Contextual AI: Integrating Macro-Context with Transformer Architectures for Social Media Analysis, Federated Learning, and Recommender Systems

by

Liam Peter Hebert

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

© Liam Peter Hebert 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:	Aijun An Professor, Dept. of Electrical Engineering and Computer Science York University
Supervisor(s):	Robin Cohen Professor, Cheriton School of Computer Science, University of Waterloo
	Lukasz Golab Professor, Dept. of Management Science and Engineering, Cheriton School of Computer Science (cross-appointment) University of Waterloo
Internal Member:	Jesse Hoey Professor, Cheriton School of Computer Science, University of Waterloo
	Freda Shi Asst. Professor, Cheriton School of Computer Science, University of Waterloo
Internal-External Member:	Olga Vechtomova Professor, Dept. of Management Science and Engineering Cheriton School of Computer Science (cross-appointment) 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

Research Presented in Part I: This part contains work expanded from papers I published on Discussion Transformers. Chapter 4 is based on Hebert et al. [2024a], with co-authors Gaurav Sahu, Yuxuan Guo, Nanda Kishore Sreenivas, Lukasz Golab and Robin Cohen. The contribution of Graphormer in section 3.4 is based on Hebert et al. [2022a], with co-authors Lukasz Golab and Robin Cohen. The qualitative analysis of Graphormer in section 3.5 is based on Hebert et al. [2023a], with co-authors Hong Yi Chen, Robin Cohen and Lukasz Golab.

Research Presented in Part II: This part contains work expanded from Hebert et al. [2023b]. This work was created in collaboration with co-authors Lukasz Golab, Pascal Poupart and Robin Cohen.

Research Presented in Part III: This part contains work expanded from Hebert et al. [2025]. This work originated from my research internship at Google Research, with co-authors Marialena Kyriakidi, Hubert Pham, Krishna Sayana, James Pine, Sukhdeep Sodhi and Ambarish Jash.

Research Presented in Part IV: While this chapter does not contain content from published work, some of the ideas in section 9.1 are inspired from work I published in Hebert et al. [2024b] (co-authors Krishna Sayana, Ambarish Jash, Alexandros Karatzoglou, Sukhdeep Sodhi, Sumanth Doddapaneni, Yanli Cai, Dima Kuzmin), Sayana et al. [2025] (co-authors Krishna Sayana, Raghavandra Vasudeva, Yuri Vasilevski, Kun Su, James Pine, Hubert Pham, Ambarish Jash, Sukhdeep Sodhi), and Su et al. [2025] (co-authors Kun Su, Krishna Sayana, Hubert Pham, James Pine, Yuri Vasilevski, Raghavendra Vasudeva, Marialena Kyriakidi, Ambarish Jash, Anushya Subbiah, Sukhdeep Sodhi).

In all of the works above, except for Sayana et al. [2025], Su et al. [2025], I was the primary author and led the conceptualization, development, experimentation, and manuscript preparation. In recognition of the valued contributions of my co-authors, without whom this thesis would not be possible, the rest of this thesis will be written using "we" instead of "I". Despite this, the reader should not be confused as to the independent authorship of this thesis.

Abstract

Context is crucial for understanding the world and making informed decisions. While existing transformer architectures excel at contextualizing information locally, such as other words in a sentence, they fail to factor in broader, macro-level contexts. We identify available yet underutilized macro contexts in three use cases: online discussions, federated learning, and recommender systems. For each, we motivate the need to leverage existing macro context and propose context-aware solutions based on the transformer architecture.

In online discussion boards, the rich conversational and multimodal macro context in which a comment is made is often overlooked. This is especially pertinent in hate speech detection. Classical solutions that examine individual comments in isolation fail to account for this context, leading to ambiguity and misinterpretation. For instance, the comment "Ew, that's gross!" has a different interpretation depending on whether it's in response to food or sensitive issues like LGBTQ rights. Furthermore, images that accompany text can also provide crucial context. We propose mDT, a novel deep learning model architecture based on graph transformer networks, which incorporates this valuable context when evaluating the hatefulness of individual comments. Our experimental results demonstrate a 7% F1 improvement over existing baselines that do not utilize this context, and a 21% F1 improvement over previous graph-based methods.

Second, we tackle the context-agnostic paradigm of federated learning. The prevalent Federated Averaging (FedAvg) method statically averages model weights, failing to account for the crucial macro-level context of heterogeneous-agent environments, leading to a suboptimal, one-size-fits-all model. For example, autonomous driving agents exploring rural roads acquire different knowledge than those in urban settings, and this environmental context is lost in the process. We propose FedFormer, a novel federation strategy that leverages transformer attention to enable each agent to weigh and selectively incorporate insights from its peers in a context-dependent manner. In turn, FedFormer enables a more effective, efficient federation that respects and adapts to environmental diversity while preserving privacy. Our experiments across environments in MetaWorld, a set of heterogeneous robotic manipulation tasks, demonstrate improvements of 1.48x to 3.41x over FedAvg.

Finally, in recommender systems, the user's intent can provide critical personalization context. Simple approaches rely on collaborative filtering, which only models implicit (micro-level) user preferences by extrapolating from historical data. Our solution, Flare, proposes a contextual recommender system that empowers users to steer recommendations via explicit natural language queries (e.g., "Staplers", "Webcams"). Flare's architecture

fuses collaborative filtering signals with semantic representations of both the user's explicit query and item descriptions, bridging the gap between long-term preferences and the context of the user's immediate goals. Our experiments using the Amazon Product Reviews datasets show a 1.7x and 2.53x increase in recall@1 and recall@10, respectively, compared to approaches that do not factor in user intent.

Acknowledgements

As with the topic of this thesis, my academic journey has been shaped by many macro factors. All that I have learned and accomplished has been a product of the incredible mentors, friends, and family I have had the fortune of surrounding myself with. While my name may appear alone on the front page of this thesis, I firmly believe that the pages would have no ink if it were not for their influence.

First, I would like to thank my supervisors, Prof. Robin Cohen and Prof. Lukasz Golab, for their guidance throughout my time at Waterloo. Robin is the kindest, patient, and most thoughtful person I have ever met, and truly embodies the meaning of "academic parent". Throughout my entire degree, Robin expertly mastered the delicate art of pushing me to reach heights beyond my wildest imagination while also keeping me sane. No email was too small, no request too silly, no timeline too short[1]. I will miss waking up to 10+ emails almost every morning, full of ideas, edits, and reminders about the many things I was involved in, interspersed with commentary on baseball, soccer, and trivia facts. I hope I get to keep the second half post-graduation.

I am particularly thankful to Lukasz for teaching me how to become a better writer and communicator. Among the many iconic "Lukasz-isms" his students share, the most impactful one, which I have been told many times, is "What is the problem space?" While this might sound simple, his keen, persistent focus on the bigger picture has forever shaped my view of research. His feedback, while at times direct, is emblematic of his high expectations (and belief!) of his students' ability to achieve great things, the greatest form of respect.

I would also like to thank my readers, Prof. Jesse Hoey, Prof. Freda Shi, Prof. Olga Vechtomova and Prof. Aijun An, for carefully reading and evaluating my thesis. Beyond providing useful feedback on the final thesis, my readers also influenced many of the ideas that went into it. Namely, Prof. Hoey's affective computing course led to the creation of Part I of this thesis and Prof. Shi's outstanding PhD thesis was a key inspiration for the structure of this thesis, beyond Prof. Shi's ideas on grounding. I also had the pleasure of engaging in great conversations with Prof. Vechtomova about applications of my work beyond the domains I explored, which motivated many ideas in Part IV of this thesis.

I also gratefully acknowledge the scholarships that funded this research. I want to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for

[1] One of my fondest, type-2 fun, memories with Robin is staying up late up to the "anywhere-on-earth" deadline (read: 5 am) for a conference submission, doing last-minute edits. Despite the crunch, Robin stayed on the phone with me the entire time. That paper, present in Chapter 2, led to many of the ideas that constitute this thesis.

supporting my research through the Vanier Canada Graduate Scholarship and the CGS-M Scholarship. NSERC's strong support for research is among the many reasons I am proud to be Canadian. I would like to thank the University of Waterloo for its support through the David R. Cheriton Scholarship and the President's Scholarship. I also want to thank the Nick Cercone scholarship for their support. While I did not have the chance to meet Prof. Nick Cercone, I strive to embody his passion for research and mentorship. I am honoured to receive this scholarship created in his legacy.

I have not personally met these people, but thank you to Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser and Illia Polosukhin for writing a pretty nifty paper called "Attention Is All You Need", which has been my obsession over the last five years.

I have also been fortunate to collaborate with many academic colleagues at the University of Waterloo, many of whom have become close personal friends. Thank you, Kathryn Carbone (still think your paper should have been called MedGeese!), Olha Wloch (still happy you kept my idea to name your thesis GASTON after your foster cat!), Anastasiia Avksientieva, Moein Shirdel, Amy Tai, Joel Rorseth, Gaurav Sahu, Owen Chambers and Cailum Stienstra. My many brainstorming discussions with Cailum in our small kitchen, using a small whiteboard, about how to apply Graph Transformers to chemical spectra will be fond memories for me.

I would also like to thank David Radke for the many enjoyable memories of bumping into each other at conferences and for convincing me to pursue a PhD. I also had the honour of working with Prof. Scott Hopkins and Prof. Pascal Poupart, for which I am very grateful. My time at Waterloo also led me to work with many brilliant undergraduate research assistants, whose efforts inspired me throughout this thesis. Thank you to Hong Yi Chen, Yuxuan Guo, Wen Zhang, Sunil Cottril, Max Hao, Ray Cao, Emily Zhang and Anna Xing. It was an utmost pleasure working with and learning from them. Thank you to my undergraduate honours thesis supervisors at Dalhousie University, Prof. Axel Soto, Prof. Evangelos Milios, Prof. Stephan Brooks and Raheleh Makki, for introducing me to AI research. Prof. Soto's lecture on RNNs inspired me to pursue AI.

As part of my academic career, I have also had the privilege of undertaking industrial internships at Gracenote, Kinaxis, Twitter, and Google Research. These internships shaped the technical skills that enabled my research and my love for code. Among the many co-workers I had the privilege of learning from, I would like to give special thanks to Clifford Wheadon, Rahaleh Makki, Shubhanshu Mishra, Krishna Sayana, Hubert Pham, Marialena Kyriakidi, Ambarish Jash and James Pine. Clifford, Hubert, Krishna and James are true programming wizards, and their wisdom taught me that code should not just be

functional but also beautiful.

Perhaps to the surprise of many, I also had a life outside of academics and work. I want to thank Savina Tam for being an incredible roommate, Fenn Tam for being an incredible cat, Keira McManus for hosting amazing murder mystery parties, and Katerina Coveny for introducing me to birding and many fun bird facts. I would also like to thank the close friends I made during my undergrad (the boys in motion), Arjav Gupta, Aurinjoy Gupta, Beth Eddy, Logan Sutherland, Lauryn Marchand, Florencia Capouyá Tacain, Jessica Nguyen, and Kassidy Walker. I love you guys very much.

Special thank you to Dr. Sanjoy Gupta for being willing to share your passion for Ophthalmology with me and for bringing two amazing people, Arjav and Aurinjoy, into my life. I am very fortunate to have met you, and I hope to carry your kindness and passion for research with me in your memory.

I also want to thank my brothers for life, Dardan Loshi, Kai Trappenburg and Michael El-Cheik. I could write multiple pages (perhaps a whole thesis) about the many fun adventures we have had together. We have laughed, we have cried, we have supported each other through breakups, we have conquered mountains of sushi, and so much more. The many nights at Kai's place during undergrad playing the Gloomhaven boardgame[2], drinking lots of coffee (trentas if you are Dardan) and eating donuts (a Mikey specialty), will forever be some of the happiest nights of my life. Kai, I will never forgive you for stealing 20+ coins in a single round when you played as the Mindthief character. I look forward to more such nights for many, many years to come.

Lastly, I would like to thank my grade 12 math teacher, Mme. Line, whose advice of "Liam should consider a university degree outside of math" gave me all the motivation I needed to do the most petty thing ever.

[2]Thank you, Issac Childres, for making an incredible boardgame.

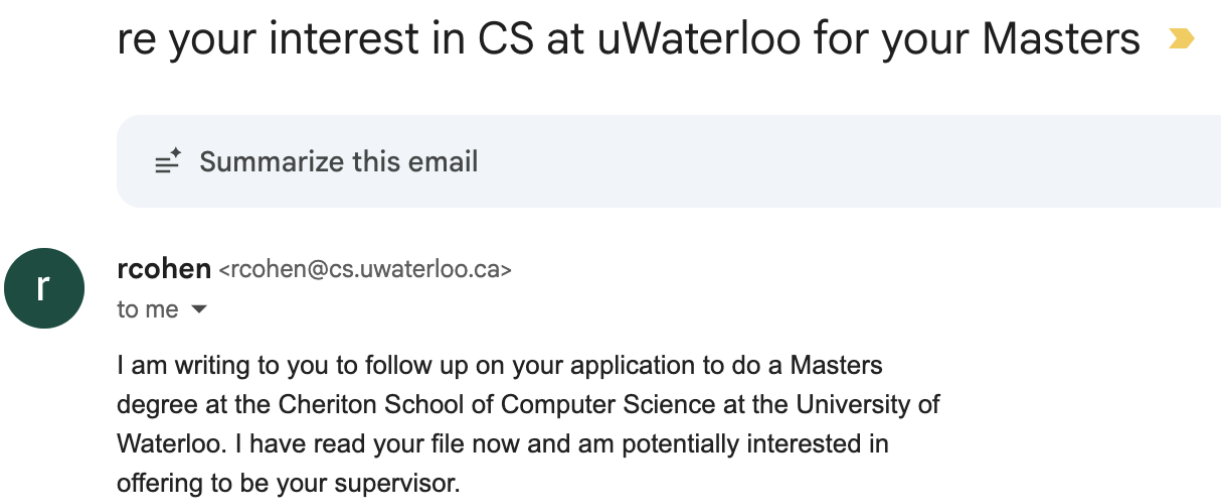


Figure 1: The email that changed my life

Dedication

To my parents, Peter and Mary, and my sister, Sarah, for your love, support and for always keeping me grounded in the bigger picture.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vii

Dedication xi

List of Figures xvii

List of Tables xx

1 Introduction 1

1.1 Guide to the Thesis and Research Contributions 3

2 Background 6

2.1 Recurrent Neural Networks 6

2.2 Attention 7

2.3 Transformers 10

2.3.1 Encoder-Only Pre-Trained Transformers . . . 12
2.3.2 Pre-training Transformer Models . . . 15

I Context of Social Media Conversations 17

3 Background and Roadmap 18
3.1 Roadmap . . . 18
3.2 Hate Speech Detection . . . 19
3.3 Graph Neural Networks . . . 20
3.4 Guiding Study of Graph Transformers for Predictive Hate Speech Detection 21
3.4.1 Methods . . . 24
3.4.2 Findings . . . 28
3.4.3 Takeaways . . . 32
3.5 Extended Qualitative Study of Graphormer . . . 33
3.5.1 Baselines . . . 33
3.5.2 Experimental Setup . . . 34
3.5.3 Findings . . . 34
3.5.4 Discussion . . . 44
3.5.5 Takeaways . . . 46
3.6 Multi-Modal Hate Speech Detection . . . 47

4 Discussion Transformers: Integrating text, images and graph transformers for Holistic Discussion Analysis and its Applications for Hate Speech Detection 49
4.1 Methodology . . . 51
4.1.1 Discussion Transformers . . . 51
4.1.2 Multi-Modal Discussion Transformer (mDT) . . . 52
4.1.3 HatefulDiscussions Dataset . . . 56

4.2 Results . . . 59
4.2.1 Experimental Setup . . . 59
4.2.2 Text-only Methods vs. Discussion Transformers . . . 59
4.2.3 Effect of Multi-Modal Bottleneck Size . . . 61
4.2.4 Effect of Constrained Graph Attention . . . 62
4.2.5 Effect of Fusion Layers . . . 62
4.2.6 Effect of Images . . . 63
4.2.7 Qualitative Analysis: BERT vs. mDT . . . 63
4.3 Future Work . . . 67
4.4 Conclusion . . . 68

II Context of Federated Agents 70

5 Background and Roadmap 71
5.1 Roadmap . . . 71
5.2 Reinforcement Learning . . . 72
5.3 Federated Learning . . . 75
5.4 Related Work . . . 77

6 FedFormer: Contextual Federation with Attention in Reinforcement Learning 78
6.1 Methods . . . 80
6.2 Results . . . 82
6.2.1 Experimental Design . . . 82
6.2.2 Meta-World Evaluation . . . 85
6.2.3 Scalability . . . 86
6.2.4 Agent Onboarding . . . 88
6.2.5 Ablation Study . . . 89
6.3 Conclusion . . . 90

III Context of Sequential Recommenders **93**

7 Background and Roadmap **94**

7.1 Roadmap . . . 94
7.2 Collaborative and Content-Based Filtering . . . 95
7.2.1 Collaborative Filtering . . . 96
7.2.2 Content-Based Filtering . . . 97
7.3 Sequential Recommender Systems . . . 98
7.4 Text-only Large Language Model Recommenders . . . 99
7.5 Interactive Recommender Systems and Critiquing . . . 102

8 FLARE: Fusing Language Models and Collaborative Architectures for Recommender Enhancement **103**

8.1 Method . . . 106
8.1.1 Task Definition . . . 106
8.1.2 Flare Model Architecture . . . 106
8.1.3 ID-Text Contrastive Learning . . . 108
8.1.4 Critiquing . . . 109
8.1.5 Training . . . 111
8.2 Experimental Results . . . 112
8.2.1 Experimental Setup . . . 112
8.2.2 Flare Performance on Existing Benchmarks . . . 118
8.2.3 Performance on Large Datasets . . . 122
8.2.4 Critiquing . . . 123
8.2.5 Out-of-Domain Critiquing . . . 124
8.2.6 Qualitative Study of Critiquing . . . 126
8.2.7 Ablation Study . . . 129
8.3 Conclusion and Future Work . . . 132

IV Putting Everything into Context **134**

9 Common Bonds and Future Directions **135**

9.1 Addressing Data Sparsity through Pre-Training 135

9.1.1 Pre-Training for Discussion Transformers 136

9.1.2 Pre-Training for Hybrid Recommenders 139

9.2 Addressing Latency by Leveraging the Context of Context 142

9.2.1 Casual KV Caching Through Context Metadata 143

9.2.2 Mixture-of-Federated Agents to Accelerate Federation Transformers 144

10 Conclusion **146**

10.1 Key Findings and Contributions of this Thesis 146

10.1.1 New Paradigm for Contextual Modelling 147

10.1.2 New Context-Aware Architectures based on the Transformer 148

10.1.3 Methodological and Empirical Contributions 149

10.2 Limitations 151

10.3 The Bigger Picture 152

10.4 Concluding Thoughts 153

References **155**

List of Figures

1 The email that changed my life x

1.1 Across three domains, this thesis shifts the unit of analysis from a local, micro-context lens (inner dashed circles) to a holistic, macro-context lens (outer solid circles). 2

2.1 The seq2seq architecture. Here, the RNN model first reads the input sequence 'ABC', then produces 'WXYZ'. The model switches to decoding when receiving the 'end-of-sequence' token. Figure taken from Sutskever et al. [2014] 7

2.2 RNN additive attention. The inputs are first encoded using a bidirectional RNN to create h_i. Then, the state s_t is created as a product of s_{t-1} and an attention weighted $\alpha_{t,i}$ representation of x. Figure taken from Bahdanau et al. [2015] 9

2.3 The transformer Architecture. Each block of the transformer architecture consists of a multi-head attention layer followed by feedforward layer. Figure taken from Vaswani et al. [2017] 11

2.4 The self-attention mechanism. Figure taken from Vaswani et al. [2017] . . 12

2.5 Attention masks used by the Transformer architecture. Causal (left), used for text generation tasks, enforces that tokens can not attend to future tokens. Bi-directional (right), used for encoder-only tasks, allows full bidirectional all-to-all attention, including future tokens. 13

2.6 Perceiver Architecture proposed by Jaegle et al. [2021]. The input sequence, consisting of M tokens with dimensionality C are compressed into a smaller set of N latent embeddings with dimensionality D through iterative cross-attention. Figure taken from Jaegle et al. [2021]. 14

3.1 Example of a Reddit Discussion 23
3.2 Graph Transformer Layer proposed by Ying et al. [2021] 27
3.3 Photo of the Drag Queen discussed in Table 3.8 40
3.4 Examples of multi-modal hate speech, where hatefulness can only be discerned by jointly considering both modalities (taken from Kiela et al. [2020]) 45

4.1 General architecture for Discussion Transformers. Z is the number of sequential Graph Transformer and Multi-Modal Fusion Blocks. 51
4.2 Multi-Modal Discussion Transformer 53
4.3 Example Discussion Structure. Each node in the discussion tree represents a comment. The shortest distance between (a, c) and (b, d) is equivalent, which demonstrates a lack of expressiveness for hierarchy. 55
4.4 Fine-grained distribution of BERT and mDT misclassification. (Acronyms above as in Table 4.2) 64
4.5 An image present in the discussion context of example 3 (Table 4.9), seen only by mDT, contextualizing comments as potentially hateful 66
4.6 An image present in the discussion context of the conversation in table 4.9, seen only by mDT, contextualizing comments as potentially hateful 67

6.1 Federated Transformer Q-Function (FedFormer). The input to FedFormer is the observation-action pair (o_i, a_i) for a given timestep i, which is passed to all encoders. Following the initial encoding, positional encoding for each agent (Agent ID Embedding) is added, and the embeddings are then aggregated by a Transformer model. The output is the predicted Q-value $Q(o_i, a_i)$. 80
6.2 Performance on Meta-World tasks 85
6.3 Performance when scaling federation methods to 5, 10 and 15 agents . . . 87
6.4 Performance of onboarding agents into pre-trained federation networks on the `window-close` task 88
6.5 Scaling Performance when replacing Transformer Network with MLP Network 90

7.1 Architecture of Recformer. Recformer represents items as text-based key-value pairs (i.e. Brand: Apple, Colour: Gold). Figure taken from [Li et al., 2023] 100

