2, v), \ldots, (u_a, v)$ be all the edges incident to some bad node v in cluster V_i with $u_j \in V_i$ for all $j \in [a]$. We know that v has at least a many edges to vertices outside cluster V_i by the definition of bad nodes. We charge each of these edges to one such inter-cluster edge incident on v, so that each inter-cluster edge receives at most one charge. Globally, each inter-cluster edge (x, y) receives at most two charges, as it receives at most one charge from each of its end-points.

We have shown that at least half the edges are removed, and we recurse on the remaining network by computing another expander decomposition and circuit simulation. This recursion is done till there are only $O(1)$ edges left in the graph, wherein the existence of a triangle can be detected trivially in $O(1)$ rounds. The total number of levels of recursion is at most $\log m/(\log(1/3\varepsilon))$.

Correctness. It is easy to see that if the algorithm outputs the existence of a triangle, there is a triangle in the graph, as the outputs of the circuits are correct. The edges we remove at each level of the recursion are incident on two good nodes from the same cluster V_i. The only way such an edge (u, v) can be a part of a triangle is if there exists some vertex $w \in V$ with $(u, w), (v, w) \in E$. However, when the circuit is simulated in V_i, the edges (u, w) and (v, w) would be included as a part of the input, as u, v are good nodes. Hence the existence of a triangle would be detected by the circuit. Therefore, when the algorithm declares that there is no triangle after all the levels of the recursion, no triangle exists in the graph.

Total number of rounds. The number of levels of the recursion is $O((\log n)/(\log(c_2/3)))$. Assuming $T_{\text{exp}}(\varepsilon, \phi, n), P_{\text{rou}}(n), Q_{\text{rou}}(n), s(n, m)$ and $d(n, m)$ are increasing functions in n, m, each level of the recursion takes:

(*i*) $T_{\text{exp}}(\varepsilon, \phi, n)$ rounds to compute the expander decomposition.

(*ii*) $O(\log n \cdot \phi^{-2})$ rounds to propagate the required information to all the nodes in each cluster before circuit simulation.

(*iii*) $P_{\text{rou}}(n) \cdot O(\log n \cdot \phi^{-2})$ pre-processing time before circuit simulation via routing requests (as $\tau_{mix} = O(\log n \cdot \phi^{-2})$).

(*iv*) $O(s(n,m)) \cdot Q_{\text{rou}}(n) \cdot O(\log n \cdot \phi^{-2})$ rounds to compute one layer of the circuit, repeated for $d(n,m)$ many layers.

This completes the proof. ∎

The following result now establishes the barrier for proving CONGEST lower bounds.

Proposition B.4.2. *If there exists a large enough constant $c > 0$ and integer n_0, such that for all $n \geq n_0$ triangle detection in CONGEST requires $n^{c/\log\log n}$ rounds, then there is an explicit function $f : \{0,1\}^N \to \{0,1\}$ such that the following is true. There exists integer N_0 such that for all $N \geq N_0$, there exists no family of circuits computing f_N with depth $O(\log N)$, constant fan-in, constant fan-out and size $N^{1+c'/\log\log N}$ for some constant c' depending on c only.*

Proof. The proof is a direct application of Lemma B.4.1 and the results of Propositions B.2.3 and B.3.2.

Firstly, in Proposition B.2.3, we set $\phi = \log^{-c_{\text{COND}}}(n)/6$. This gives:

$$\begin{aligned} &T_{\text{exp}}(\phi \cdot \log^{c_{\text{COND}}}(n), \phi, n) \leq (\log n)^{c_{\text{EXP}}} \cdot \phi^{-4} \\ &\implies T_{\text{exp}}(1/6, (\log n)^{-c_{\text{COND}}}, n) \leq (\log n)^{c_{\text{EXP}}+4c_{\text{COND}}}. \end{aligned}$$

This also sets the mixing time of the clusters as $O(1) \cdot (\log n)^{2c_{\text{COND}}+1}$, by Eq (B.1).

Next, we set parameter k in Proposition B.3.2 to be $k = \log\log n/100(c_{\text{Q}} + c_{\text{P}})$. This gives:

$$\begin{aligned} &P_{\text{rou}}(n) \leq O(\log\log n) \cdot (n^{400(c_{\text{Q}}+c_{\text{P}})/\log\log n} \cdot (\log n)^{\log\log n}) \leq n^{(2+400(c_{\text{Q}}+c_{\text{P}}))/(\log\log n)}, \\ &Q_{\text{rou}}(n) \leq O(1) \cdot (\log n)^{\log\log n} \leq n^{1/\log\log n}, \end{aligned}$$

for large enough n.

Let $N := 2m\log n$, and f_N be the function that takes as input m edges, each represented by $2\log n$ bits for both the end-points, and computes whether there is a triangle in the input graph. Given a circuit of depth $O(\log N) = O(\log n)$, and size $s(n,m)$, by Lemma B.4.1, there is an algorithm that solves triangle detection in CONGEST in

$$\begin{aligned} &(T_{\text{exp}}(1/6, (\log n)^{-c_{\text{COND}}}, n) + O(\log^{2c_{\text{COND}}+1} n) \cdot (P_{\text{rou}}(n) + Q_{\text{rou}}(n) \cdot s(n,m) \cdot d(n,m))) \cdot O(\log n) \\ &= n^{O(1/\log\log n)} \cdot s(n,m). \qquad \text{(by our equations above, and depth } O(\log n)) \end{aligned}$$

By our premise, we get $s(n,m) \geq n^{c'/\log\log n}$ for some constant c', which is a function of c. This proves that the size of the circuits is at least $m \cdot n^{c'/\log\log n} \geq m^{1+\Omega(1/\log\log m)}$, for an explicit function on at least $N = \Theta(m\log m)$ bits. ∎

Barrier aspect of Claim B.4.2. We can now provide the main message of this appendix. Claim B.4.2 implies a *barrier* because of the following: the current lower bounds in circuit complexity on the size of a circuit with constant fan-in and fan-out computing an explicit function on N bits using arbitrary Boolean operator gates in $O(\log N)$ depth is $(3+1/86)N-$

$o(N)$ (see [GKW21]). Obtaining any superlinear lower bound on the size of such circuits has been a major open problem in complexity theory for decades. A lower bound of $n^{\Omega(1/\log\log n)}$ rounds in CONGEST for triangle detection then implies an $N^{1+\Omega(1/\log\log N)}$ size lower bounds for an explicit problem (detecting a triangle) on above circuits, way above the longstanding $\Theta(N)$ threshold of existing lower bounds.

Remark B.4.3. *Claim B.4.2 identifies $n^{\Omega(1/\log\log n)}$ rounds as a barrier for proving distributed CONGEST lower bounds for triangle detection. However, the arguments here can imply even more.*

It is plausible that the runtime of the expander routing in Definition B.3.1 (which is the bottleneck here) can be improved much beyond what is currently known. In fact, it is possible that these problems can be solved in $\mathrm{polylog}(n)$ *rounds in CONGEST. Thus, in absence of a lower bound that rules out possibility of such an algorithm for expander routing, any lower bound for distributed triangle detection of the form $\Omega(\log^c(n))$ rounds, for some sufficiently large constant $c > 1$, has the* potential *of breaking current circuit complexity barriers.*

As a result, we believe that at this stage, namely in absence of a concrete lower bound on Definition B.3.1 for the parameters mentioned earlier, the limit *of one can realistically hope for an unconditional lower bound for distributed triangle detection is* $\mathrm{polylog}(n)$ *and even $\Omega(\log n)$ rounds.*