7.2 Text-to-text recommendation task of P5. Figure taken from Geng et al. [2022] 101

8.1 Sequential recommender model architecture with text encoder augmentation. Text tokens are encoded with a frozen (blue) language model (mT5), and the associated embeddings are resampled with the Perceiver. Components in green boxes are trained using masked language modelling (MLM). 107

8.2 Flare adapted to incorporate recommendation critiques. Critiques are textual descriptions incorporated as a separate modality into the Perceiver to guide the model's predictions. 110

8.3 Annotated example of a product in the Amazon Product Reviews dataset. Note the category string of "Office Products - Office & School Supplies - Notebooks & Writing Pads - Self-Stick Notes". 113

List of Tables

3.1 Community Sample Size for the RedditHate Dataset 26

3.2 Label Distribution 26

3.3 Model Performance (L2 Loss) of Graphormer compared to GAT on each label, with All representing performance on the whole dataset 29

3.4 Model L2Loss Performance on Different Communities 30

3.5 Model Performance when using different labelling weights in equation 3.7, where all weights w_c, w_r, w_i are set to 1 unless stated. 32

3.6 Conversation on r/gay containing contextual speech leading to false positive predictions 36

3.7 Conversation on /r/MensRights containing contextual affiliation-directed hate speech 37

3.8 Conversation on r/rupaulsdragrace containing reclaimed language and multi-modal context 39

3.9 Conversation from r/conservatives containing inciteful speech regarding Black Lives Matter 42

3.10 Conversation from r/politics requiring long-range forecasting from community cues 43

4.1 Definitions of Hate Speech Used 58

4.2 Label Distribution of Hateful Discussions 59

4.3 mDT Model Hyperparameters. The search space for each parameter is denoted by the values in [...] 60

4.4 Performance of mDT against Text-Only Methods 61

4.5 Effect of Bottleneck Size on mDT Performance 61
4.6 Effect of Constraining Graph Attention 62
4.7 Effect of Fusion Layers 62
4.8 Effect of Excluding Images 63
4.9 Text instances misclassified by BERT and mDT. Note: The ground truth for all the examples shown here is "Hateful". We have also redacted portions of the text for brevity, indicated by [...]. 65

6.1 Hyperparameters used by each method explored in our study. Shared parameters apply to all methods. 84

8.1 Statistics for the Recformer- and CALRec-processed Amazon Product Reviews dataset. I/U is the average number of items per user, or average sequence length. Note that the Clothing dataset preprocessing follows the same steps as CALRec 112
8.2 Hyperparameters used by Bert4Rec for each Recformer dataset 115
8.3 Hyperparameters used by Bert4Rec for each CALRec dataset 116
8.4 Hyperparameters used by Flare for each Recformer dataset 116
8.5 Hyperparameters used by Flare for each CALRec dataset 117
8.6 Flare performance against prior baselines on the RecFormer-processed [Li et al., 2023] datasets. Baseline results are quoted from those papers. Bert4Rec* is our tuned Bert4Rec, which outperforms previously reported Bert4Rec baselines. The %ID column is the difference between Flare and Bert4Rec*. The best result is **bolded**; the second best is underlined. 119
8.7 Flare performance against prior baselines on the CALRec-processed [Li et al., 2024] datasets. Bert4Rec* is our tuned Bert4Rec, which outperforms previously reported Bert4Rec baselines. The %ID column is the difference between Flare and Bert4Rec*. The best result is **bolded**; the second best is underlined. 120
8.8 Performance of Flare and Bert4Rec* (tuned Bert4Rec) scaling on "Clothing", a subset with over 370k items. Flare achieves improved performance when increasing the model size. The % columns show the performance improvement of the best model over Recformer. 122

8.9 Flare's performance when utilizing precise (4 categories) and broad (2 categories) textual critiquing. Flare achieves stronger performance under both forms of critiquing, demonstrating that it can leverage textual critiques, though it shows some room for improvement in broad critiquing. 124

8.10 Performance of Flare's category alignment when mutating the critiquing category string to another valid category on Office. Flare's predictions are nearly entirely within the desired category, even when the target category differs significantly from the user's prior actions. 126

8.11 Example user trajectory in the "Office Products" category. The 5th (*) item, the box of T50 staples, is held out and is the target next item. Note the preference for black. 127

8.12 Ranked recommendations generated for the user in Table 8.11 using Bert4Rec* and Flare. Both approaches fail to retrieve the correct item (T50 Staplers). 128

8.13 Top 5 recommendations generated for the user in Table 8.11, using the category of the correct item as a critique instruction. The correct item is retrieved at rank 1, with all other items retrieved within the critique category. Titles shortened are denoted with [...] 129

8.14 Top 5 recommendations generated for the user in table 8.11, using a product category from the user history (in-history critique) and using one outside the user's history (outside-history critique). Both commands are followed, and all retrieved items are within the target category, demonstrating Flare's robust ability to follow instructions. Titles shortened are denoted with [...]. 130

8.15 Top 5 recommendations generated for the user in table 8.11, comparing using broadened categories and more precise categories as critiquing commands. Note that the precise critique was not seen during training, but Flare leveraged the more specific information to retrieve relevant items. . . 131

8.16 Flare ablation study using the Clothing dataset. 132

Chapter 1

Introduction

The ability to make sound judgments is fundamentally tied to the understanding of context. Isolated pieces of information, devoid of their surrounding circumstances, are often ambiguous and can lead to flawed conclusions. Consider the single word "fire." In a crowded theatre, it is a life-saving alarm. On a firing range, it is a command. In a boardroom, it is a termination notice. The word itself is merely data; its context—the environment, the speaker's intent, and the audience—is what transforms it into meaningful information upon which to act. The importance of considering context presents both a foundational challenge and a significant opportunity for the field of artificial intelligence, where systems are increasingly tasked with making critical decisions where context is abundant. It is this challenge that drives this thesis.

The advent of the Transformer architecture marked a significant development in Natural Language Processing, particularly regarding the representation of context. Unlike Recurrent Neural Networks (RNNs) or Naive Bayes classifiers, which are limited by sequential processing constraints or independence assumptions, Transformers utilize self-attention mechanisms to model global dependencies between inputs regardless of their distance in the sequence [Vaswani et al., 2017]. This architecture has demonstrated superior performance in capturing complex contextual nuances within text. Consequently, the utility of the Transformer has extended beyond NLP, becoming a prevailing architecture for tasks requiring contextual modelling in fields such as computer vision [Kim et al., 2021, Tschannen et al., 2025, Alayrac et al., 2022], recommender systems [Sun et al., 2019, Kang and McAuley, 2018], and multi-agent systems [Iqbal and Sha, 2019].

Despite this progress, we find that these applications often focus on a "local" or "micro" context: the relationship between words in a single document or pixels in an image. For

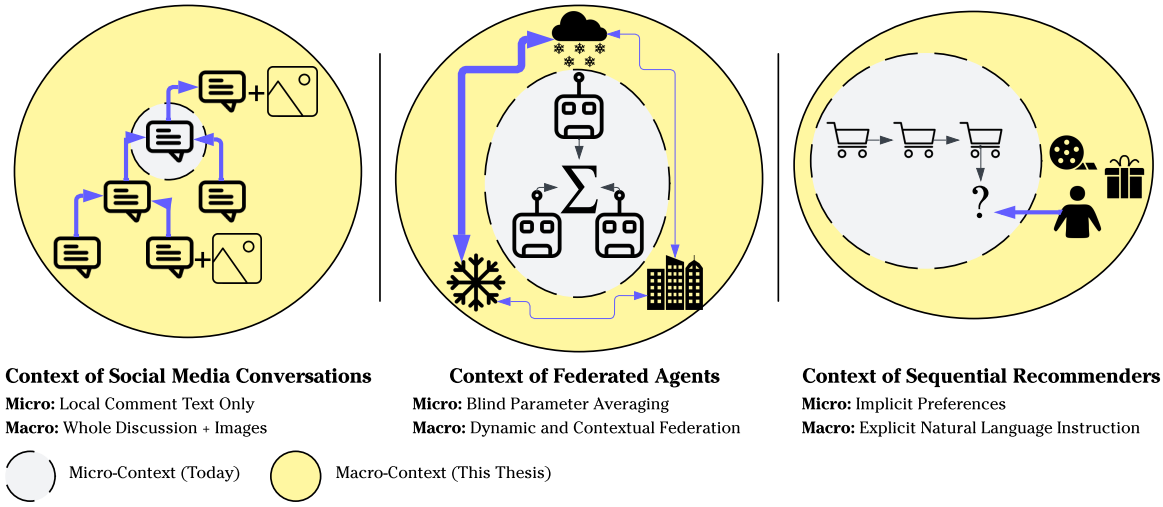


Figure 1.1: Across three domains, this thesis shifts the unit of analysis from a local, micro-context lens (inner dashed circles) to a holistic, macro-context lens (outer solid circles).

many complex, real-world problems, a wealth of additional context exists at a "macro" level, encompassing broader environmental, social, or interactive signals. We argue that systems failing to leverage this macro context risk making uninformed decisions that misrepresent their inputs and ultimately limit their effectiveness. As such, this thesis explores the following question: "*How can we effectively ground Transformer architectures to make decisions using broad macro-level context?*".

To explore this question, we analyze three domains in which understanding macro-context is pivotal yet underutilized, and we propose novel Transformer-based architectures explicitly designed to integrate richer forms of context, thereby disambiguating inputs for more nuanced and robust AI systems. Across all three domains, our contributions center on adapting the Transformer's attention mechanism to model macro-level context. By repurposing the model's ability to weigh relevance, we enable the dynamic integration of broader signals, from conversational structures and agent networks to explicit user intent, directly into the decision-making process.

1.1 Guide to the Thesis and Research Contributions

This thesis is organized into four parts. The first three parts of this thesis each explore a different domain where macro-context is available (Figure 1.1). Our central thesis in each of these parts can be formulated as identifying available yet underutilized contexts, developing novel transformer-based architectures to incorporate them, and proposing methodological contributions to assess their usefulness. As our pursuit of macro-context-aware models spans many domains, each part begins with a domain-specific background chapter rather than a single, encompassing one. Chapter 2 contains the common background for all parts of this thesis, introducing transformers and attention.

Part I: Context of Social Media Conversations In Part I, we explore the domain of social media analysis, with a specific focus on hate speech detection. Existing methods typically restrict their analysis to the text or image of a single, isolated comment. We argue, however, that the broader macro-context context of the discussion is essential to disambiguate the a comment's intended meaning. For example, a phrase like "That's gross!" can be benign or hateful depending entirely on the thread it belongs to. Capturing this context presents a significant technical challenge, as standard text Transformers cannot naturally model the branching, tree-like structure of a discussion. Furthermore, standard Graph Neural Networks often struggle to capture long-range dependencies, such as those required to model semantic dependencies within a discussion.

To address these challenges, we propose the Multi-Modal Discussion Transformer (mDT). Our solution is built on the intuition that conversations share structural properties with molecules—just as atoms form molecules through bonds, comments form discussions through replies. Using this formulation, we adapt Graph Transformers, originally designed for molecular modelling [Ying et al., 2021], to model interactions among comments within a discussion tree. mDT further innovates by interweaving text and image layers directly into graph layers, enabling active grounding of micro-comment semantics in macro-discussion semantics. To validate this paradigm, we introduce HatefulDiscussions, a novel dataset of complete multimodal discussion trees that contain all requisite text, image, and graph modalities. Our experimental results demonstrate a 7% improvement in F1, measuring overall classification performance, over existing text-only baselines, underscoring the critical value of incorporating structural macro-context. Notably, we also demonstrate a 21% improvement in F1 over previous graph-based methods, further validating our architecture's incorporation of graph-based context.

Part II: Context of Federated Agents In Part II, we turn our attention to multi-agent federated reinforcement learning. Here, the "micro-context" is an individual agent's local information (e.g., the current environment of an agent), while the "macro-context" is the collective knowledge of a fleet of agents operating in diverse, heterogeneous environments (e.g., sunny vs. rainy conditions). The primary challenge in this domain is that the prevalent federated learning method, Federated Averaging (FedAvg) [McMahan et al., 2017], aggregates agents' weights via averaging, resulting in a single one-size-fits-all model for all agents. In heterogeneous environments, this approach may result in a model failing to specialize, as knowledge from one agent in one environment may introduce noise and degrade performance in a conflicting environment. For instance, knowledge gained in one context (e.g., a self-driving car on a rural road) may be irrelevant or even detrimental in another (e.g., a dense urban environment).

To address this, we propose FedFormer, a Transformer-based federation strategy that treats the federation process as a contextual routing problem. By leveraging the attention mechanism in Transformers, FedFormer enables an agent (the Query) to dynamically weigh the contributions of other agents (the Keys and Values) based on their relevance to its current environment. This enables agents to selectively learn from relevant peers while ignoring conflicting information, based on macro-level information. Notably, FedFormer achieves a more effective and efficient federation while still preserving privacy. We evaluate FedFormer on the Meta-World robotic benchmark, a set of heterogeneous robotic manipulation tasks, and find that FedFormer significantly outperforms FedAvg, achieving between 1.48x to 3.41x faster convergence. FedFormer's architecture also uniquely enables the ability to onboard new agents into existing federation networks, reducing the number of steps required for convergence relative to standard methods by up to 11x compared to FedAvg and 6.5x over non-federated methods (Soft Actor Critique [Haarnoja et al., 2018]).

Part III: Context of Sequential Recommender Systems In Part III, we examine how to incorporate user intent into hybrid sequential recommenders. Sequential recommender systems utilize the temporal context of previous interactions (i.e. purchases or clicks) to recommend items to users. This process is often implemented through collaborative filtering, which leverages community homophily by recommending items based on learned behavioural similarities among users. Hybrid sequential recommenders extend this idea by incorporating content features, such as product descriptions and other attributes, into the recommendation process, bolstered by the strong world knowledge of Large Language Models (LLMs). However, this emphasis on community homophily often comes at the expense of user agency, resulting in rigid recommendations that fail to capture a user's current intent.

To this end, we propose Flare, a hybrid sequential recommender that efficiently integrates collaborative filtering, content features, and natural-language user intent (Chapter 8). Our architecture utilizes a Perceiver Transformer network to address the integration challenge. The Perceiver employs a small set of latent vectors to "bottleneck" and efficiently fuse multimodal inputs, combining CF embeddings, text descriptions, and user queries, into a compact learnable representation. Notably, this enables our model to adapt to explicit natural-language queries while leveraging historical patterns. To evaluate Flare's ability to capture intent, we propose a novel evaluation paradigm and metrics that directly measure the overlap between intent and retrieved items. Our results using a dataset of Amazon product reviews [Ni et al., 2019] show an improvement of 2.53x in recall@10, measuring retrieval capabilities, compared to approaches that do not factor in user intent.

Part IV: Putting Everything into Context In the final part of this thesis, Part IV, we discuss the common threads that unite our contributions in this thesis and the common directions for future work. Here, we also discuss how the empirically validated ideas in this thesis can have a broader social impact, intersecting with themes in AI for Social Good and Responsible AI. We believe that these directions are especially crucial today as AI systems assume increasingly consequential roles.

Across these three diverse applications—hate speech detection, federated learning, and recommender systems—a familiar narrative emerges. Models often operate on isolated, local information while ignoring the richer macro-level context in which it is embedded. By developing novel architectures that explicitly model conversational structures, environmental heterogeneity, and direct user intent, this thesis demonstrates a path toward more intelligent, adaptable, and grounded AI systems.

Chapter 2

Background

Understanding the context of information is especially important in Natural Language Processing (NLP), a domain of artificial intelligence that aims to understand and generate language. We begin this thesis by first introducing prior work that laid the foundation for the attention mechanism and the transformer architecture, the primary focus of this thesis. We start by introducing recurrent neural networks in Section 2.1, then additive attention in 2.2 and finally transformers in Section 2.3. Within section 2.3, we also discuss encoder-only transformers in Section 2.3.1, the variant we use in this thesis, and transformer pre-training in Section 2.3.2.

2.1 Recurrent Neural Networks

Much of the work in this thesis traces its origins to the Recurrent Neural Networks (RNNs). RNNs are networks that attempt to capture temporal dependencies within a sequence of items, such as words in a sentence [Elman, 1990, Jordan, 1986, Lang et al., 1990, Jaeger, 2001]. This process differs significantly from standard feed-forward neural networks, which can only process fixed-size inputs. RNNs achieve this by sequentially updating an internal hidden state vector (h_t) that acts as a memory of the sequence processed so far.

At a high level, RNNs maintain a hidden state h_t at each time step t, from which a corresponding output y_t is derived. The hidden state is constructed by learning a function combining the current input x_t with the hidden state from the previous step, h_{t-1}. The model's output y_t at that timestep is then computed as a learned projection of h_t. This

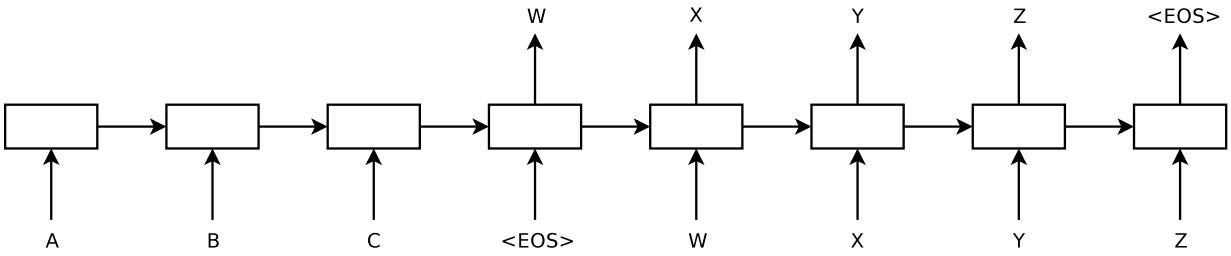


Figure 2.1: The seq2seq architecture. Here, the RNN model first reads the input sequence 'ABC', then produces 'WXYZ'. The model switches to decoding when receiving the 'end-of-sequence' token. Figure taken from Sutskever et al. [2014]

recurrent process can be formally expressed as:

$$h_t = \sigma_h(\mathbf{W^x} x_t + \mathbf{W^h} h_{t-1}) \tag{2.1}$$

$$y_t = \mathbf{W^y} h_t \tag{2.2}$$

where $\mathbf{W^x}$, $\mathbf{W^h}$ and $\mathbf{W^y}$ are learned weight matrices, and σ_h is a non-linear activation functions (typically tanh). This mechanism allows the RNN to gradually build a contextual representation of the sequence, with the final hidden state h_t theoretically encoding a summary of the entire input. Follow-up work includes Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Units (GRU) networks [Cho et al., 2014], with various improvements, such as forget and context gates, to enhance memory.

RNNs have also demonstrated strong performance on text generation tasks, such as the seminal seq2seq method by Sutskever et al. [2014] shown in figure 2.1. In seq2seq, an encoder LSTM processes the source sentence ('ABC') to create a final hidden state vector h_x, ignoring intermediary y_t predictions, summarizing the input sequence x. Then, h_x is used to prompt a decoder LSTM to auto-regressively generate an output sentence ('WXYZ'), where the predictions y_t are fed back into the model as x_{t+1}.

2.2 Attention

The reliance on fixed-size hidden state vectors h_t led to two fundamental challenges. The first was the information bottleneck: By forcing RNNs to compress the meaning of a potentially long, complex sentence into a single fixed-size vector, RNNs would lose information

as more tokens are processed. This, in turn, would cause RNNs to "forget" crucial information from the beginning of the sequence [Pouget-Abadie et al., 2014]. The second issue is the vanishing gradient problem [Hochreiter and Schmidhuber, 1997]. As the network processed a sequence, the gradients of earlier tokens on the final output would diminish with each successive update. Consequently, RNNs struggle to learn how to capture long-range dependencies.

To address the information bottleneck, the attention mechanism was introduced by Bahdanau et al. [2015], inspired by theories of human attention. The core insight was that instead of forcing the decoder to rely on a single summary of the input, it should be allowed to "look back" at the entire input sequence at every step of the generation process and learn to focus on the most relevant parts. This process is performed by creating a contextual representation of the input c_t, weighted by attention weights $\alpha_{t,i}$, which are calculated based on the importance of the input word at position i at timestep t. For example, consider the sentence "Liam wrote an incredible thesis". When processing the word "thesis", the word "Liam" would likely receive greater attention due to its contextual relevance, while "an" would receive less. The attention mechanism can be seen as analogous to database retrieval: the model sends a *query* and gets a reply that is a weighted sum of *values*, proportional to how closely the query resembles its corresponding *key*.

In practice, Bahdanau et al. [2015] implements the additive attention mechanism (figure 2.2) by extending RNNs to create an attention context vector c_t and state vector s_t as:

$$y_t = g(y_{t-1}, s_t, c_t) \tag{2.3}$$
$$s_t = f(y_{t-1}, s_{t-1}, c_t) \tag{2.4}$$

where f and g are learned RNN layers. The context vector c_t is constructed as follows: First, the input x is encoded using an encoder bidirectional LSTM, to create an initial set of representations $h_t \in H$ for each token in $x_t \in X$ (as in equation 2.2). Then, an alignment score $e_{t,i}$ is computed between the previous state s_{t-1} (the query) and each of the H representations (the keys). This is done using a single-layer feed-forward network as follows:

$$e_{t,i} = \mathbf{v^T} tanh(\mathbf{W^Q} s_{t-1} + \mathbf{W^K} h_i) \tag{2.5}$$

where $\mathbf{W^Q}$, $\mathbf{W^K}$ and $\mathbf{v}$ are learnable weight matrices trained with the rest of the model. It is the addition $\mathbf{W^Q} s_{t-1} + \mathbf{W^K} h_i$ that gives additive attention its namesake.

Following the computation of the alignment scores $e_{t,i}$, each score is normalized using a softmax function, ensuring that all scores sum to 1:

$$\alpha_{t,i} = softmax(e_{t,i}) = \frac{exp(e_{t,i})}{\sum_{k=1}^{|H|} exp(e_{t,k})} \tag{2.6}$$

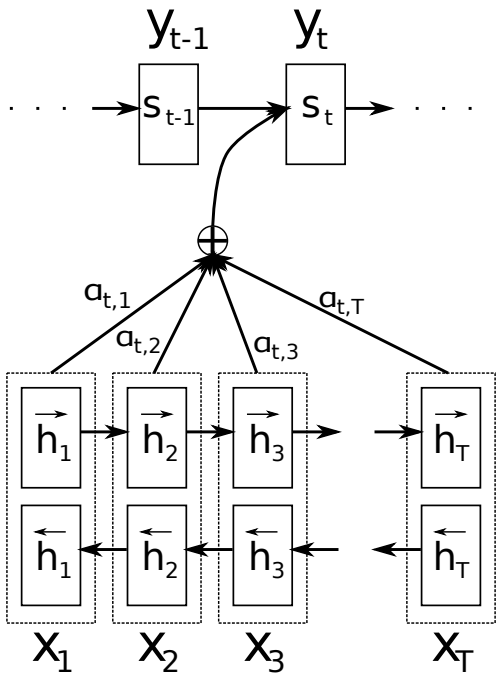


Figure 2.2: RNN additive attention. The inputs are first encoded using a bidirectional RNN to create h_i. Then, the state s_t is created as a product of s_{t-1} and an attention weighted $\alpha_{t,i}$ representation of x. Figure taken from Bahdanau et al. [2015]

where $|H|$ is the cardinality of encoder representations. Then, the context vector (c_t) is created as the attention $\alpha_{t,i}$ weighted sum of all encoder representations

$$c_t = \sum_{i=1}^{|H|} \alpha_{t,i} h_i \tag{2.7}$$

where the resulting context vector c_t is a weighted summary of the input sequence as it relates to the current state s_t. In essence, each attention weight $\alpha_{t,i}$ represents the amount of "attention" the model pays to each encoded token.

2.3 Transformers

Encouraged by the performance gains enabled by attention, Vaswani et al. [2017] proposed removing the sequential nature of RNNs entirely by replacing all model layers with attention operations. The result is the Transformer architecture (figure 2.3), the primary focus of this thesis. The key innovation behind the Transformer is the self-attention mechanism. The core idea behind self-attention is to replace the recurrent context vectors s_t and c_t by applying the attention mechanism directly to all tokens $x \in X$ in the input.

Extending equation 2.5, self-attention (figure 2.4) is computed as

$$SelfAttention\left(X\mathbf{W^Q}, X\mathbf{W^K}, X\mathbf{W^V}\right) = Softmax\left(\frac{X\mathbf{W^Q} \cdot (X\mathbf{W^V})^T}{\sqrt{d}}\right) X\mathbf{W^V} \tag{2.8}$$

where $\mathbf{W^Q}$, $\mathbf{W^K}$, $\mathbf{W^V}$ are learned projection matrices that create query, key, and value representations, respectively. The alignment score computed by $X\mathbf{W^Q} \cdot (X\mathbf{W^V})^T$ is divided by the square root of the embedding dimension of X, $\sqrt{d}$, to reduce the variance of the resulting sum.

Note the difference in the attention softmax operation compared to additive attention, which now computes in parallel an $\mathcal{R}^{|X|\times|X|}$ attention matrix representing all-to-all attention between all tokens in the input at each layer. The result is a vastly more expressive and efficient attention operation, enabling higher-order relationships between tokens and richer contextual representations.

A causal attention mask is typically applied during text generation, where the attention weights for future tokens are set to $-\infty$ to prevent earlier tokens from attending to future ones (left side of figure 2.5). Additionally, Transformer models typically consist of multiple self-attention blocks stacked, each operating on a subset of the embedding dimension d

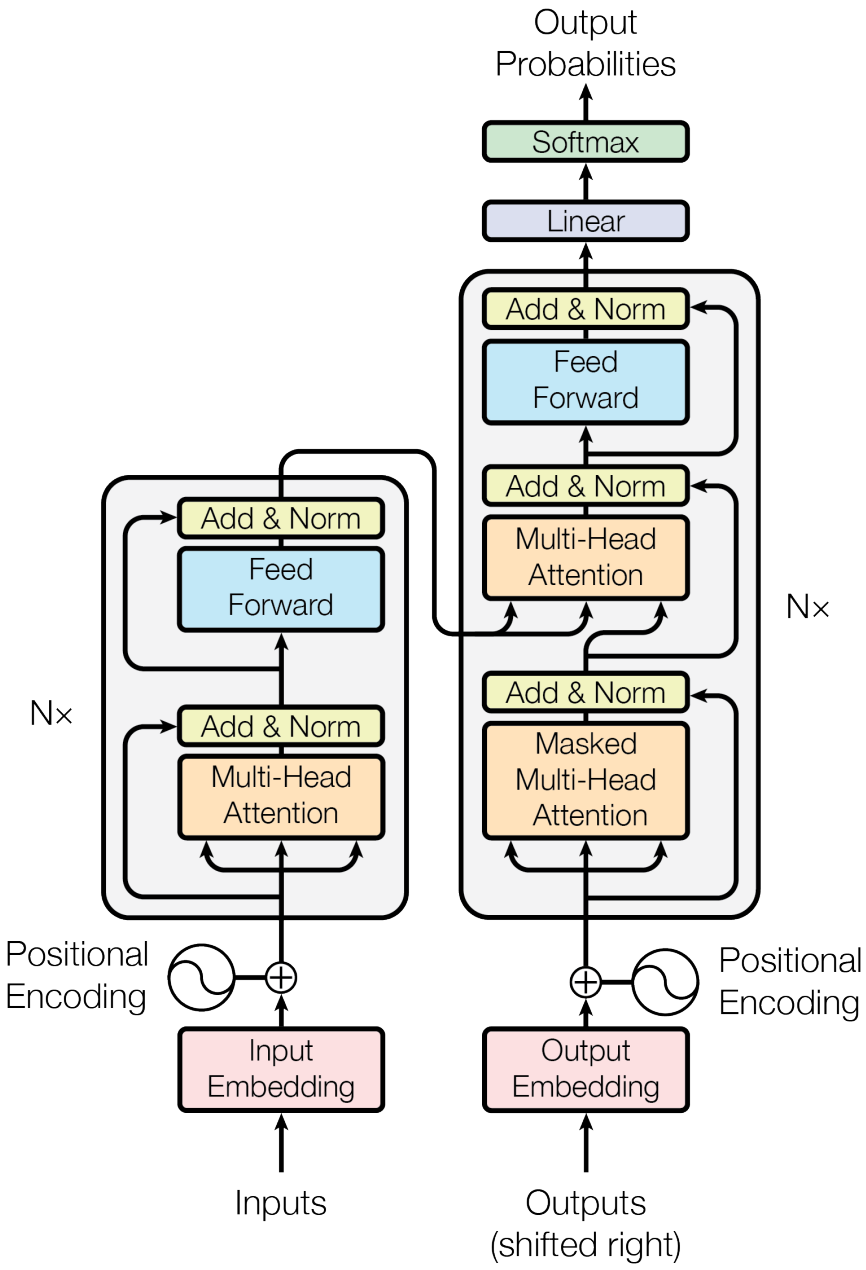


Figure 2.3: The transformer Architecture. Each block of the transformer architecture consists of a multi-head attention layer followed by feedforward layer. Figure taken from Vaswani et al. [2017]

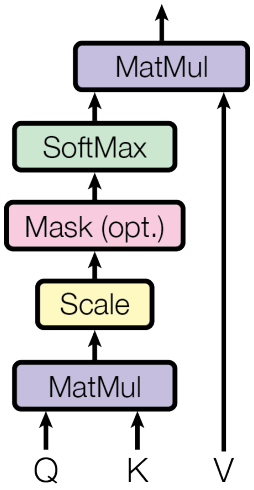


Figure 2.4: The self-attention mechanism. Figure taken from Vaswani et al. [2017]

(referred to as an attention head), to form the multi-head attention block [Vaswani et al., 2017]. This improves efficiency and enables reasoning on different embedding subspaces.

Putting it all together, a transformer consists of sequential layers of self-attention followed by a feed-forward layer FFN, given as

$$X^{l+1} = X^l + FFN(MultiHeadAttention(X^l\mathbf{W}^Q, X^l\mathbf{W}^K, X^l\mathbf{W}^V))$$

where X^{l+1} is fed into the next transformer layer [1]. Note that, without sequential recursion, Transformer layers cannot directly differentiate the order of their inputs. To address this, Transformer models encode position by inserting positional information into X. This is typically done using either learned additive positional encodings at the input token level [Vaswani et al., 2017, Devlin et al., 2019], or, more recently, using position-wise vector rotations during each self-attention operation [Su et al., 2024].

2.3.1 Encoder-Only Pre-Trained Transformers

While the original Transformer proposed by Vaswani et al. [2017] used an encoder-decoder architecture for sequence-to-sequence tasks (such as translation), subsequent work found that the encoder stack alone yields powerful representations for language understanding.

[1] For the sake of brevity, we omit the layer normalization layers which occur after multi-head attention and the feed forward layer.

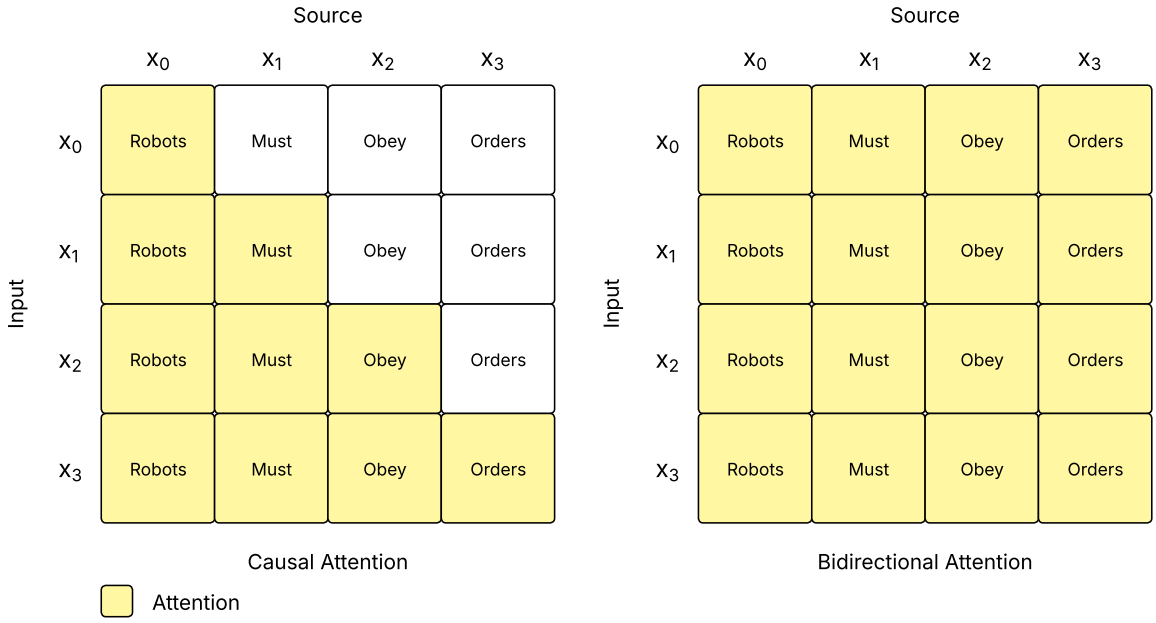


Figure 2.5: Attention masks used by the Transformer architecture. Causal (left), used for text generation tasks, enforces that tokens can not attend to future tokens. Bi-directional (right), used for encoder-only tasks, allows full bidirectional all-to-all attention, including future tokens.

The most influential model in this class is BERT (Bidirectional Encoder Representations from Transformers), proposed by Devlin et al. [2019].

The encoder-only architecture consists of a stack of Transformer layers (as described in section 2.3) that process the input sequence in parallel. Unlike autoregressive models (such as GPT [Brown et al., 2020]) or the original Transformer model, which must use a causal mask to limit attention to future tokens, the encoder-only Transformer uses bidirectional self-attention (right side of figure 2.5). This means that every token in the sequence can attend to every other token, including future tokens, within each layer.

A key contribution of the BERT encoder-only architecture is the introduction of the "[CLS]" (short for classification) token, appended at the start of the input sequence. Because the self-attention mechanism allows every token to attend to all others, the hidden state of the "[CLS]" token can aggregate information from the entire input sequence. The final representation of this token serves as a fixed-length, holistic representation of the full

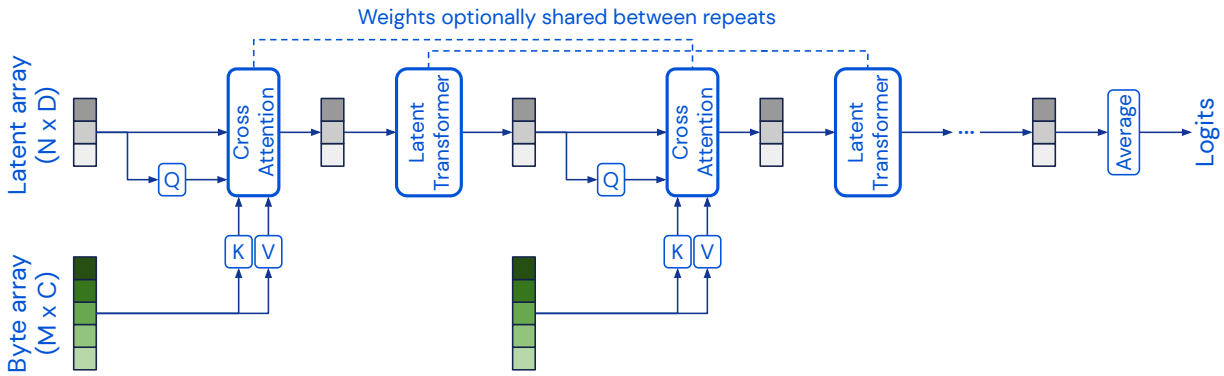


Figure 2.6: Perceiver Architecture proposed by Jaegle et al. [2021]. The input sequence, consisting of M tokens with dimensionality C are compressed into a smaller set of N latent embeddings with dimensionality D through iterative cross-attention. Figure taken from Jaegle et al. [2021].

input text, which can be fed into a simple classification layer (e.g., a linear projection layer followed by softmax) for tasks such as sentiment analysis or hate speech classification. While the [CLS] token is the standard mechanism for sequence aggregation, recent research has explored alternative methods to derive fixed-length global representations. For instance, Sentence-BERT [Reimers and Gurevych, 2019] demonstrated that applying mean pooling to the output token representations often yields superior semantic embeddings.

Of particular relevance to this thesis is the Perceiver architecture [Jaegle et al., 2021], which is used in Chapter 8 (Figure 2.6). Unlike standard Transformers that maintain a 1:1 ratio between input and output tokens, the Perceiver is designed to distill high-dimensional inputs into a concise, fixed-size latent representation. This is achieved through an asymmetric cross-attention mechanism. In this setup, the Query (Q) matrices are projected from a small set of fixed, learnable latent vectors, while the Key (K) and Value (V) matrices are projected from the original input sequence. The attention mechanism is then refined during training to select and compress only the most salient information from the variable-length input into the compact latent vectors, creating an information bottleneck. This decoupling of the latent space from the input size allows the Perceiver to efficiently process and fuse very large or multi-modal contexts.

2.3.2 Pre-training Transformer Models

Pre-training has been extensively explored in natural language processing (NLP) to address data scarcity [Radford et al., 2018, Devlin et al., 2019]. The core premise behind unsupervised pre-training is that leveraging linguistic information from unlabeled data can be a valuable alternative to further manual annotation [Radford et al., 2018]. For instance, training a model first to understand English syntax and grammar is a transferable skill to determining whether an English comment contains hate speech. Building on the concept of transferable skills, pre-training proposes simple tasks that operate on virtually unbounded amounts of data to learn foundational skills relevant to downstream tasks. This, in turn, enables Transformer models, such as BERT [Devlin et al., 2019], to acquire substantial world knowledge applicable across many domains, including those examined in this thesis.

Pre-training tasks in NLP can be categorized into two groups: generative and contrastive [Hu et al., 2020a]. Generative pre-training tasks involve reconstructing masked or permuted input sequences to their original form, with a focus on refining token-level representations. A seminal example of generative pre-training is the masked language modelling (MLM) task used by bidirectional transformers, proposed by Radford et al. [2018]. In MLM, tokens in a sequence are randomly replaced with "[MASK]" tokens (i.e. "[CLS] Liam wrote a [MASK] thesis!") and the model is tasked to predict what the missing word was (i.e. "incredible"). Notably, this task can be applied to arbitrary text and exercises the model's ability to learn language semantics, a valuable capability for downstream tasks [Radford et al., 2018]. In practice, generative pre-training is typically performed either as a classification task, in which the model selects the original input from a set of options (e.g., tokens) and optimizes a cross-entropy loss [Radford et al., 2018], or directly using embedding similarity with an L2 loss [Hu et al., 2020a].

While generative tasks aim to refine models at the local token level, contrastive tasks aim to refine models' ability to capture global information (i.e., at the sentence or document level). In essence, contrastive tasks align global representations of examples so that semantically similar examples (positives) are pulled closer together and dissimilar examples (negatives) are pushed apart in a shared embedding space [Oord et al., 2018]. A prevalent example of contrastive pre-training is the next-sentence prediction task, where a model is given two sentences and must predict whether they follow each other (i.e., are correlated), using random sentences as negatives [2]. A similar task can be formulated in document retrieval, where a model is tasked with aligning document representations with corresponding

[2]We note that this task can also be formulated as a classification task, where both input sequences are concatenated into one model.

search query representations, where random documents are sampled as negatives [Nogueira et al., 2020].

In practice, contrastive pre-training is often formulated using the InfoNCE loss [Oord et al., 2018]. Given two embedding sets $a \in A$ and $b \in B$ where items a_i, b_i are positive pairs, InfoNCE loss consists of optimizing

$$\frac{1}{2\mathcal{B}} \sum_{i}^{\mathcal{B}} \left(\overbrace{log \frac{exp((\mathbf{a}_i \cdot \mathbf{b}_i)/\tau)}{\sum_j^{\mathcal{B}} exp((\mathbf{a}_i \cdot \mathbf{b}_j)/\tau)}}^{a \to b \text{ softmax}} + \overbrace{log \frac{exp((\mathbf{a}_i \cdot \mathbf{b}_i)/\tau)}{\sum_j^{\mathcal{B}} exp((\mathbf{a}_j \cdot \mathbf{b}_i)/\tau)}}^{b \to a \text{ softmax}} \right) \tag{2.9}$$

where $\mathbf{a}$ and $\mathbf{b}$ consist of L2 normalized representations of a and b, respectively, τ is a temperature scaling parameter used to sharpen the similarity distribution. $\mathcal{B}$ denotes the sample size of A and B used in the current model iteration, referred to as the batch size. Note that the denominator is based on similarity to other embeddings in the batch, referred to as in-batch negatives. We discuss the potential of contrastive pre-training in Part III and Part IV of this thesis.

Part I

Context of Social Media Conversations

Chapter 3

Background and Roadmap

3.1 Roadmap

In part I of this thesis, our focus is on understanding the complex, multi-faceted nature of social media discussions. In particular, we are interested in proposing methods to disambiguate the meaning of online comments towards hate speech detection, using the broader discussion as a vital source of context.

We start by defining the relevant background in this chapter. Section 3.2 details related work in multi-modal models, models that jointly process both text and images, and their applications towards hate speech detection. Next, section 3.3 details related work on Graph Neural Networks (GNNs) and techniques for processing graph-structured data, such as molecules. In our work, we adapt GNNs to instead model the connections between comments. Towards this, section 3.4 presents our preliminary study of applying Graph Transformers, a Transformer variant of Graph Neural Networks, to model conversational context (section 3.4, complemented by a qualitative study in section 3.5). These studies motivate many design decisions made in chapter 4, such as the power of Graph Transformers to model context, and the need to consider the impact that images can have on meaning. Section 3.6 establishes related background in multi-modal hate speech detection.

In Chapter 4, we introduce the Multi-Modal Discussion Transformer (mDT). Our architecture combines Graph Transformers with multi-modal image and text models to holistically encode conversations. In contrast to our study in section 3.4, which sequentially encodes comments and then aggregates them using a Graph Transformer, mDT interweaves discussion context with comment-modality layers. This enables comments to be

actively perceived in the context of the discussion, rather than only considered post-hoc. To evaluate our unique paradigm of multi-modal discussion-aware hate speech detection, we propose HatefulDiscussions, a dataset that includes complete multi-modal discussion trees.

Within that Chapter, section 4.1.1 outlines the mDT architecture and section 4.1.3 outlines the HatefulDiscussions Dataset. We then describe our results in section 4.2, comparing mDT's performance against other comment-only and graph-based hate speech detection methods in section 4.2.2. We also conduct various ablation studies, varying our multi-modal mechanism (section 4.2.3), graph attention mechanism (section 4.2.4), graph fusion mechanism (section 4.2.5) and the effect of including images (section 4.2.6). We conclude with a qualitative analysis of mDT's behaviours in section 4.2.7 and a discussion on future directions 4.3. Section 4.4 summarizes our contributions.

3.2 Hate Speech Detection

Transformer-based encoding models, such as BERT, have significantly improved natural language processing by capturing textual semantics [Devlin et al., 2019]. Inspired by these developments, methods such as HateXplain [Mathew et al., 2021] and HateBERT [Caselli et al., 2021] have been introduced to detect hateful comments on social platforms, focusing solely on text. Similar work in Babanejad et al. [2020] has explored applying BERT for affective sarcasm detection.

Progress in automated hate speech detection relies on the availability of high-quality, annotated datasets for evaluation. Several prominent datasets have emerged in this field, each with its own characteristics and focus. HateXplain [Mathew et al., 2021] contains post annotations for three-class hate classification (hate, offensive, or benign), the specific community targeted by the speech and the rationale behind each annotation decision. HateXplain contains posts collected from Twitter and Gab and aims to facilitate research on detecting hate speech and understanding its rationales [Mathew et al., 2021]. Another dataset focusing on granular hate speech detection is ETHOS [Mollas et al., 2022], which includes multi-class labels for posts regarding the type of hate speech present (e.g., violence, directed vs. general, and categories such as gender, race, and religion). Other datasets, such as Davidson et al. [2017], MetaHate [Piot et al., 2024], Gong et al. [2021], Founta et al. [2018], and Zampieri et al. [2019], focus on binary classification but cover a broader range of posts.

Despite this progress, many of these techniques overlook a vital modality: the context of discussions. The prevailing emphasis remains on datasets and techniques that analyze

singular comments, bypassing the contextual significance of the prior discussion. This research addresses this limitation by proposing methods to ground comments in a discussion context.

3.3 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as the dominant paradigm for learning representations from graphs, such as social networks, molecular structures, and knowledge graphs. Formally, given a graph $G = (V, E)$ with a set of vertices V and edges E, the goal of a GNN is to learn a d-dimensional feature vector (embedding) h_v for every node $v \in V$ that captures both its semantic content and its structural role within the graph.

The majority of modern GNNs, such as GCN [Chen et al., 2018], GraphSAGE [Hamilton et al., 2017] and GAT [Veličković et al., 2017], operate under the *Message Passing* framework. This paradigm relies on the graph's local connectivity to propagate information. Each layer of a Message Passing GNN consists of two operations: Aggregate and Combine.

For a specific node v at layer k, the Aggregate operation pools feature information from its local neighborhood $\mathcal{N}(v)$:

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right) \tag{3.1}$$

where $a_v^{(k)}$ is the aggregated message. Common aggregation functions include summation, mean pooling, and max pooling. Subsequently, the Combine operation updates the node's representation by fusing its current state with the aggregated message:

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right) \tag{3.2}$$

Through this iterative process, a GNN with k layers enables a node to capture information from neighbours k hops away. However, this local reliance presents a significant limitation: capturing long-range dependencies requires stacking many layers. This often leads to the "oversmoothing" problem, where node representations converge to indistinguishable values, causing the model to lose track of the distinct structural context required for holistic analysis.

Graph Transformers arose to address this limitation by adapting the self-attention mechanism to graph-structured data [Ying et al., 2021]. Conceptually, Graph Transformers generalize the Message Passing framework by replacing the fixed, local Aggregate operation

with a learnable, global attention mechanism. Instead of aggregating only from immediate neighbours $\mathcal{N}(v)$, a node attends to *all* other nodes in the graph to compute its update.

In this paradigm, the structural information, previously implicit in the neighbourhood aggregation, must be explicitly injected into the model. Since standard self-attention is permutation-invariant (i.e., it treats the graph as a "bag of nodes"), Graph Transformers rely heavily on structural encodings (such as centrality or spatial encoding) to inform the attention mechanism about the graph's topology [Ying et al., 2021]. This shift from local message passing to global, structure-aware attention is the foundational principle behind the architectures proposed in the following section.

3.4 Guiding Study of Graph Transformers for Predictive Hate Speech Detection

This section showcases the first step we took toward developing the Multi-Modal Discussion Transformer, discussed in the following chapter. Here, we examined whether considering discussion context could help predict the propensity of comments to turn into hate speech using a graph transformer called Graphormer. In the following section (section 3.5), we also conduct an extended examination of Graphormer's qualitative behaviours, which motivated many design decisions in mDT. In particular, our findings motivated the need to include multimodal representations of comments.

To achieve proactive detection, we hypothesize that it is vital to consider the unique latent communal norms and behaviours that underpin social groups. Content that may trigger a hateful reaction from members of one group can elicit a different response from members of another group. This focus is inspired by a recent study by Cinelli et al. [2021], which found that online communities are prone to the "echo chamber" effect, where extreme ideals can be encouraged and amplified. As such, learning these cultural norms and behaviours is essential to properly model the direction of a conversation and determine whether it leads to hate speech.

Considering all these observations, the question is how to design a framework that can perform a rich analysis of branching social media discussions with a view to proactive hate speech prediction, and also how best to design experiments to demonstrate whether community-specific analysis provides essential benefits.

The focus of this study is on Reddit, a social platform where discussions take place in diverse topic-oriented communities called subreddits. These communities are highly

varied, ranging from `aww`, to discuss cute animals, and `politics`, to discuss political affairs. Beyond their subject differences, prior work by Waller and Anderson [2021] found that these communities exhibit significant differences in their social makeup and discussion behaviours. For example, discussions on `The_Donald` were found to have a significant right-leaning political bias, whereas `politics` shows an opposing left-leaning bias. Within these communities, conversations take place in branching tree structures where any user can reply to other users' comments, forming a linked chain of replies that can branch into sub-discussions. In addition to textual replies, users can explicitly express their opinion by upvoting (approving) or downvoting (disapproving) other users' comments.

The need for a deeper analysis of discussions, with critical consideration of the relevant context, becomes apparent when examining the Reddit excerpt in Figure 3.1. Focusing on the first interaction is insufficient for detecting a tendency towards hate, as these initial comments are actually relatively benign. Indeed, what is perhaps most troubling about this excerpt is the snowballing effect of anti-social sentiment as the discussion progresses.

In this work, we study a novel approach to platform moderation that addresses these challenges. First, it is important to encode the complex lexicon unique to online discussions properly [Belli et al., 2020]. To do this, we use a fine-tuned BERT language model trained on hateful text from various social media platforms [Mathew et al., 2021, Devlin et al., 2019]. This allows our system to utilize contextual language embeddings of each node in the discussion graph as initial feature representations.

Next, to capture the structure of conversations, we propose adaptations to Graphormer [Ying et al., 2021], a state-of-the-art graph transformer network. Notably, our application of Graphormer is significantly different from its original purpose of detecting global molecular properties. Instead, we use Graphormer to predict whether nodes (comments) in our discussion graph are likely to lead to hateful conversations. By adopting this model, we benefit from a self-attention aggregation over the entire graph, enabling the model to learn contextual relationships among relevant comments in the discussion. Our end-to-end solution uses the language model's embeddings as the representation for each vertex in our discussion graph, which are then aggregated and transformed by our Graphormer model to produce final node-level predictions.

To train such a system, we assemble a dataset of 333,487 discussion graphs from different Reddit communities. Each node in a discussion tree is then algorithmically labelled with an ordinal value (0-4) based on the prevalence of hate speech in the discussion that follows under that node. By utilizing a ranked representation of hate, we recognize that encouraged hateful behaviour can be more damaging than discouraged hateful behaviour, and that not all hateful behaviour is equivalent. We then investigate the impact of modelling specific

[redacted] · 3 yr. ago 2

America

How could Flynn have any information that would implicate the president, if the president were innocent?

2.3k Give Award Share Report Save

[redacted] · 3 yr. ago

Going further, why would Trump himself attempt or instruct others to attempt the 10 obstructive acts detailed in the report if he was innocent?

Why would his associates have had to stonewall, destroy evidence, tamper with witnesses, and lie to investigators if he was innocent?

Why has he claimed executive privilege over the redacted portions of the report if he's innocent and "totally exonerated?"

It still boggles my mind that conservatives don't seem to possess the basic cognitive function required to utilize simple logic.

1.4k Give Award Share Report Save

[redacted] · 3 yr. ago

Answer: because the trump-drunk crowd doesn't care what he's guilty of--he's the mascot for their team.

They. Do. Not. Care.

605 Give Award Share Report Save

[redacted] · 3 yr. ago

11 more replies

[redacted] · 3 yr. ago

Even if we ignore the smoke that suggests fire, obstruction regardless of being guilty or otherwise, is still illegal and should be an impeachment.

If you are obstructing because you don't want a bad hair day photo go get out to the public, illegal. If you are obstructing just to be contrarian, illegal. If you are obstructing because you're worried you'll be mistakenly found guilty, well now you're guilty of obstruction.

53 Give Award Share Report Save

4 more replies

Figure 3.1: Example of a Reddit Discussion

communal influence by measuring the performance of our system against variants trained on a single Reddit community.

We conclude this study with a reflection on how our proactive approach to predicting hate speech can improve users' online mental health. This final discussion will shed more light on the significant value we derive from considering trees of conversations in social media, rather than simply interpreting whether a single comment conveys hate speech.

3.4.1 Methods

Data

To train our model, we collected Reddit posts from 2018 to 2019 using the Pushshift API[1], focusing on the `The_Donald`, `IAmA`, `AmITheAsshole`, and `Politics` communities. This resulted in a dataset of 333,487 posts and 6,531,455 comments. These communities were selected for their frequent and lively discussions on various topics and for known differences in community norms [Parmentier and Cohen, 2019, Waller and Anderson, 2021].

Due to the dataset's size, it is not feasible to label every subtree by hand. Instead, we rely on several proxy features. First, we use the scores (upvotes and downvotes) that users assign to comments on Reddit. We use an aggregated tally of these scores to model the inflammatory nature of a given comment to a community. We also use the BERT Hate-Explain hatefulness prediction of comment text [Mathew et al., 2021] and the graph structure as proxy variables. This approach of using algorithmic labelling builds on prior work on labelling hate speech on social platforms, such as Parmentier and Cohen [2019] and Brassard-Gourdeau and Khoury [2021], among others. In each of these systems, algorithmic methods such as HateXplain substitute human reviewers to label a sufficiently large dataset.

Leveraging these proxy terms, we compute the hate propensity label of a comment L_{c_i} by computing a weighted sum (w_c, w_r and w_i) over three symbolic terms that estimate how hateful the conversation is stemming from that node: Context C_{c_i}, Reaction R_{c_i}, and Influence I_{c_i}. For a given comment c_i with parent comment p_i and replies $s \in S_i$, we

[1] https://github.com/pushshift/api, which Reddit has since taken down

compute each term as follows:

$$W_c = hate_c \times score_c \tag{3.3}$$
$$C_c = W_p \tag{3.4}$$
$$R_c = W_c \tag{3.5}$$
$$I_c = \frac{1}{|S_i|} \sum_{s \in S_i} L_s \tag{3.6}$$
$$L_c = \frac{w_c C_c + w_r R_c + w_i I_c}{w_c + w_r + w_i} \tag{3.7}$$

where W_c is a weighted hate score calculated using the upvote-downvote score ($score_c$) of the comment and $hate_c$ is the hate prediction logit created by HateXplain, mapped to $(-1, 1)$. W_p is the weight of the parent comment. By mapping the hatefulness score to $(-1, 1)$, we allow benign comments (-1) to cancel out the hatefulness of other comments. As I_c depends on the labels of replies, we compute L_c depth-first, starting with the leaf nodes and aggregating to the root comment. We found that setting label weights $w_c = w_r = w_i = 1$ yielded the most balanced performance, explored in section 3.4.2.

Since L_c can have a wide range due to the incorporation of $score_c$, we quantized L_c into buckets of [< 0], [0-5], [6-20], [21-500], [500 <] to standardize L_c to a 0-4 ordinal scale of increasing propensity. Using ordinal values enables social media moderators to prioritize the most hateful content and deploy distinct policies for each ordinal class.

Finally, to analyze the model's ability to predict hate speech proactively, we use a subset of each labelled discussion tree as the model's input. We achieve this by taking a horizontal slice of the discussion tree, ensuring the tree has a maximum depth of 4 comments from the root. We then removed all comments that did not lead to a sub-discussion of at least two comments, ensuring a valid I_c signal was included in the label. As a result, the leaf nodes of our trimmed discussion tree are parent nodes of at least two other comments, requiring the model to predict the hatefulness of an unseen future discussion. Table 3.1 shows the number of comments before and after filtering for the four communities we selected, and Table 3.2 indicates the final distribution of labels.

Graphormer

Our approach modifies the Graphormer model of Ying et al. [2021] with the HateXplain model by Mathew et al. [2021]. Notably, the original Graphormer architecture was proposed and geared towards molecular modelling and graph-level predictions [Ying et al.,

Subreddit	Comments	Comments after Filtering
The_Donald	6 531 455	3 005 612
Politics	6 127 723	1 457 321
AmItheAsshole	4 135 270	1 381 200
IAmA	183 284	30 415

Table 3.1: Community Sample Size for the RedditHate Dataset

Subreddit / Label	0	1	2	3	4
AmItheAsshole	1 103 308	183 108	39 815	45 371	9598
IAmA	27 890	2 240	216	68	1
The_Donald	2 310 601	441 746	133 534	117 091	2640
Politics	1 087 286	264 570	54 311	45 563	5 591
Total	4 529 085	891 664	227 876	208 093	17 830

Table 3.2: Label Distribution

2021, Stienstra et al., 2024, 2025]. However, our task is to make node-level predictions of the propensity to engage in hateful conduct. Despite this, our focus on capturing the entire discussion context makes the Graphormer architecture appealing.

Given a discussion C with comments $c \in C$, each represented with text t_c, we start by encoding each comment using a frozen HateXplain BERT model. Concretely, we create initial comment embeddings as

$$c^{(0)} = BERT(t_c)$$

and utilize the final pooler token $c_{[CLS]}$ to represent each comment $c \in D$. This step establishes a foundational understanding of the text that comprises each comment.

After creating embeddings for each comment in the discussion, we deploy G Graph Transformer layers to aggregate contextual information from the other comments (Figure 3.2. Note that our use of Graph Transformers differs significantly from the original proposed use for molecular modelling [Ying et al., 2021]. We do this by representing the discussion as a graph, where each node is a comment, and edges connect parent comments to replies, akin to a molecule.

Since transformer layers are position-independent, Graph Transformers include two learned structure encodings. The first is Centrality Encoding, denoted z, which encodes node degrees in the graph [Ying et al., 2021]. Since social media discussion graphs are

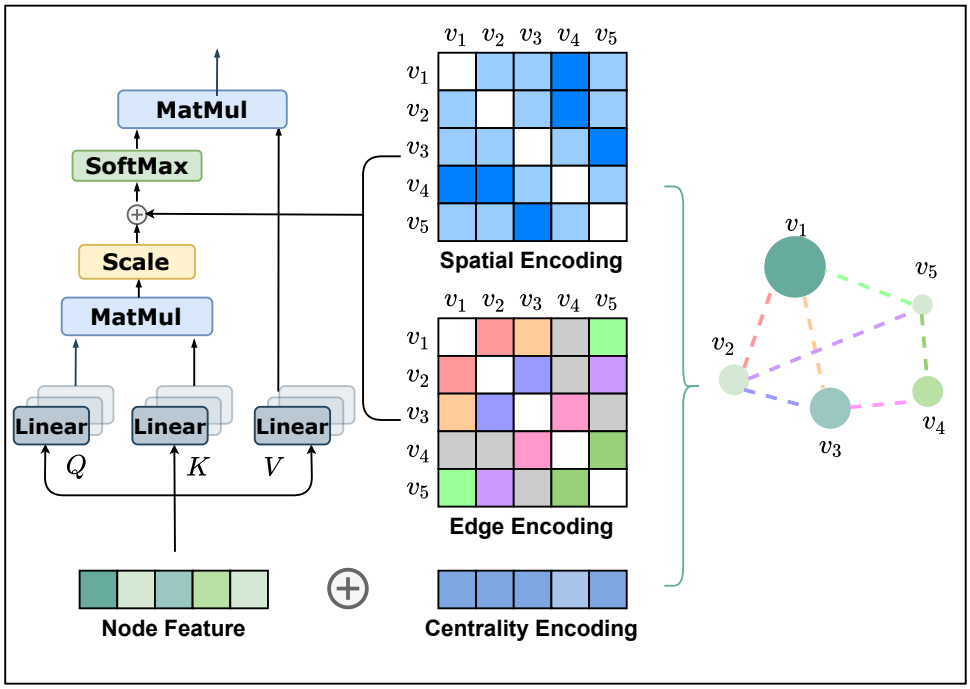


Figure 3.2: Graph Transformer Layer proposed by Ying et al. [2021]

directed, the degree of comments is equivalent to the number of replies a comment receives, plus one for the parent node. We implement this mechanism as

$$h_c^{(0)} = c^{(0)} + z_{deg(c)} \tag{3.8}$$

where $h_c^{(0)}$ is the initial embedding of c in the graph and $z_{deg(c)}$ is a learned embedding corresponding to the degree $deg(c)$ of the comment.

The second structure-encoding is Spatial Encoding, denoted $s_{(a,b)}$, which encodes the graph's structural relationship between nodes a and b in the graph. This encoding is added as an attention bias term during the self-attention mechanism. That is, we compute the self attention $A_{(a,b)}$ between nodes a, b as

$$A_{(a,b)} = \frac{(h_c \times W_Q)(h_v \times W_K)}{\sqrt{d}} + s_{(a,b)} \tag{3.9}$$

where W_Q and W_K are learned weight matrices and d is the hidden dimension of h. In this study, we follow previous work that utilizes Graph Transformers and adopt the shortest path between nodes as the spatial encoding. Here, this represents the number of turns between comments.

After G successive Graph Transformer layers, we take the final node embedding of each comment to represent our contextually grounded embeddings c^*, which we use for classification.

A naive approach to training Graphormer for the proactive hate speech task (section 3.4.1) would be to treat the 0-4 scale as a classification problem with five distinct classes. However, this would ignore the dependent ordinal nature of the labels and the risk of penalizing predictions that are close. For instance, given a ground truth label of $L_c = 3$, predicting either $L_c = 2$ or $L_c = 4$ would be more correct then predicting $L_c = 0$. To capture this dependency, we model the task as a regression problem using an L_2 loss, mapping the comment embeddings c^* to a single logit. Note that we produce predictions for every comment in the discussion in a single forward pass, amortizing the computational overhead of the graph transformer layers.

3.4.2 Findings

In our experimental evaluation, we first investigate whether global discussion context enables more accurate detection of future hateful discussions (results in Section 3.4.2). This is motivated by contrasting against other attempts at detecting hate speech, which often

Method	All	0	1	2	3	4
Graphormer	1.066	0.556	0.946	1.275	2.213	3.817
GAT	1.485	0.473	1.228	1.872	2.661	5.883

Table 3.3: Model Performance (L2 Loss) of Graphormer compared to GAT on each label, with All representing performance on the whole dataset

rely on just a single comment or the closest neighbours of a comment [Parmentier et al., 2021, Brassard-Gourdeau and Khoury, 2021, Huang et al., 2018]. Second, under the lens of modelling diverse discussion semantics, we investigate whether models trained on specific communities outperform those trained on all communities (results in Section 3.4.2). We then conclude with a sensitivity analysis of our hate labelling function in Section 3.4.2.

We use the standard Graphormer variant, consisting of 10 transformer layers with a hidden dimension of 769, as described in the original work. Additionally, we utilized the AdamW optimizer with a peak learning rate of 2e-4, which linearly decayed after a warmup period [Loshchilov and Hutter, 2018]. We applied an L2 regression loss for training and as an evaluation metric. This loss function is given as

$$L_2 = \sum_{i=0}^{n} (y - y_{pred})^2 \tag{3.10}$$

where n is the number of labels (in our case, 5), y is the ground truth label and y_{pred}.

This is important because our dataset is imbalanced towards smaller hate-label values. Due to computational constraints, a batch size of 16 was used over ten epochs, and only the Graphormer model was fine-tuned. To evaluate the model's performance, we split the comments into 10% for validation and 20% for testing, following the approach of Chen et al. [2018]. We utilize the validation set for hyperparameter tuning in our experiments. Each model was trained for ≈ 5 hours with 4 RTX 2080 Ti GPUs

Impact of Deeper Context

We begin by investigating the impact of including the entire discussion graph during inference. To measure this, we selected the Graph Attention Network (GAT) architecture [Veličković et al., 2017] as a baseline, as implemented in PyTorch Geometric [Fey and Lenssen, 2019]. This is inspired by previous efforts to utilize graph models to detect hate

Target Communities	Train	Test
All	1.060	1.066
The_Donald	0.4913	0.4911
Politics	0.4042	0.4041
IAmA	0.3687	0.3444
AmITheAsshole	0.4388	0.4397

Table 3.4: Model L2Loss Performance on Different Communities

speech on social networks, in particular Parmentier et al. [2021], which utilized this architecture. It is important to note that our use case differs: we seek to detect comments that will *lead* to hateful discussion. Similar to the Graphormer architecture, the GAT architecture utilizes an attention mechanism to aggregate neighbouring node embeddings. However, an essential difference in this architecture is that the aggregation operation is constrained to the direct neighbours in each layer. This is contrasted by our proposed approach, which performs self-attention aggregation over the entire graph at each layer.

We compare our approach against a two-layer GAT model, following the recommended configuration by the original authors and prior work [Parmentier et al., 2021]. This implies two iterations of the Aggregation-Combine operation. Table 3.3 shows the results: our approach (Graphormer) outperforms GAT by an average L2 loss of 0.419. Our results illustrate that both models perform well at classifying comments with a low level of future hateful discussion (0-1). This can be attributed to the prevalence of these comments in our dataset. However, notable differences in model performance are observed when detecting comments that lead to more prevalent hate speech (2-4), where our model outperforms GAT by an average L2 loss of 2.066.

We hypothesize that Graphormer's gains are due to its better contextualization within the discourse of larger discussion graphs, which are often strong indicators of future lively discussions. An example of this would be an instigating comment in a debate, where a better understanding of the entire debate could serve as a better predictor of what could be considered hateful and an instigating rebuttal. That said, it is worth noting that performance at the higher echelon is not optimal, with an average prediction inaccuracy of ≈ 2 ordinal levels; however, Graphormer still improves on GAT.

Impact of Community Specific Models

A key question in this work was whether hate could be better predicted by modelling the behaviours and cultural norms of specific communities. To evaluate this, we trained variants of the model on specific communities in our dataset. We compare these results with a general-purpose variant trained on data from all four communities (Table 3.4). We see that general models trained on all subreddits perform significantly worse than community-specific models. When comparing L2 loss, each community-specific model outperforms general models by a factor of two, for an average loss of ≈ 0.4. The `IAmA` subreddit benefited most from a community-specific focus, with an average L2 loss of 0.3687, followed by the `politics` subreddit.

We hypothesize that this performance boost came from two advantages. First, each community we selected is associated with vastly different topics: `The_Donald` focuses on alt-right politics, and `AmITheAsshole` on personal advice. As such, likely, comments that would be ignored in `The_Donald` could be inflammatory in other communities. A similar trend can be noticed in `politics`, which is politically left-leaning and would be more sensitive to hateful political discourse [Waller and Anderson, 2021]. By modelling specific communities, we hypothesize that our model captured these different cultural norms.

Second, some communities discuss in a formal, structured manner, such as in a question-and-answer format. `IAmA` is an example of this: a community that focuses on interviewing celebrities by asking public questions, in contrast to other communities that have free-flowing discussion. This difference was highlighted in our results, which showed that `IAmA` benefited most from a community-specific approach. By modelling the question-and-answer format and sampling the entire interview, we hypothesize that our model better understood the topics discussed and the unique conversational flow.

Sensitivity of the Labelling Function

We now investigate the sensitivity of the label weights w_c, w_r, w_i defined in the labelling function L_c used in our task, which combines the value of the Context-term C_c (label of the parent node), Reaction-term R_r (label of the current node) and Influence-term I_c (aggregated labels of child nodes) (equation 3.7). This evaluates the likelihood that our results depend on the chosen labelling weights. To do this, we created four variants of the dataset based on different weighting configurations: Equal, which weights each term equally; Influence-Weighted, Reaction-Weighted, and Context-Weighted, which double the weight of each term in the labelling function (weighted by 2 instead of 1). We conducted each experiment on the entire retrieved dataset.

Labeling Weight Variant	Train L2Loss	Test L2Loss
Equal ($w_c = w_r = w_i$)	1.060	1.066
Influence-Weighted ($w_i = 2$)	1.180	1.200
Reaction-Weighted ($w_r = 2$)	1.072	1.073
Context-Weighted ($w_c = 2$)	0.985	0.994

Table 3.5: Model Performance when using different labelling weights in equation 3.7, where all weights w_c, w_r, w_i are set to 1 unless stated.

Table 3.5 shows the results. Each variant performs similarly, with a Loss range of (0.994 - 1.200). As a result, we conclude that the labelling weights are not sensitive to specific values. We see that the Context-Weighted variant achieved the best performance, with a Test L2 loss of 0.994, whereas the worst-performing variant was the Influence-Weighted variant, with a Test L2 loss of 1.200. It is important to note that the Influence term for each node is calculated from the entire discussion tree, despite the input being constrained to a depth of 4, with each remaining comment being the parent of at least two other, potentially hidden, comments. As a result, the Influence-weighted variant is more difficult than the other variants.

3.4.3 Takeaways

The scope of this work differs in several ways from previous work. First, we studied the influence of communities on what content is shared and how it is reacted to. We accomplished this by including the entire discussion graph during inference and creating specific models for individual communities. This scope differs from traditional hate speech systems that examine a comment's text in isolation, without considering its discussion context or the community. As we have shown in our experimental results, including more context, in terms of the discussion and the community, can lead to better performance.

Second, the model's output differed from previous work by predicting ordinal values from 0 to 4, denoting the intensity and likelihood of a hateful discussion following an initial comment. There are two benefits to this objective. First, as the scope of social platforms grows, it becomes essential for moderators to prioritize which content to investigate first and which to quarantine immediately. By including a range of intensity values, we also support users of these platforms in choosing an intensity level that is comfortable for them. This would allow users with mental health challenges to self-regulate the content they want to see without restricting the free speech of users who are more comfortable seeing intense

content.

3.5 Extended Qualitative Study of Graphormer

In this section, we conduct a detailed qualitative analysis of Graphormer (section 3.4) for hate speech detection in social networks, offering insights into where the method yields the most impressive outcomes compared to competitors and identifying scenarios where challenges arise in achieving ideal performance. In particular, we highlight the key observation that comments on social platforms have evolved to incorporate images and external articles. These additional elements can provide essential context for understanding the content that follows.

We will conclude with a discussion of how to extend our method to include image processing, ultimately enabling holistic processing of online posts. We will also revisit the concern of mental health well-being and discuss how our more comprehensive automated solution for hate speech prediction can serve as a basis for significant steps forward.

3.5.1 Baselines

Comment-Only Hate Speech Models

To evaluate recent work in hate speech detection, we selected MuRIL by Das et al. [2022] and Bert-HateXplain by Mathew et al. [2021]. Both systems are based on the BERT transformer architecture, which can generate rich text embeddings for classification tasks [Devlin et al., 2019]. We refer to these methods as comment-only hate speech models.

The primary difference between the two methods lies in the data on which both systems were trained. For Bert-HateXplain, the authors collected a combined dataset of 20,148 hateful tweets and posts from social platforms Twitter and Gab. For MuRIL, the authors combined the HateXplain dataset [Founta et al., 2018] (85,775 tweets) with the datasets of Davidson et al. [2017] (24,783 tweets). This combined approach was found to outperform HateXplain to become state-of-the-art in hate speech detection [Das et al., 2022].

Graph Hate Speech Models

In addition to evaluating Graphormer (section 3.4), we also assess a baseline approach that uses Graph Attention Networks (GAT) [Veličković et al., 2017]. GAT models have previ-

ously been adapted for hate speech detection by Parmentier and Cohen [2019], Parmentier et al. [2021]. Like Graphormer, GAT models use attention to create node embeddings within a graph. However, this attention is constrained to direct node neighbours, a limitation that Graphormer overcomes by using transformers. This can result in predictions that focus more on the immediate discussion context than on the broader global context. Other work has examined graph-based approaches for hate speech detection [Mishra et al., 2019, Tian et al., 2022]; we focus here on comparisons with the models described above.

3.5.2 Experimental Setup

Like in section 3.4, we focus on Reddit as the social platform for this study. We are particularly interested in analyzing Reddit, as prior work on Reddit communities has found that they exhibit significant differences in their social makeup and communal behaviours [Waller and Anderson, 2021]. For example, communities such as r/conservatives demonstrate a right-leaning bias, whereas r/politics contains a polarizing left-leaning bias. This provides a unique environment to test how our Graphormer behaves across vastly different communities.

We analyzed 32 discussions from 16 different communities centred around contentious topics and unique communities. Each discussion was chosen based on the number of comments it had. We also draw from Reddit conversations sampled in Kurrek et al. [2020] for examples of reclaimed speech. From the discussions sampled, we selected five interesting examples to display in the section that follows.

For each comment in the discussion, we predict labels ranging from 0 to 4 using comment-only and graph-based hate speech models. To match the ordinal predictions given by Graphormer (section 3.4), we map the zero-to-one prediction given by comment-only models to bins of width 0.20 ([0-0.20], [0.20 - 0.40], [0.40 - 0.60], [0.60-0.80], [0.80-1]). To capture graph methods' ability to adapt to evolving conversations, we initialize the discussion graph with the initial post and its immediate replies (depth 1). We then iteratively predict each comment's label by gradually increasing the depth of the discussion tree provided to the graph models. As such, graph methods are limited in predicting the direction of conversations without considering future comments.

3.5.3 Findings

We focus on capturing examples from two categories of conversational hate speech. First, we begin with samples of contextual hate speech, which are harmful comments that directly

refer to or respond to the prior discussion. Examples of this kind of hate speech would be a detrimental commentary on contentious topics, such as reacting negatively to gay rights. We hypothesize that comment-only methods would fail to capture the contextual nuances that underpin the hatefulness of the target text. This can result in false positives or false negatives when comments are judged in isolation.

The second category of hateful speech we study is incisive hate speech: comments that at first glance appear to be neutral but are designed to prompt harmful discourse by other users. These examples aim to evaluate the ability of graph hate speech models to proactively predict the direction of conversations toward hatefulness, rather than just detecting hate in individual comments. As a result, this direction also evaluates the ability of graph methods to capture the dynamic nature of social media discussions, where conversations are not static but grow over time as users add replies to content posted by others. Examples of this kind of speech would be comments concerning U.S. President Biden in right-leaning political subreddits, which can prompt hateful comments and threats.

Contextual Hate Speech

We start by analyzing a conversation that took place on the subreddit /r/gay, a community centred around LGBTQ topics (Table 3.6). In this conversation, users discuss a tweet from openly gay pop artist Lil Nas X. Our analysis of this discussion thread shows how a lack of context can lead to false predictions about the tweet's hatefulness.

The conversation begins with the initial post "Anyone else loving Lil Nas X meming on biggots", referring to his tweet stating "i thought y'all didn't like political correctness. what happened?". This comment resulted in a high prediction of hatefulness by comment-only BERT-HateXplain and a moderate prediction by the Graph methods. However, the hate predictions from the graph methods quickly neutralize as the conversation shifts into a discussion of his latest song, "Montero" (depths 5 and 6). Given this context, it is clear that these methods are not hateful but rather discuss the song's lyrics. However, this context is not available to comment-only methods. The inaccuracy introduced without this crucial context is evident in the comment-only techniques, where both systems predicted high hateful scores for both comments. This illustrates the ability of graph methods to maintain conversational context when predicting hate scores.

Next, we turn to an example of a discussion that requires context to detect hate speech. For this, we focus on the subreddit /r/MensRights. This community advocates for increased men's rights by discussing social issues that adversely impact them, which frequently devolves into harmful misogyny [Schmitz and Kazyak, 2016]. This pattern of

Depth	Text	Graphormer	GAT	Bert-HateXplain	MuRIL
0	Anyone else loving Lil Nas X meming on biggots? [image]	2	1	3	0
1	What was this in reference too?	1	1	0	0
2	Biggots are mad about his most recent song/music video	1	2	1	0
3	Ah ok! I'll have to check it out!	0	0	0	0
4	The song is called Montero (call me by your name).	0	0	0	0
5	My favorite line is "I wanna f*ck you cuz I envy you", and that hit hard	0	0	2	4
6	'I want that jet leg from f*ckin' and flyin' call me in	0	1	3	4

Table 3.6: Conversation on r/gay containing contextual speech leading to false positive predictions

Depth	Text	Graphormer	GAT	Bert-HateXplain	MuRIL
0	How to get banned from r/Feminism [image]	3	3	0	0
1a	i guess they were on their period and they want their feelings to matter at that point and you got banned for it	3	3	1	0
1b	Feminism is cancerous anyways	3	2	1	0
1c	Wow, they are almost as fragile as the donald. I guess all special snow flakes need their own safe space.	1	2	1	0
1d	Absolutely horrible.	1	0	1	1

Table 3.7: Conversation on /r/MensRights containing contextual affiliation-directed hate speech

abuse is evident in the discussion presented in Table 3.7. Here, the user posts an image of a brief exchange from the r/Feminism subreddit. In this exchange, the user was banned from that community for advocating that women have the right to *feel* safe, rather than the right to *be* safe, regardless of reassurances and comfort. By posting this exchange on r/MensRights, the user aimed to frame the r/Feminism community in a negative light, claiming they were banned for sharing a valid point.

In this discussion, graph methods enabled an accurate understanding of the negative context of feminism in response to the hateful comments that followed. Most surprisingly, none of the comments that followed were labelled as hate speech by the comment-only methods. Such examples include "Feminism is cancerous anyways", which was appropriately labelled as hateful by graph methods, receiving a prediction of 3 from Graphormer, but mislabeled as innocuous by text methods. We assume that this significant difference in prediction comes uniquely from the graph structure of the discussion.

Finally, we examine a conversation that incorporates reclaimed language previously perceived as harmful to a specific community. For this, we sample a discussion from /r/rupaulsdragrace, a community dedicated to the LGBTQ drag competition Ru Paul's Drag Race (Table 3.8). In this conversation, users are commenting on one competitor's manufactured drag outfits. In this community, slurs such as "f*ggot" and "b*tch" are reclaimed as favourable terms to refer to LGBTQ members and competitors[2]. However, these slurs are more often used in a hateful context, proving a challenge for methods that do not consider specific contexts.

Analyzing the predictions of each method, we see that comment-only methods assign high hate scores to comments at depths 1a and 2b. However, both comments are, in fact, positive and supportive of the competitor mentioned in the initial post. Upon inspecting the content of these comments, we can infer that the false optimistic prediction is likely due to the use of reclaimed slurs without contextualizing them within the community and prior discussion. This behaviour is contrasted with graph methods, which yielded more accurate scores, likely because other comments informed them about the dress's fashion.

It is important to note that both the graph and comment-only methods can only infer context by inspecting the text of comments. In this conversation, graph methods were likely able to infer context from the other comments, which discuss the dress's fashion (depths 1b and 1c). However, upon reviewing the initial post, we see that the user included an image of a drag queen (Figure 3.3). This picture provides immense context for the discussion, particularly its focus on fashion and the LGBTQ community. As such, we hypothesize

[2]A common slogan on the show is "Yass b*tch", which is used to cheer on competitors

Depth	Text	Graphormer	GAT	Bert-HateXplain	MuRIL
0	*SPOILERS* Always and Forever, paparazzi who? [image - Figure 3.3]	1	1	1	0
1a	Am I the only f*ggot that LIVED for this look?	1	2	4	3
2a	I honestly truly thought it was an amazing concept and I love the final result	0	0	1	0
2b	Not at all. F*ck fashion. I want fashion I'll by the Vogue fall guide. Give me something creative I haven't seen before.	0	0	2	4
1b	Say what you will about the look, but can we appreciate the fact that this b*tch got over a dozen Canon DSLRs plus the lenses on this dress? That shit ain't cheap.	1	1	3	4
1c	This look is ridiculous. I love it, but my God, who thinks of this shit.	0	0	1	4

Table 3.8: Conversation on r/rupaulsdragrace containing reclaimed language and multimodal context



Figure 3.3: Photo of the Drag Queen discussed in Table 3.8

that future work incorporating multi-modal posts could provide important context for the comments that follow.

Inciteful Hate Speech

In our next set of examples, we investigate graph networks' ability to predict the direction of conversations. We start by analyzing a conversation that took place on /r/conservatives, a community for discussing right-wing policies (Table 3.9). These policies often include advocacy for gun rights and strong support for election denial [Block, 2021]. Discussions in these political communities have become increasingly polarized and hateful towards members of the opposite party [Waller and Anderson, 2021].

In this discussion, users are referring to a tweet from a black user concerning the difficulty for people to purchase weapons due to a lack of governmental IDs. The conversation is centred around confounding the Black Lives Matter movement with gun advocacy and election denial ("Black Guns Matter, as does election integrity"). The discussion then devolves into affiliation-based hate as users claim left-leaning users are racists and associate black activist groups ("the black panthers" and "BLM") with armed violence.

Examining the predictions of both groups of methods, we observe that both graph methods assign very high hatefulness scores to many of the comments in the discussion. This is especially evident in the comments at depth 2, where users delve into accusations of racism. However, it is essential to note that these comments are more akin to debate rather than explicit hate speech. This is reflected in the predictions from the comment-only methods, which consistently predict low hatefulness scores for these comments. As such, it can be inferred that these high predictions stem from a belief that the conversation will take a hateful turn, given the context thus far. Indeed, this is the case as later comments (depth 5) intensify the discussion and accuse members of the Black Lives Matter movement of armed violence. However, it is also important to note that the earlier predictions appear to conflate polarizing political discourse with hate (depth 1) without providing sufficient context for discussion.

To further examine the ability of Graphormer to predict the direction of conversations, we investigate a conversation from /r/politics. This community is known to have strong left-leaning political views and a distinct, polarized user base compared to/r/conservatives [Waller and Anderson, 2021]. In the sampled conversation (Table 3.10), users discuss comments made by former President Trump regarding the Ukraine war. The conversation begins benign but becomes combative, with one user shifting the blame for the Ukrainian

Depth	Text	Graphormer	GAT	Bert-HateXplain	MuRIL
0	Black Guns Matter, as does election integrity [image]	3	1	0	1
1	That's Hilarious. I think its interesting how the left keeps trying to rub "BLACK PEOPLE ARE BUYING GUNS" in the faces of conservatives, like we would somehow be opposed to that. [...] glad that gun ownership is expanding among all demographics	4	3	1	2
2a	It's on every conservative news and media platform as a big positive, but the left doesn't pay attention to that literally all they think is conservatives racist, therefore black guns bad for them.	3	2	1	2
2b	They want us to be divided by race like they are. They are so racist they can't imagine us being united by our love of fundamental rights	4	3	0	0
	[...]				
4	I agree. The black panthers had the right idea. BLM members should arm too.	2	0	0	0
5	They have been.... smh	4	1	0	0

Table 3.9: Conversation from r/conservatives containing inciteful speech regarding Black Lives Matter

Depth	Text	Graphormer	GAT	Bert-HateXplain	MuRIL
0	Trump, who was impeached for withholding nearly $400 million in military aid from Ukraine, said 'this deadly Ukraine situation would never have happened' if he were in office [article]	2	2	0	0
1	This happened under Biden's watch. That is a fact	3	3	0	0
2a	Russia has been threatening Ukraine for the last 8 years	2	0	0	0
3	Whatever, still happened under Biden's watch. Not Trump's.	3	2	0	0
4	Right, I'm sure the situation would've been so much better under the leadership of a failed jackass lapdog for Putin.	4	3	3	0
2b	You might want to do a little reading about Ukraine. Your comment is completely ludicrous	3	3	0	0

Table 3.10: Conversation from r/politics requiring long-range forecasting from community cues

war onto the current left-leaning president. This trend culminates in a climax at depth 4, where the user escalates to using hateful language against Trump.

Investigating the predictions, we observe a similar trend to the previous example: each prediction from the comment-only methods remains neutral, except for the hateful comment at depth 4. However, the contentiousness of the conversation, given the previous instigating comments, is captured by graph methods, resulting in high predictions by both methods. This is especially evident in the prediction at depth 1, which appears to capture the contentious relationship between comments concerning Biden and Trump. Indeed, we can see that the conversation did turn hateful later on, validating this prediction.

3.5.4 Discussion

In our analysis, we focused on analyzing two types of difficult hate speech: contextual and incitement. Contextual hate speech requires conversational context to understand, and incisive hate speech is not inherently hateful but is designed to incite further hateful comments. Both types of speech pose difficulties for current comment-only approaches, which rely heavily on context to make accurate predictions.

Starting with contextual hate speech, we analyzed three different conversations originating from /r/gay, /r/MensRights, and /r/rupaulsdragrace. Using comment-only methods, we identified numerous predictions that were either false positives or false negatives, depending on the comment's text in isolation. For false positives, we found that comment-only methods tended to predict high hate scores for comments that contained slurs (Table 3.6 and Table 3.8). However, upon reading the rest of the conversation, it becomes clear that many of these slurs are utilized in a non-derogatory context. The same can be said for false negatives, where antagonistic replies can lose their hateful context when considered in isolation (Table 3.7). However, in each of these conversations, we found that graph methods perform well in capturing the vital discussion context required to understand these comments appropriately. Between the two graph methods we evaluated, both GAT and Graphormer performed similarly well in the examples we explored.

To examine the ability of graph and comment-only methods to capture incisive speech, we analyzed two discussions from /r/conservatives and /r/politics, communities with polarizing user bases [Waller and Anderson, 2021]. We found that graph methods are sensitive to counter-speech as evidence of inciting hateful discourse. This is especially evident in Table 3.10, where users who disagree on the cause of the Ukraine war yield high hatefulness predictions from both graph methods. While there may be some validity to these predictions about their contentiousness, this raises concerns about how to moderate these



Figure 3.4: Examples of multi-modal hate speech, where hatefulness can only be discerned by jointly considering both modalities (taken from Kiela et al. [2020])

heated debates. However, in the case of the example in Table 3.9, the most inciteful comments (depth 2a, 2b, and 5) are appropriately labelled as such. In each of these cases, comment-only methods predicted each comment as neutral, even if the comments were hateful (depth 2b). We found that Graphormer was more sensitive to higher predictions than GAT when faced with these types of comments.

In each example, we also evaluated graph networks' ability to adapt to evolving social media conversations. Coping with this dynamic change is crucial for the successful implementation of AI for social impact in real life. We evaluate this behaviour by iteratively predicting comments in the discussion graph depth-wise, unlike Graphormer (section 3.4). As a result, we constrain graph models to predict labels solely based on the context provided by previous comments, mirroring how the system would be deployed in real-world settings. Despite this constraint, we observe that graph systems can still make accurate predictions. This is best illustrated in Table 3.6, where the graph models adjusted their predictions to become less hateful as the conversation progressed.

We also found that many of the examples we retrieved included multimodal information. Such examples include the discussion in Table 3.6, which involves an image of a tweet, and the discussion in Table 3.10, which concerns an article about Trump and the Ukraine war, among others. When investigating contextual hate speech, Table 3.8 presents an example in which the image (Figure 3.3) provides essential context for the comments that followed.

An analysis of this picture reveals that the discussion centers on an LGBTQ drag queen competing in an elaborate dress. However, without this context, we found that comment-only methods misclassified supportive speech using reclaimed LGBTQ vernac-

ular as hateful. This is especially concerning given that these predictions could serve to suppress communities that are vitally important to the mental health of minority populations [Lucero, 2017, Fish et al., 2020]. Furthermore, memes sent on online platforms are often only hateful if one considers both the image and the text caption together, as seen in Figure 3.4 [Kiela et al., 2020]. By taking a holistic view of conversations by encoding images, text, and discussion structure together, we hypothesize that hate speech detection methods will be able to avoid many false predictions, such as those incurred in Table 3.8. Furthermore, following Tian et al. [2022], it would be possible to include user-level information into this graph representation.

Finally, it is also essential to analyze the mental health impact given by a graph approach to hate speech. By reformulating hate speech as a graph prediction task, we can train systems that leverage discussion context to predict the direction of conversations. This can allow moderators on social platforms to be alerted to potentially harmful comments and deploy mitigation strategies to shield users who are susceptible to mental health effects. We see an example of such a discussion in Table 3.9, where users who are susceptible to trauma from guns and race can be warned ahead of time by utilizing the proactive graph predictions. Furthermore, by using an increasing ordinal scale (from zero to four) for predicting hate, users can select their level of comfort by choosing the intensity of contentious comments they are comfortable viewing. As the conversation develops, these predictions can then be updated with further context and revised accordingly. An example of where this would be useful is the discussion in Table 3.8, where additional comments add credence to the innocence of previous comments. By providing these scores, platform owners can enable users to control the content they see through self-moderation. Another valuable opportunity for deploying our methods, as demonstrated by qualitative analysis, is to assist platforms in curbing the proliferation of hate speech. This can be achieved by greater prediction of impending escalating harm and caution in imposing penalties when the discussion is hate-based.

3.5.5 Takeaways

In this section, we explored the impact of Graph Transformer Networks on hate speech detection. To achieve this, we conducted an extensive qualitative analysis of graph- and comment-only methods in conversations from various Reddit communities. When examining contextual hate speech, we found that Graph Transformer Networks can prevent both false positives and false negatives incurred by comment-only methods. In these cases, context played a key role in understanding the nature of the analyzed comments. We also

found similar performance gains when analyzing discussions about incisive speech. However, we also found that debates were prone to high levels of hate prediction despite being mostly civil.

One key insight we discovered was the need to incorporate additional modalities to further contextualize comments. Among the examples we retrieved, many centred on an image or an article. We hypothesize that adopting a holistic view of conversations, encompassing all modalities, can help prevent false positives. These findings motivate our work with mDT.

3.6 Multi-Modal Hate Speech Detection

As discussed in the previous section, other modalities such as images can dramatically affect semantic meaning [Kiela et al., 2020, Hebert et al., 2021]. The connection between text and image modalities has led to several works exploring multimodal methods, models that jointly process text and images. CLIP, proposed by Radford et al. [2021], aligns text and image embeddings generated by separate modality-specific models with contrastive pre-training. ViLBERT [Lu et al., 2019] and Flamingo [Alayrac et al., 2022] also use distinct text and image transformer models, but employ cross-attention across both modalities to share information within each layer. Building on cross-attention, VilT [Kim et al., 2021] proposes merging input word tokens and image tokens into a single sequence, which is then processed by a single unified transformer encoder. On the other hand, some methods have focused on improving the computational efficiency of multimodal models, such as Nagrani et al. [2021], which proposed utilizing shared bottleneck tokens and constraining the shared cross-attention to use only those tokens.

These innovations in multi-modal language grounding have also enriched hate speech detection. Focusing on the HatefulMemes challenge, Velioglu and Rose [2020], Muennighoff [2020], Zhu [2020] propose various combinations of ensemble models, each encoding different image and text features, which are then combined at the output level. Liang et al. [2022] employs graph convolutional networks to merge text and image tokens, primarily for sarcasm detection. Meanwhile, Sahu et al. [2021] incorporates inter-modality information by adapting GAN-Fusion [Sahu and Vechtomova, 2021] towards hate speech detection, utilizing generative adversarial networks to encode and align modality embeddings [3]. Cao

[3]In Sahu et al. [2021], each generator model aims to reproduce the embedding created by the other modality using the encoding of the first modality as input ($z_{image} \rightarrow z'_{text}$, $z_{text} \rightarrow z'_{image}$). The discriminator is tasked to distinguish generated embeddings z' from their native representations z

et al. [2022] incorporates multiple modalities by mapping meme images to text descriptors using a generative model, which is then appended to the comment text for hate speech classification. Finally, Singh et al. [2022] incorporates image and text representations of product reviews to accurately disambiguate complaints.

Chapter 4

Discussion Transformers: Integrating text, images and graph transformers for Holistic Discussion Analysis and its Applications for Hate Speech Detection

Social media has democratized public discourse, enabling users worldwide to express their opinions and thoughts freely. As of 2023, the social media giant Meta has reached 3 billion daily active users across its platforms [Meta, 2023]. While this level of connectivity and access to information has led to many benefits, it has also resulted in the alarming rise of hate speech [Das et al., 2020]. The pervasive spread of hateful rhetoric has caused significant emotional harm to its targets [Vedeler et al., 2019], triggered social divisions and polarization [Waller and Anderson, 2021], and has caused substantial damage to the mental health of users [Wachs et al., 2022]. These findings emphasize the urgent need for a comprehensive solution to automate the identification of hate speech as a critical first step toward combating this alarming practice.

Initially, automated hate speech detection models were limited to text-only approaches such as HateXplain [Mathew et al., 2021], which classify the text of individual comments. Such methods have two significant weaknesses. First, hate speech is contextual. Social media comments are often conversational and influenced by others' comments in the discussion thread. For example, a seemingly innocuous comment such as "That's gross!"

can become hateful in a discussion about immigration or minority issues. Second, social media comments have evolved to include images, which can influence the context of the accompanying text. For instance, a comment may be innocuous, but including an image may transform it into a hateful remark. While recent work has expanded transformers to include images, such as ViT [Kim et al., 2021] and CLIP [Radford et al., 2021], the discussion's context remains an underexploited resource.

In this chapter, we propose the Multi-Modal Discussion Transformer (mDT) architecture to solve this challenge. Multi-Modal Discussion Transformers enable holistic representations of comments by incorporating macro-level multi-modal discussion context. To evaluate our work, we also present a novel dataset, HatefulDiscussions, containing complete multi-modal discussion graphs from various Reddit communities and a diverse range of hateful behaviour.

We compare mDT against various comment-only text models and a seminal text-only variant of the Discussion Transformer architecture, Graphormer, from which mDT takes its roots. We also conduct an ablation study of the components of our architecture. We conclude by discussing the potential of our model to deliver social value in online contexts by effectively identifying and combating antisocial behaviour in online communities. We also propose future work towards more advanced multimodal solutions that can better capture the nuanced nature of online behaviour.

To summarize our contributions:

1. We propose a novel fusion mechanism at the core of mDT that interweaves multi-modal fusion layers with graph transformer layers, enabling multimodal comment representations that are actively grounded in the discussion context.

2. We propose a novel graph structure encoding that captures the hierarchical conversational structure of social media discussions.

3. We introduce a dataset of 8266 annotated Reddit discussions, totalling 18359 labelled comments, with complete discussion trees and images to evaluate the effectiveness of the Discussion Transformer architecture.

The code and datasets for mDT are available at https://github.com/liamhebert/MultiModalDiscussionTransformer, and we provide a revised general-purpose Discussion Transformer library at https://github.com/liamhebert/MDT-2.

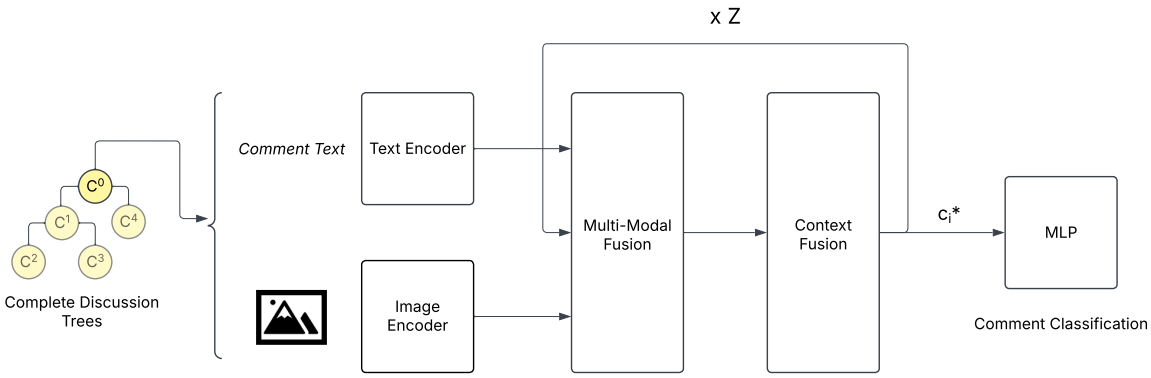


Figure 4.1: General architecture for Discussion Transformers. Z is the number of sequential Graph Transformer and Multi-Modal Fusion Blocks.

4.1 Methodology

4.1.1 Discussion Transformers

This research proposes a novel paradigm for grounding the semantic meaning of comments within the discussion context, called Discussion Transformers. The general task solved by Discussion Transformers is as follows: Given a discussion graph C consisting of comments $c_i = (t_i, i_i)$ composed of text t_i and images i_i, the goal of Discussion Transformers is to create contextually grounded embeddings of each comment $c_i^* \in C*$ that capture conversational context. Then, each c_i^* can be used for downstream comment classification tasks, such as hate speech detection.

At a high level, Discussion Transformers can broadly be broken down into three key components, illustrated in Figure 4.1.

- **Initial Encoding**: We first create an initial embedding of each comment in the discussion, irrespective of the surrounding context. This step aims to interpret the surface meaning of each comment modality, which can later be used for context grounding.
- **Context Fusion**: After creating initial embeddings for each comment, this step integrates context from other comments to ground and adjust the semantic meaning of each comment.

- **Classification**: Finally, after processing Z context fusion layers, we obtain our final comment representations c_i^*. We apply a classification head to each c_i^* comment to predict whether the comment is hateful or not.

In summary, the architecture of Discussion Transformers is structured as Initial Encoding $\rightarrow$ Context Fusion Layers $\times Z \rightarrow$ Classification, where Z denotes the number of Context Fusion layers to apply. It is this paradigm that the Multi-Modal Discussion Transformer builds upon.

4.1.2 Multi-Modal Discussion Transformer (mDT)

Our proposed mDT architecture expands upon Graphormer (Section 3.4) through three key innovations. First, we incorporate mechanisms for multimodal representations of comments, including images as contextual signals. Second, we update the context fusion layers to instead intertwine with the comment modality layers via a process called mid-fusion. This enables active grounding of comment representations in context as they are processed, rather than grounding them post hoc. Finally, we also update the graph positional encodings $s_{(a,b)}$ to capture the hierarchical nature of discussions better.

Following the framework in Section 4.1.1, mDT comprises an Initial Pre-Fusion stage, followed by Modality Fusion and Context Fusion stages that inject context.

Initial Pre-Fusion

Given a discussion D with comments $c \in D$, each represented with text t_c and optional image i_c, we start with pre-trained BERT and ViT models to encode text and images, respectively. Both models consist of N layers with the same hidden dimension, d. In our experiments, we used BERT-base and ViT-base, each with $N = 16$ layers and $d = 768$ hidden dimensions. Given these models, the Initial Pre-Fusion step consists of the first K layers of both models with gradients disabled (frozen), denoted as

$$t_c^k = BERT_{init}(t_c), i_c^k = ViT_{init}(i_c) \tag{4.1}$$

where $K < N$. This step establishes a foundational understanding of the images and text that comprise each comment.

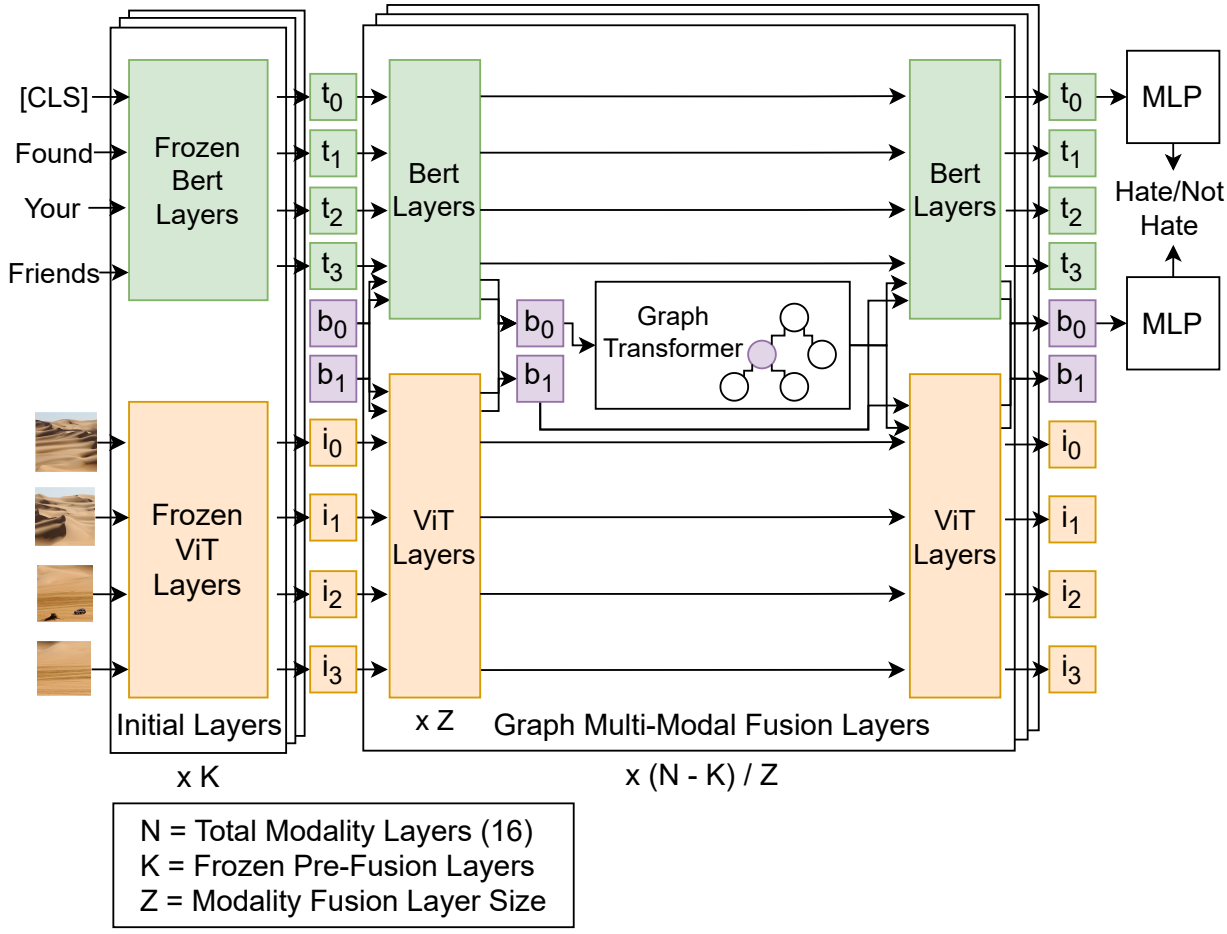


Figure 4.2: Multi-Modal Discussion Transformer

Modality Fusion

After creating initial text and image embeddings t_c, i_c for all comments $c \in D$ in the discussion, we move to the Modality Fusion step. We adopt the bottleneck mechanism proposed by Nagrani et al. [2021] to encode inter-modality information. We concatenate b shared modality bottleneck tokens $B \in R_{b \times d}$ to t_c and i_c, transforming the input sequence to $[t_c^k \mid\mid B], [i_c^k \mid\mid B]$. We then define a modality fusion layer l as

$$[t_c^{l+1} || B_{t,c}^{l+1}] = BERT_l([t_c^l || B_c^l]) \tag{4.2}$$

$$[i_c^{l+1} || B_{i,c}^{l+1}] = ViT_l([i_c^l || B_c^l]) \tag{4.3}$$

$$B_c^{l+1} = Avg(B_{t,c}^{l+1}, B_{i,c}^{l+1}) \tag{4.4}$$

where both modalities only share information through the B bottleneck tokens. This design forces both modalities to compress information to a limited set of tokens, improving performance and efficiency. If no images are attached to a comment, then $B_c^{l+1} = B_t^{l+1}$.

Context Fusion

Then, after Z ($< (N - K)$) modality fusion layers, we deploy Graph Transformer layers to aggregate contextual information from the other comments in the discussion. Given that the tokens in B_c encode rich inter-modality information, we leverage these representations to represent the nodes in our discussion graph. Using $b_c^0 \in B_c$ to represent each comment $c \in D$, we aggregate each embedding using a transformer model to incorporate discussion context from other comments. Our novel use of bottleneck tokens to represent graph nodes enables modality models to maintain both a modality-specific pooler token ([CLS]) and a graph-context representation (b_0).

Since transformer layers are position-independent, we include two learned structure encodings. The first is Centrality Encoding, denoted z, which encodes node degrees in the graph [Ying et al., 2021]. Since social media discussion graphs are directed, the degree of comments is equivalent to the number of replies a comment receives, plus one for the parent node. We implement this mechanism as

$$h_c^{(0)} = b_c^0 + z_{deg(c)} \tag{4.5}$$

where $h_c^{(0)}$ is the initial embedding of b_c^0 in the graph and $z_{deg(c)}$ is a learned embedding corresponding to the degree $deg(c)$ of the comment.

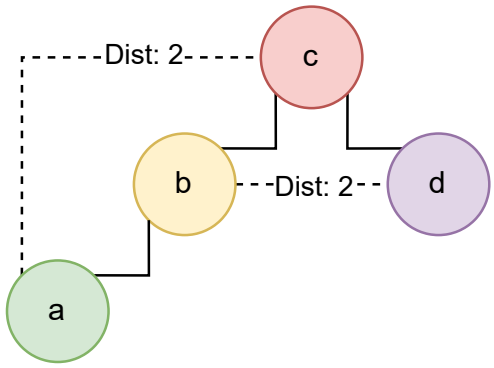


Figure 4.3: Example Discussion Structure. Each node in the discussion tree represents a comment. The shortest distance between (a, c) and (b, d) is equivalent, which demonstrates a lack of expressiveness for hierarchy.

The second structure-encoding is Spatial Encoding, denoted $s_{(c,v)}$, which encodes the graph's structural relationship between nodes c and v. This encoding is incorporated as an attention-bias term within the self-attention mechanism. That is, we compute the self attention $A_{(c,v)}$ between nodes c, v as

$$A_{(c,v)} = Softmax\left(\frac{(h_c W_Q)(h_v W_K)}{\sqrt{d}} + s_{(c,v)}\right) \tag{4.6}$$

where W_Q and W_K are learned weight matrices and d is the hidden dimension of h.

In previous graph transformer networks, $s_{(c,v)}$ is encoded as a learned embedding representing the shortest distance between nodes c and v in the graph [Ying et al., 2021]. However, this metric is ill-suited to the hierarchical structure of discussions, in which equivalent distances can represent different interactions. This is best seen in the example discussion illustrated in Figure 4.3. When utilizing the shortest distance to encode structure, the distance between nodes a and c is the same as the distance between nodes b and d in this graph. However, b and d represent direct replies to the same parent post, whereas a is two comments underneath c.

We propose a novel hierarchical spatial encoding based on Cantor's pairing function to account for this. Cantor's pairing function uniquely maps sets of two numbers into a single number $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Given comments a and b, we first calculate the number of hops upward $u_{(a,b)}$ and hops downward $d_{(a,b)}$ to reach b from a. In the example above, the distance between a and d is $u_{(a,d)} = 2, d_{(a,d)} = 1$. We then compress both numbers into a

single index using the proposed position-independent variant of Cantor's pairing:

$$s_{(c,v)} = s_{(v,c)} \tag{4.7}$$
$$= Cantors(u, d) \tag{4.8}$$
$$= \frac{(u+d)(u+d+1)}{2} + min(u, d) \tag{4.9}$$

This uniquely maps $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $s_{c,v} = s_{v,c}$. We utilize this function to index learned spatial embeddings in the self-attention mechanism.

After G graph transformer layers, the final representation of h_c^G replaces b_c^0 for the next set of Z modality fusion layers. We denote the combination of Z Modality Fusion and G Graph Transformer layers as a Graph Multi-Modal Fusion module. Finally, after $(N-K)/Z$ Graph Multi-Modal Fusion modules, we predict logits using the final embedding of b_c^0 and the [CLS] embedding of t_c. This novel interweaving of graph transformers and fusion layers via modality bottleneck tokens ensures that fusion models produce representations grounded in the discussion context. Notably, this differs from previous approaches that utilize graph neural networks, which sequentially process individual comments before applying a set of graph layers.

4.1.3 HatefulDiscussions Dataset

To train our model, we require a complete multimodal discussion graph dataset. However, datasets used by other works [Mathew et al., 2021, Zampieri et al., 2019, Kiela et al., 2020] consist of individual labelled comments and are predominantly text-only. To address this issue, we curated a novel benchmark comprising multiple datasets annotated by human annotators and augmented it to include complete multimodal discussion graphs. Our final dataset comprises 8266 Reddit discussions with 18359 labelled comments from 850 communities. Note that our architecture can be extended to other platforms, such as Facebook, Twitter, and TikTok, since they also use tree structures. Since discussions on these platforms are typically smaller with less complexity, using Reddit allows us to best stress-test our model.

The first type of hate speech included in our benchmark is Identity-Directed and Affiliation-Directed Abuse. We retrieved labelled examples of this from the Contextual Abuse Dataset (CAD) developed by Vidgen et al. [2021a]. According to the authors, Identity-Directed abuse refers to content containing negative statements against a social category, encompassing fundamental aspects of individuals' community and sociodemographics, such as religion, race, ethnicity, and sexuality, among others. On the other

hand, Affiliation-Directed abuse is defined as content expressing negativity toward an affiliation, where an affiliation is a voluntary association with a collective, such as a political affiliation or occupation [Vidgen et al., 2021a]. We selected both of these forms of abuse from CAD due to the similarity in their definitions—abuse that is directed at aspects of a person's identity rather than a specific individual directly.

Next, slurs form the second type of hateful content within our dataset, sampled from the Slurs corpus [Kurrek et al., 2020]. Notably, historically derogatory slurs can undergo re-appropriation by specific communities, such as the n-slur in African American Vernacular, transforming them into non-derogatory terms. Therefore, we hypothesize that understanding the contextual nuances of slur use is essential for distinguishing between non-derogatory and derogatory instances.

The last type of hateful content we include is person-directed abuse, hate speech or offensive content that specifically targets and attacks an individual or a group of individuals. We source labelled examples from the Learning to Intervene (LTI) dataset [Qian et al., 2019] to include instances of abuse that require context to discern.

Note that what we consider to be hate speech, broadly speaking, is abusive content targeted either towards a specific person (person directed), identity (identity directed) or affiliation (affiliation directed) based on protected characteristics such as race, ethnicity, sexual orientation or other offensive speech. For completeness, more granular definitions of hate speech used by the sampled datasets are presented in Table 4.1.

For each labelled comment, we retrieved the corresponding complete discussion tree using the Pushshift Reddit API and downloaded all associated images[1]. To refine our dataset, we filtered out conversations without images and constrained comments to a maximum degree of 3 and conversations to a maximum depth of 5. By trimming the discussion tree, we reduce computational complexity and focus on the most relevant parts of the conversation [Parmentier et al., 2021]. We map each retrieved label to Hateful or Neutral and treat the problem as a binary classification. The distribution of each label is shown in Table 4.2.

Most images in HatefulDiscussions, such as the root post, appear in the discussion context rather than directly attached to labelled comments. In our case, only 424 labelled instances have an associated image. Still, all 8000 discussions have an image in the prior context. Therefore, the challenge is to interpret and incorporate multimodal discussion context to disambiguate the meaning of comments that may not contain those modalities.

[1] At the time of writing, Reddit has suspended access to the Pushshift API; however, our dataset remains complete.

Label	Definition
Slurs Corpus [Kurrek et al., 2020]	
Derogatory Slur	"Any [slur] that is understood to convey contempt towards a targeted individual or group"
Not Derogatory Slur	"Meaningful [slur] by targeted or non-targeted groups for an alternate non-derogatory purpose"
Homonym (HOM)	"A slur with one or more non-derogatory alternative meanings"
Learning to Intervene Dataset [Qian et al., 2019]	
Person Directed Neutral	Sampled discussions not classified as "Person Directed Hate"
Person Directed Hate	"We define hate speech as a direct attack on people based on what we call protected characteristics: race, ethnicity, national origin, religious affiliation, sexual orientation, caste, sex, gender, gender identity and serious disease or disability"
Contextual Abuse Dataset [Vidgen et al., 2021a]	
Neutral	"Content which does not contain any abuse. Non-hateful slurs or counter speech and as such would not fall into any of the other categories"
Identity Directed Hate	"Content which contains a negative statement against an identity. An identity is a social category that relates to a fundamental aspect of individuals' community, socio-demographics, position or self-representation. It includes [...] Religion, Race, Ethnicity, Gender, Sexuality, Nationality, Disability and Class"
Affiliation Directed Hate	"Content which express negativity against an affiliation. We define 'affiliation' as a (more or less) voluntary association with a collective. Affiliations include [...]: memberships (e.g. Trade unions), party memberships (e.g. Republicans), political affiliations (e.g. Right-wing people) and occupations (e.g. Doctors)."

Table 4.1: Definitions of Hate Speech Used

Label	Count
Derogatory Slur	4297
Not Derogatory Slur (NDG)	2401
Homonym (HOM)	364
LTI Person Directed Neutral	4116
LTI Person Directed Hate	1313
CAD Affiliation Neutral	4892
CAD Identity Directed Hate	701
CAD Affiliation Directed Hate	275
Neutral	11773
Hateful	6586

Table 4.2: Label Distribution of Hateful Discussions

4.2 Results

4.2.1 Experimental Setup

We conduct a 7-fold stratified cross-validation with a fixed seed (1) and report the average performance for each model. We report overall accuracy and class-weighted Precision, Recall, and F1 to account for label imbalance[2]. Underlined values denote statistical significance using Student's t-test with p-value < 0.05 in all results. Our hyperparameter search space is presented in Table 4.3. All experiments were conducted using 2x A100-40 GB GPUs, a 12-core Intel CPU, 80GB of RAM, and Linux.

4.2.2 Text-only Methods vs. Discussion Transformers

To assess mDT's performance, we compared it against several state-of-the-art hate speech detection methods. For comment-only approaches, we evaluated BERT-HateXplain [Mathew et al., 2021], Detoxify [Hanu and Unitary team, 2020], and RoBertA Dynabench [Vidgen et al., 2021b]. We also compared mDT against a BERT model trained on the HatefulDiscussions training set, referred to as BERT-HatefulDiscuss. To compare against

[2]While in this work our metrics focused on overall performance, we note that quantifying false-positive performance would be a valuable direction for future work, including ROC. We represent mDT's risk of misclassification for hate speech detection with the datasets used in Figure 4.4.

Hyperparameter	**Value**
Pre-Fusion Layers (K)	4 - [0, 2, 4, 6]
Modality Fusion Layer Stack (Z)	2 (8 total) - [1, 2, 4]
Graph Layer Stack (G)	2 (8 total) - [1, 2, 4]
Bottleneck Size (B)	4 - [1, 4, 8, 16, 32]
Max Spatial Attention	5 - [2, 5, 4096 (inf)]
Attention Dropout	0.3 - [0, 0.1, 0.3, 0.5]
Activation Dropout	0.3 - [0, 0.1, 0.3, 0.5]
Graph Dropout	0.4 - [0, 0.2, 0.4, 0.6]
Hidden Dimension (d)	768
Graph Attention Heads	12 - [4, 6, 12, 24]
Modality Attention Heads	12
Optimizer	Adam
Batch Size	48
Epochs	10 w/ Early Stopping
Learning Rate	$3e^{-5} \rightarrow 3e^{-7}$
Learning Rate Scheduler	Polynomial Decay
Warm up Updates	500
Total Updates	3350
Positive Class Weighting	1.5 - [1, 1.5]
Negative Class Weight	1
Freeze Initial Encoders	Yes - [Yes, No]

Table 4.3: mDT Model Hyperparameters. The search space for each parameter is denoted by the values in [...]

Method	Accuracy	Precision	Recall	F1
BERT-HateXplain	0.742	0.763	0.742	0.747
Detoxify	0.687	0.679	0.696	0.677
RoBertA Dynabench	0.811	0.822	0.811	0.814
BERT-HatefulDiscuss	0.858	0.858	0.858	0.858
Graphormer	0.735	0.594	0.759	0.667
mDT (ours)	**0.880**	**0.880**	**0.880**	**0.877**

Table 4.4: Performance of mDT against Text-Only Methods

Bottleneck Size	Accuracy	Precision	Recall	F1
4	**0.880**	**0.880**	**0.880**	**0.877**
8	0.863	0.864	0.863	0.863
16	0.864	0.850	0.853	0.852
32	0.874	0.872	0.874	0.872

Table 4.5: Effect of Bottleneck Size on mDT Performance

previous graph-based approaches, we evaluated the text-only Graphormer model proposed in Section 3.4. Although Graphormer was initially focused on predicting hate speech, we adapted the model for standard hate speech detection and used it to classify comments.

Our results (Table 4.4) show that mDT outperforms all evaluated methods across all metrics. Specifically, mDT achieves 14.5% higher accuracy and 21% higher F1 score than Graphormer. This indicates that our approach to incorporating graph context has significantly improved over the previous approach that did not include this modality. Although the performance gap between BERT-HatefulDiscussions and mDT is narrower, we still outperform all text-only methods. We observed F1-score improvements of 20%, 13%, and 6.3% over Detoxify, BERT-HateXplain, and RoBERTa Dynabench, respectively.

4.2.3 Effect of Multi-Modal Bottleneck Size

Next, we investigated the impact of increasing the number of bottleneck interaction tokens (B) in mDT, added during the modality fusion step. Adding more bottleneck tokens reduces the compression required by BERT and ViT models to exchange information. Table 4.5 presents the results, which show that using four bottleneck tokens yields the best performance. We also observe a slight performance drop when the number of bottleneck to-

Attention Window	Accuracy	Precision	Recall	F1
2	0.866	0.866	0.866	0.866
5	**0.880**	**0.880**	**0.880**	**0.877**
∞	0.870	0.861	0.850	0.855

Table 4.6: Effect of Constraining Graph Attention

Fusion Layers	Accuracy	Precision	Recall	F1
6	0.868	0.856	0.854	0.855
8	0.872	0.871	0.844	0.855
10	0.866	0.867	0.866	0.862
12	**0.880**	**0.880**	**0.880**	**0.877**

Table 4.7: Effect of Fusion Layers

kens exceeds four, underscoring the importance of compression when exchanging modality encodings between models.

4.2.4 Effect of Constrained Graph Attention

In Section 3.5, we explored the limitations of graph transformers for hate speech prediction, finding that discussion context can sometimes mislead graph models into making incorrect predictions. In light of this, we investigate the impact of constraining the attention mechanism of our graph transformer network to only attend to nodes within a specified maximum number of hops from a source node. We report the results in Table 4.6 and find that constraining the attention window to 5 hops achieves better performance. However, we observed that the performance gains from the 5-hop constraint were lost when attention was further restricted to two hops. Our findings suggest that a balance is required when constraining graph attention to achieve optimal performance.

4.2.5 Effect of Fusion Layers

Next, we investigate the effect of increasing the number of Multi-Modal Fusion Layers (Z) in our mDT model. To ensure full utilization of the 16 available layers, any unused layers were allocated to the Initial Pre-Fusion step (K). Our results in Table 4.7 indicate that

Usage of Images	Accuracy	Precision	Recall	F1
With Images	**0.880**	**0.880**	**0.880**	**0.877**
Without Images	0.832	0.835	0.822	0.828

Table 4.8: Effect of Excluding Images

utilizing 12 fusion layers leads to the best performance. Interestingly, the performance gains did not follow a linear trend with the number of fusion layers. Specifically, we observed that using 8 fusion layers outperformed using 10 layers but remained inferior to 12 layers. Further research should investigate the potential benefits of scaling beyond 12 fusion layers by utilizing larger modality models.

4.2.6 Effect of Images

We also investigated the impact of removing images in mDT. Our findings (Table 4.8) support the hypothesis that images provide crucial contextual information for detecting hateful content: excluding images resulted in a 4.8% decrease in accuracy and a 4.9% decrease in the F1 score. It is worth noting that, even without images, mDT outperformed Graphormer (Table 4.4), indicating that our approach yields substantial gains over prior graph-based methods for hate speech detection beyond image inclusion. The results of this experiment underscore the importance of considering multiple modalities for hate speech detection and suggest that future research should further explore improvements by leveraging additional contextual information.

4.2.7 Qualitative Analysis: BERT vs. mDT

We next qualitatively compare a text-only BERT model against the proposed mDT architecture. We find that the text-only BERT model misclassifies 385 of 2717 test instances. Upon passing those test instances through mDT, we found that it corrected BERT's labels in 161 of 385 cases. We further note that BERT and mDT disagree on 264 test instances, of which mDT is correct in 161 (61%). Figure 4.4 shows a fine-grained distribution of misclassified test examples by class. Using mDT results in an overall decrease in misclassifications ($385 \rightarrow 327$), with a significant reduction in false positives (fewer misclassifications for the 'Not Hateful' class). However, we notice that BERT and mDT both struggle to detect the presence of hate speech in derogatory slurs (DEG) and identity-directed (IdentityDirectedAbuse) comments.

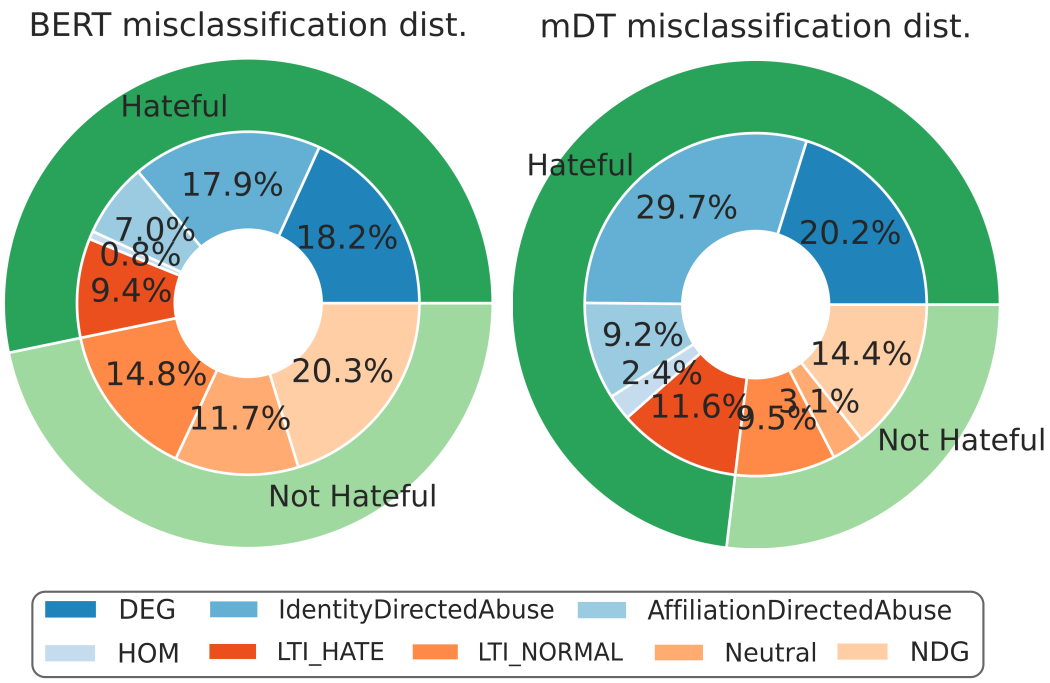


Figure 4.4: Fine-grained distribution of BERT and mDT misclassification. (Acronyms above as in Table 4.2)

Primary Text	**Context (only seen by mDT)**	BERT	mDT
Now imagine if virtuous keyboard sjws had their way? Their mascot should be Ralph Wiggum.	[. . .] Preferred pronouns: go/*f-slur*/yourself [. . .] If the Chinese in my corner of NZ only sold to Chinese they'd starve by Thursday. [. . .] They just wanna *b-slur* about something because their own life sucks.	Not Hateful	Hateful
whoa brah.. leave my tranny out of this	[image] (Figure 4.5) [. . .] that's *f-slur* retarded [. . .] Just spit my drink [. . .]	Not Hateful	Hateful
"That *n-slur* was on PCP Johnson" Lmao	[image] (Figure 4.6) [. . .] Its' a common pattern when dealing with these shootings. * Kill black dude [. . .] * Wingnut welfare kicks in as racist *f-slur* create gofundme of over half a million *f-slur* dollars for cops family [. . .]	Not Hateful	Hateful
uwu owo uwu	[. . .] That is not even close to what feminism is. What you are talking about is radical Feminism [. . .] Got banned from my sexual minority subreddit (r/bisexual) for not believing that all bisexuals should actually be pansexuals [. . .]	Hateful	Not Hateful

Table 4.9: Text instances misclassified by BERT and mDT. Note: The ground truth for all the examples shown here is "Hateful". We have also redacted portions of the text for brevity, indicated by [. . .].



Figure 4.5: An image present in the discussion context of example 3 (Table 4.9), seen only by mDT, contextualizing comments as potentially hateful

Table 4.9 shows some hateful test instances misclassified by the two models. We note that the main text under consideration (an individual comment) may not, on its own, exhibit hate speech; however, considering it in context (the rest of the discussion thread and the image) helps mDT correctly classify the test instances as hate speech. Consider the second example in Table 4.9. The word "tranny" is a common acronym for "transmission" on social media, but considering the context, it is an abusive discussion directed toward the transgender community. This is further contextualized by the accompanying image in the discussion, leading to evidence of hateful interpretations (Figure 4.5). Images within the discussions for each example provide similar contextual evidence, as seen in the third example and Figure 4.6.

We also found some intriguing test examples in which adding context proved misleading to the model, while BERT confidently classified the main text as hateful. For instance, in the last example in Table 4.9, comments in the context are largely non-abusive, misinterpreting the primary text as non-abusive. This suggests that while adding context results in a net decrease in misclassifications, a largely neutral context may also mislead the model. This is likely due to our strong emphasis on discussion context when obtaining the final classification, as reflected in our use of the discussion node embedding b_c^0.



Figure 4.6: An image present in the discussion context of the conversation in table 4.9, seen only by mDT, contextualizing comments as potentially hateful

4.3 Future Work

While we find mDT an effective method for analyzing social media discussions, we have noted that it struggles when the discussion context is predominantly neutral. To address this, we propose using a first-stage text-ranker to compute semantic relevance between comments, thereby filtering out unrelated messages.

We also note that many contextual signals in social media discussions remain untapped beyond text, images, and discussion structure. Incorporating named entity recognition techniques such as Mishra et al. [2022], Hebert et al. [2022b] could integrate deeper analysis of real-world knowledge and be well complemented by the contextual nature of mDT. Another exciting step forward would be to expand our study of individual communities to identify indicators of their propensity toward hateful conduct [Waller and Anderson, 2021]. We are particularly interested in capturing the culture of platforms that host diverse communities, including marginalized ones, which exchange unique, reclaimed vernacular that is at risk of being misinterpreted as hate speech. The contextual nature of mDT, as captured by graph transformers, holds much promise for advancing these extensions.

Another promising direction is the incorporation of generative large language models (LLMs) into hate speech detection. Recently, works such as ShieldGemma [Zeng et al.,

2025] and LlamaGuard [Chi et al., 2024] have been employed to moderate chatbot conversations. A notable key benefit of generative LLM-based hate speech detection systems is the ability to describe and update platform rules in natural language. However, the large parameter sizes of these models (often 8 billion parameters or more, several orders of magnitude larger than mDT) would incur significant latency, which may make them unsuitable for the high-throughput nature of social media conversations. Future work should explore how to bring the adaptability and world knowledge of generative LLMs to mDT without compromising throughput.

Finally, the versatility of mDT's core mechanisms makes it a promising tool for a range of applications beyond hate speech detection. This approach could be applied to other domains such as online product reviews [Jagdale et al., 2019], political discourse analysis [Lyu and Luo, 2022], and popularity analysis [Chen et al., 2022a, Tan et al., 2022], where understanding the discussion context is critical for accurate interpretation.

4.4 Conclusion

This chapter presented mDT: a holistic approach to detecting hate speech in social media. Our model leverages graphs, text and images to reason about entire discussion threads. At the core of our approach is the introduction of hierarchical spatial encodings and the coupling of text, image, and graph transformers through a novel bottleneck mechanism. We also presented a new dataset of complete multi-modal discussions containing a broad spectrum of hateful content, enabling future work into robust graph-based solutions for hate speech detection[3].

One significant contribution is the demonstration that discussion-oriented multimodal analysis can improve the detection of abusive behaviour online. Compared to several key competitors, our experimental results demonstrate the quantitative improvements achieved through our method. Notably, we accomplish a 21% improvement in F1 over previous methods that incorporate discussion context, such as Graphormer (Section 3.4). Furthermore, our initial qualitative analysis demonstrates the valuable impact of our holistic multi-modal approach.

Beyond enhanced holistic discussion analysis, our work provides a richer understanding of conversational dynamics, supporting community-centric prediction. This is primarily

[3]We leave additional comparisons with image-text multi-modal methods to future work to resolve fairness considerations; these competitors require comments to be multi-modal, whereas we can notice relevant discussion responses with images.

driven by our novel improvements to graph transformers, a method gaining momentum in AI molecular modelling, which reveal their potential to capture relationships in complex multimodal discussions. We hypothesize that this expressiveness in capturing context can aid in disambiguating false positives, preventing further marginalization of communities, and proactively mitigating hateful behaviours.

Overall, our approach provides a path forward for addressing hate speech on social media and encourages the exploration of holistic, graph-based, and multimodal models for interpreting online discussions. We believe our research can help foster healthier, more inclusive online environments, thereby improving individuals' mental health.

Part II

Context of Federated Agents

Chapter 5

Background and Roadmap

5.1 Roadmap

In Part II of this thesis, we explore the contextual nature of multi-agent federated reinforcement learning. We start by defining the relevant background in this chapter. Section 5.2 provides the foundational background of reinforcement learning, up to the Soft Actor-Critic algorithm proposed by Haarnoja et al. [2018], which we utilize in this work. Next, Section 5.3 introduces the background of federated learning and methods for merging multiple models in a privacy-preserving manner, typically through parameter averaging. Finally, Section 5.4 covers related work in multi-agent reinforcement learning connected to our proposed methods in Chapter 6.

In Chapter 6, we introduce FedFormer. Our intuition behind FedFormer is that not all agents in federated learning are relevant, and that relevance depends on the environment in which the agent operates. From this perspective, we propose adapting the Transformer architecture by treating the federated learning process as a contextual aggregation problem. Under this framework, we treat the agent's current state as the Query, while the other agents' federated representations of that same state serve as the Keys and Values. The attention mechanism then computes a new state representation as a weighted sum of the other agents' insights, with weights dynamically determined by the relevance of each peer's "Key" to the current agent's "Query".

Within that Chapter, Section 6.1 introduces the FedFormer Architecture. Section 6.2 details our experimental results on the MetaWorld benchmark [Yu et al., 2020], a set of heterogeneous robotic manipulation tasks. Our evaluation explores performance on

individual tasks (Section 6.2.2), the scalability of FedFormer to larger agent pools (Section 6.2.3), and its unique ability to onboard new agents (Section 6.2.4). We conclude with a brief discussion of future directions for FedFormer in Section 6.3.

5.2 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm concerned with how an intelligent agent should take actions in an environment to maximize the cumulative reward. Unlike supervised learning, where a model is trained on a static dataset of labelled examples (e.g., classifying an image as a "cat" or "dog"), an RL agent learns through trial and error. RL agents interact within an environment, observe the consequences of their actions, and adapt their behaviour to maximize some objective. This framework is widely applied in domains such as robotics (learning to walk or grasp objects), game playing (learning to play Chess or Go), and autonomous driving (learning to navigate traffic).

The mathematical foundation of RL is the Markov Decision Process (MDP) [Sutton et al., 1999]. An MDP provides a formal framework for modelling decision-making in situations where outcomes are partly random and partly under the decision-maker's control. An MDP is defined by a tuple (S, A, P, R, γ), consisting of the following components:

- State space S: The set of all possible situations or configurations (states) the environment can be in. For a robotic arm, a state $s \in S$ would represent the angles of its joints and the position of the object it is trying to grasp.
- Action space A: The set of all possible moves or decisions (actions) available to the agent. For the robotic arm, an action $a \in A$ could be the application of a specific voltage to a motor that actuates a joint.
- Transition function P: A probability distribution $P(s'|s, a)$ that defines the probability of reaching state s' when executing action a in state s, modeling the dynamics of the environment. This captures the potentially stochastic nature of the environment, in which actions may not yield deterministic outcomes (e.g., a robot may accidentally slip when taking a step).
- Reward function R: A function $R(s, a)$ that returns the expected immediate reward of executing action a in state s. Rewards signal "good" and "bad" behaviour. For example, the robot arm could receive a reward of 1 for successfully grasping the object, -1 for dropping it, and -0.1 per second to encourage speed. In this thesis, we use the reward as our performance metric.

- Discount factor γ: A scalar value $\gamma \in [0, 1)$ that determines the importance of future rewards. Setting γ close to 0 makes the agent "myopic", caring only about immediate rewards, while setting γ closer to 1 makes the agent "farsighted", valuing long-term cumulative gain.

Given this MDP, the goal of an RL agent is to optimize a policy $\pi_\theta(a|s)$, parameterized by θ, that indicates the expected probability of executing action a in s. The optimization objective is to find an optimal policy π^* that maximizes the expected sum of discounted rewards over time [Sutton et al., 1999]. We can quantify the value of following a policy π starting in state s using the value function $J^\pi(s)$:

$$J^\pi(s_0) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{5.1}$$

Here, the term γ^t ensures that far future rewards are weighted less than immediate rewards, and prevents the sum from growing to infinity in tasks with infinite horizons. An optimal policy yields the highest possible value J^π for every state compared to any other policy (i.e., $\pi^* = argmax_\pi(J^\pi(s))$ for all s).

While $J^\pi(s)$ dictates the value of a specific state, it does not specify which action to take next that would maintain that value or improve upon it. To bridge that gap, we can utilize the action-value function (or Q-function), $Q^{\pi_\theta}(s, a)$. This function estimates the value of taking a specific action a in state s, and then following policy π_θ thereafter:

$$Q^{\pi_\theta}(s_0, a_0) = R(s_0, a_0) + \gamma \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{5.2}$$

Note that the value of a state $J^\pi(s)$ is simply the expected value of the Q-functions for all available actions in that state, weighted by the policy's probability $\pi_\theta(a|s)$ of taking them.

In most real-world scenarios, the agent does not know the transition distribution P or the reward function R in advance. Instead, the agent learns from experience, estimating the optimal policy based on sampled trajectories of states, actions and rewards (i.e., $s_1, a_1, r_1, s_2, a_2, r_2, ...$) that it experiences as it interacts with the environment, referred to as episodes. In practice, this is done by storing experiences into a replay buffer D, storing tuples $(s, a, r, s') \in D$, where s and s' are the current and resulting states, respectively [Sutton et al., 1999]. These tuples are then sampled during training.

Soft Actor-Critic (SAC)

There are many different approaches to find the optimal policy π^*. Of these, Policy Gradient methods update the parameters θ of a policy π_θ by taking steps in the direction of the gradient of the value function, increasing the probability of high-reward actions: [Sutton et al., 1999, Williams, 1992]:

$$\nabla_\theta J^{\pi_\theta}(s_t) = \nabla_\theta log(\pi_\theta(a_t|s_t)) \sum_{t'=t}^{\infty} \gamma^{t'-t} R(s_{t'}, a_{t'}) \tag{5.3}$$

Although stable, Policy Gradient methods often exhibit high variance and sample inefficiency because the sum of an episode can vary significantly across episodes [Konda and Tsitsiklis, 1999]. Conversely, value-based methods, such as Q-Learning [Watkins et al., 1989], first optimize the Q-function (equation 5.2) and then derive a policy π from it. Value-based methods are more sample efficient, but can be unstable during training [Watkins et al., 1989].

Actor-Critic methods aim to combine the strengths of both approaches. Proposed by Konda and Tsitsiklis [1999], Actor-Critic methods maintain two separate neural networks: the Actor network ($\pi_\theta(a|s)$ parameterized by θ), which is the policy that determines the agent's behaviour a in state s, and the Critic network ($Q_\psi(s, a)$ parameterized by ψ) which estimates the expected return (Q-value) of that action. These networks are jointly optimized so that the Critic learns to predict the value of the Actor's actions, and the Actor updates its policy to maximize the Critic's value estimate.

In this thesis, we focus on the Soft Actor-Critic (SAC) variant of the Actor-Critic method, proposed by Haarnoja et al. [2018]. SAC distinguishes itself from standard Actor-Critic methods by modifying the RL objective to maximize both the expected sum of rewards and the policy's entropy. This aims to encourage the agent to explore the environment more widely to avoid converging to a local optimum. This is implemented by modifying the objective function $J^\pi(s_t)$ as

$$J^\pi(s_0) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, s_t) + \alpha H(\pi(\cdot|s_t))\right)\right] \tag{5.4}$$

where H is a function measuring the entropy of policy π in state s_t as:

$$H(\pi(\cdot|s_t)) = \mathbb{E}_{a\sim\pi}\left[-log\ \pi(a|s_t)\right] \tag{5.5}$$

Here, α is a temperature parameter that determines the relative importance of the entropy term (encouraging exploration) against the reward (encouraging exploitation).

Putting everything together, SAC trains the Critic network to minimize the error between its Q-value prediction and a target value:

$$\mathcal{L}_Q(\psi) = \mathbb{E}_{(s,a,r,s')\approx D}\left[\left(Q_\psi(s,a) - y(r,s')\right)^2\right] \tag{5.6}$$

$$y(r,s') = r + \gamma\left(Q_{\hat{\psi}}(s',a') - \alpha\ log\ \pi_\theta(a'|s')\right) \tag{5.7}$$

where y represents the "ground truth" target, composed of the immediate reward r and next state s'. $Q_{\hat{\psi}}$ is the target Q-function, an average of past Q-functions, used to stabilize training.

Conversely, the Actor is trained to maximize the estimates provided by the Critic while maintaining high entropy. This is achieved by minimizing the following loss:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{s\sim D, a\sim\pi_\theta}\left[\alpha\ log\ \pi_\theta(a,s) - Q_\psi(s,a)\right] \tag{5.8}$$

By minimizing this loss, the Actor increases the probability of actions that the Critic assigns high Q-values ($Q_\psi(s,a)$), while the entropy term ($\alpha\ log\ \pi$) encourages exploration.

5.3 Federated Learning

Machine learning often follows a centralized approach, in which diverse data from various sources are collected and merged into a cohesive dataset to train a model. While effective, this paradigm faces challenges when handling sensitive private data requiring security. For example, in healthcare or autonomous driving, raw data such as patient records or camera feeds are too sensitive to transmit to a centralized repository.

Federated Learning (FL) aims to address these challenges. First proposed by McMahan et al. [2017], FL is a distributed training paradigm that enables multiple remote clients to collaboratively train a shared model without sharing raw, potentially sensitive, training data. The key premise of FL is to transmit locally trained model weights rather than data.

There are four key properties of federated learning: distribution, data protection, generality and status equality [Qi et al., 2021]. Distribution states that federated model training is performed in parallel across all participating models. Data protection ensures that the training data held by each participating party is not transmitted to other parties. Generality states that federated models must generalize to diverse environments. Finally, status equality ensures that training favours all participating parties equally and often uses identical infrastructure.

The foundational algorithm for FL is Federated Averaging (FedAvg) [McMahan et al., 2017]. Given a set of N parties, each with loss functions $\{\mathcal{F}_i\}_{i=1}^N$ and datasets $\{D_i\}_{i=1}^N \in D$, FedAvg updates a centralized model as

$$\forall i, w_i(t+1) = w_i(t) - \gamma \nabla F_i(w_i(t)) \tag{5.9}$$

$$w_g(t+1) = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i(t+1) \tag{5.10}$$

where $w_i(t)$ is the model update computed by agent i at time step t, γ is a fixed learning rate and w_g are the parameters of the centralized model. Under this policy, one can tune how often each model communicates its parameters to the centralized source.

While FL was initially designed for supervised tasks, it is natural to extend its principles to Reinforcement Learning. Here, FL is used to federate training of the policy and critic networks [Qi et al., 2021], with various domain-specific modifications. Instead of each agent having a private labelled dataset, each agent processed a local reply buffer D of its own experiences. Applications of Federated Reinforcement Learning include tackling the game Pong [Nadiger et al., 2019]; optimize edge computation for IoT devices [Ren et al., 2019, Wang et al., 2020b, Tehrani et al., 2021], and towards real-time electric vehicle charging [Zhang et al., 2022].

Another direction in federated reinforcement learning is to use agent-level metadata to weight their contributions. Lim et al. [2020] and Chu et al. [2022] weigh agent contributions according to their average reward. Similarly, Wang et al. [2020a] proposes weighting FedAvg based on metadata such as batch size, average loss, dataset size, and hit rate. We refer to this class of federated methods as Weighted Federated Averaging. Since most prior work relies on FedAvg with minor adjustments toward domain-specific goals, or on different model architectures or additional preprocessing steps before applying FedAvg, we focus on directly comparing FedAvg and Weighted FedAvg.

However, a critical limitation of applying FedAvg to RL is that it assumes that, although data may differ, all agents are solving a consistent task that a single average set of parameters can solve [Ren et al., 2019, Wang et al., 2020b, Tehrani et al., 2021]. This does not always hold in FRL, where agents can operate in heterogeneous environments [Qi et al., 2021]. For example, a self-driving car might be operating in a rural environment while another learns in a dense urban center. Averaging the weights of both models might yield a one-size-fits-all model that performs poorly across environments [Qi et al., 2021]. This limitation defines the primary research focus of Part II. Rather than simply averaging the model weights, we argue that aggregation should be contextualized by learning

relationships between agents and that the agent's current environment should determine federation weights.

5.4 Related Work

A similar field to federated reinforcement learning is multi-agent reinforcement learning (MARL). MARL focuses on tasks in which multiple agents interact in the same environment and share insights to solve cooperative or competitive tasks [Stone and Veloso, 2000]. A significant challenge with these systems is the environment's non-stationarity, in which each agent has its own perspective within the same environment, and actions taken by a single agent can affect the environment for others [Zhang et al., 2021]. Therefore, it is critical that multiple agents can efficiently share information, as in federated learning [Foerster et al., 2016, Sukhbaatar et al., 2016, Mao et al., 2018]. However, MARL relies on a continuous flow of observations between agents and often uses a centralized model to plan each agent's actions [Mao et al., 2018, Han and Gmytrasiewicz, 2019], thereby rendering these methods unsuitable for federated learning.

Chapter 6

FedFormer: Contextual Federation with Attention in Reinforcement Learning

Reinforcement learning has become a ubiquitous tool in applications such as autonomous vehicles [Sallab et al., 2017, Shalev-Shwartz et al., 2016], where vehicles learn to navigate complex environments, and Internet-of-Things devices [Mohammadi et al., 2017, Bu and Wang, 2019, Xiong et al., 2020]. However, to obtain robust performance, an extensive collection of training episodes is often needed. In safety-critical systems, such as autonomous vehicles, training episodes must also be diverse enough to account for the numerous edge cases the system may encounter. As a result, the scale and diversity of data collection needed to achieve adequate performance quickly becomes infeasible for a single party. To address these challenges, a common data-collection technique is crowdsourcing, in which training episodes are collected from many agents that explore their local environments and then transmitted to a central controller to train a generalized model [Qi et al., 2021].

However, this strategy comes with a crucial challenge: privacy. In our example of autonomous vehicles, the training data consists of sensitive images of local surroundings that must be transmitted over the internet to a centralized server. This presents an opportunity for malicious actors to intercept this data. Motivated by this challenge, privacy-preserving federated machine learning and federated reinforcement learning have been proposed as solutions. Here, rather than transmitting training observations, local model weights are transmitted instead. A centralized controller then aggregates these model weights to form a new standard model, without any sensitive data being seen.

The current state-of-the-art general method for combining agents in federated reinforcement learning is FedAvg [McMahan et al., 2017]. This strategy calculates the weighted average of transmitted model weights across all agents, with each agent's weight proportional to the number of observations it has seen. However, several drawbacks come with this approach. In reinforcement learning, policy and critic networks are already prone to instability, and averaging their parameters could further exacerbate instability. In addition, averaging model weights across multiple agents often hinders individual exploration in non-identical environments, a challenge when federating models to explore diverse environments [Qi et al., 2021]. An example of this is seen in autonomous vehicles, where knowledge gained from agents exploring rural roads is less helpful to agents exploring urban areas. Despite this, contributions from both sets of agents would still be equally weighted under FedAvg, potentially creating a final agent that performs worse in both environments.

To address these drawbacks, inspired by recent advances in natural language processing, we propose Federated Transformers (FedFormer), a novel federated learning policy based on transformer encoder models [Vaswani et al., 2017, Devlin et al., 2019]. Rather than averaging model weights, we employ transformer encoders to learn contextual relationships among agents. We then leverage these learned relationships to contextually federate agents together during inference. This method enables local agents to maintain models tailored to their specific environment, facilitating local exploration, while also incorporating contextually relevant insights from other agents. Importantly, these relationships can be learned without divulging confidential metadata, such as geolocation data, health information or other sensitive attributes. In addition, we compute these relationships at each time step, enabling dynamic federation as an agent's environment changes throughout the episode. This allows our approach to provide a more efficient aggregation of model insights during inference.

We evaluate FedFormer using the Meta-World environment [Yu et al., 2020], a benchmark of robotic manipulation tasks. Despite training in heterogeneous environments, our approach yields higher episodic returns than FedAvg and non-federated Soft Actor-Critic (SAC) [Haarnoja et al., 2018] single-agent methods. This is despite SAC being trained on the entire set of environments, representing performance without the privacy constraints of the federation. We also demonstrate the onboarding performance of FedFormer, finding that convergence speed increases by a factor of 5 by leveraging pre-trained agents in the federated network, even in previously unseen environments. Notably, these results are obtained without intervention by the other agents, preserving privacy and increasing effectiveness. When scaled to 10 and 15 agents, FedFormer outperforms FedAvg's episodic return by factors of 2.4 and 3.8, respectively, at the last epoch. This illustrates that FedFormer performs well when scaled to larger agent pools (with matching or improved

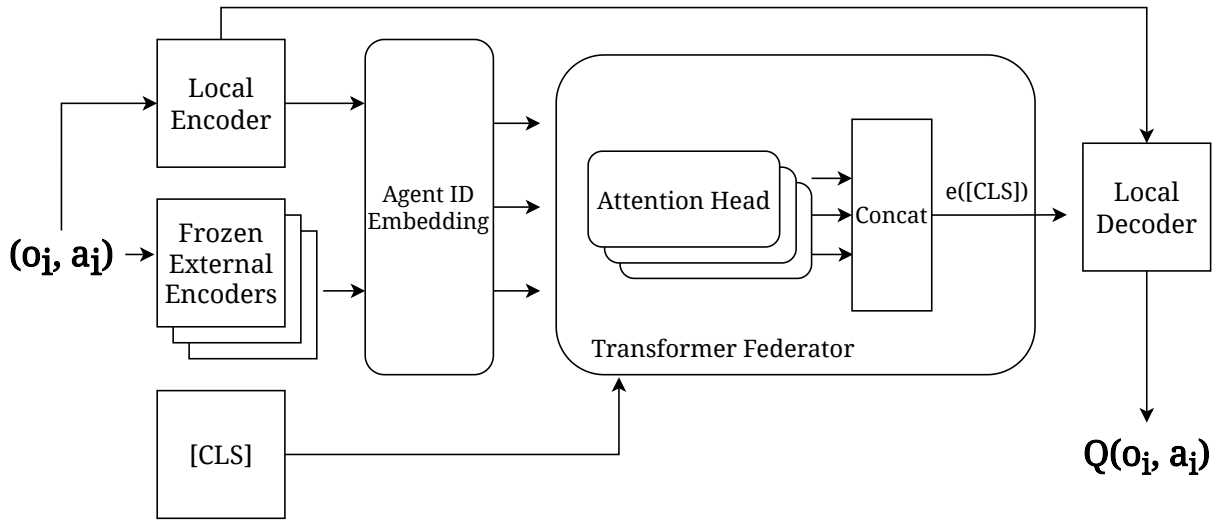


Figure 6.1: Federated Transformer Q-Function (FedFormer). The input to FedFormer is the observation-action pair (o_i, a_i) for a given timestep i, which is passed to all encoders. Following the initial encoding, positional encoding for each agent (Agent ID Embedding) is added, and the embeddings are then aggregated by a Transformer model. The output is the predicted Q-value $Q(o_i, a_i)$.

performance), whereas FedAvg fails to achieve noticeable improvements or shows degraded performance. In all, our results demonstrate that FedFormer outperforms state-of-the-art FedAvg, scales without performance degradation, and is more effective than single-agent Soft Actor-Critic while adhering to the privacy constraints of federated learning.

The code for FedFormer is available at https://github.com/liamhebert/FedFormer.

6.1 Methods

Our FedFormer model adapts the SAC algorithm by introducing a contextual federated Q-function (Section 5.2). We follow the double Q-network formulation of Soft Actor-Critic proposed by Fujimoto et al. [2018] to enhance model stability. That is, each agent has two Q-networks and two target Q-functions, all implemented as FedFormer Q-functions. The actor network remains the same as SAC, formulated as a standard feedforward neural network projecting to the action space.

The overall architecture of our federated Q-function is shown in Figure 6.1. Given a

network of N agents, we initialize the Q-network of each agent with $N-1$ frozen external encoder networks, each representing the other agents, and one local encoder network. We also initialize a local transformer encoder and a final output decoder network. Each encoder network uses the same infrastructure, except that the gradients for each external network are disabled.

The first step in inference is to generate encodings for each encoder network from the same action-observation pair. Following BERT, we first encode agent identity via element-wise addition of learned embeddings, according to the encoder network that generated the representation [Devlin et al., 2019] (IdEmbedding). This enables the aggregator network to identify the source of each encoding, potentially aiding the learning of which agents are more relevant to the current agent. We also include a special `[CLS]` embedding in the input set to encode the global representation of the transformer network. This stage of encodings E is thus given as

$$E = \{FF_i(o,a) + IdEmbedding(i), \forall i \in N, CLSEmbedding\} \tag{6.1}$$

where FF_i is the encoder network for agent i. We then feed our set of encodings, E, into a transformer encoder. Given this set of embeddings, each layer of our transformer encoder network computes:

$$Q = E\mathbf{W_Q},\ K = E\mathbf{W_K},\ V = E\mathbf{W_V}, \tag{6.2}$$

$$Attn(E) = softmax\left(\frac{QE^\top}{\sqrt{d}}\right)V \tag{6.3}$$

where $Attn(E)$ is an attention-weighted representation of E and $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$ are all learnable weight matrices representing the Query, Key and Value embeddings of the input. In our work, we utilize a two-layer transformer encoder architecture, which can be further tuned in future work.

After aggregating each representation through our transformer encoder, we concatenate the transformed encoding of the `[CLS]` token ($e_{[CLS]}$) with the local encoder representation $FF_l((o,a))$ into a final decoder network, predicting the Q-value for that action-observation pair for agent i. This is given as:

$$Q_i(o,a) = FF_d(e(\texttt{[CLS]}), FF_{local}(o,a)) \tag{6.4}$$

Since each agent has its own transformer encoder network that is trained according to the agent's own local loss function, the `[CLS]` encoding is therefore trained to be a global

representation relevant to the task of that agent. During backpropagation, we fine-tune the end-to-end model apart from the external encoder networks.

At the end of every epoch, the frozen external networks are updated with the local encoder networks from the other agents. Since each agent needs only to transmit the weights of its local encoder network to the other agents, network bandwidth increases relative to a FedAvg-based strategy, but not proportionally due to the increase in model complexity. In practice, network bandwidth is kept negligible because local encoder networks are small. In our experiments, each agent was required to upload 1.8MB and download 1.8MB $\times N-1$, which is negligible under modern network conditions. In addition, we only replace the encoders in the Q-function, not the target Q-function, as the target Q-function is eventually replaced by the Q-function regardless, as part of Soft Actor-Critic [Haarnoja et al., 2018].

An interesting aspect of our architecture is the ease of agent onboarding, the process by which agents are introduced into the federation network. Since transformer networks can scale to a variety of input sizes, onboarding new agents simply involves adding a new encoder network to the set of external encoder networks. This is significantly different from FedAvg, which risks introducing untrained noise into other networks through parameter averaging.

6.2 Results

6.2.1 Experimental Design

To evaluate FedFormer, we utilize the Meta-World benchmark [Yu et al., 2020], which comprises a set of robotic manipulation tasks that can be parameterized with random variables. As a result, we can initialize a wide variety of diverse heterogeneous environments while maintaining the same task, a challenge for traditional FedAvg methods. For this work, we evaluate our methods on a set of seven tasks that involve operating a robotic arm to push buttons, open and close windows, doors, and drawers, and reach specific destinations, among others. [1]

We compare our solution against Federated Averaging (FedAvg) [McMahan et al., 2017], Weighted FedAvg (FedWeightedAvg) as implemented in Chu et al. [2022] and non-federated Soft Actor-Critic (SAC) [Haarnoja et al., 2018]. As discussed in Chapter 5,

[1] While the Meta-World benchmark can assess how well a meta-learning agent improves over a collection of tasks, we instead evaluate how well our solution performs on each of the several different tasks individually.

recent advances in federated reinforcement learning are grounded in federated averaging with many domain-specific adjustments. Therefore, we focus our evaluation on comparing underlying federation strategies, as our method can behave as a direct replacement for FedAvg. We also compare our methods against SAC to quantify the performance degradation that occurs when privacy is preserved in a federated setting. Our experiments against SAC are trained on a centralized set of environments used by the federation methods. It is essential to note that performance degradation is expected due to privacy-preserving restrictions. As such, matching or exceeding non-federated methods is a challenging benchmark and is often referred to as an upper limit for federated learning [Qi et al., 2021].

In each experiment, we use the same hyperparameters as proposed for SAC in the Meta-World paper [Yu et al., 2020], except that we train for 250 epochs, each with 600 training iterations, due to computational limitations. We utilize a batch size of 1200 steps and keep a replay buffer of size 10^6. Each agent comprising the evaluated federation strategies learns from a set of five sampled environment configurations and is tested on a different set of five sampled environments. As such, each agent has access to a small, constrained set of heterogeneous environments. Both federation strategies (FedAvg and our FedFormer) have five participating agents unless otherwise stated. For non-federated SAC, the single agent learns from the full set of sampled environments: 25 for training and 25 for testing. To ensure robustness, we use 10 random seeds per method and average the results, with error bars representing the standard error, as shown by the shaded areas in the figures.

We use a three-layer feedforward network with two output heads to predict the mean and variance of a tanh distribution over the action space, serving as the policy network. For FedFormer, our Q-network is a two-layer transformer with a hidden size of 400. Each encoder and decoder network consists of a three-layer feed-forward network, with each layer having a hidden size of 400, except for the final decoder layer, which has a single output unit representing the Q-value. For FedAvg and SAC, the Q-network consists of a three-layer feedforward network, with each hidden layer having 400 units, except for the final layer, which has a single output unit representing the Q-value. We include a detailed list of all hyperparameters in Table 6.1.

To implement our work, we modified an SAC implementation in RLKit [RAIL-Berkeley and contributors, 2021] by adding additional modules borrowed from Garage [Contributors, 2019] and PyTorch [Paszke et al., 2019]. RLKit and Garage are licensed under the MIT license[2], and Pytorch is licensed under the BSD-3 "Revised" license [3]. Experiments were run in parallel on a computing cluster comprising RTX 3080, RTX 2080Ti, V100, and P100

[2]https://spdx.org/licenses/MIT.html
[3]https://github.com/pytorch/pytorch/blob/master/LICENSE

Parameter	Value
Shared Parameters	
Batch Size	1200
Number of Epochs	250
Path Length	500
Gradient Steps per Epoch	600
Discount Factor	0.99
Optimizer	Adam
Policy Hidden Sizes	(256, 256, 256)
Policy Activation Function	ReLU
Policy Learning Rate	3×10^{-4}
Policy Minimum Standard Deviation	e^{-20}
Policy Maximum Standard Deviation	e^{2}
Soft Target Interpolation	5×10^{-3}
Use Automatic Entropy Tuning	True
FedFormer	
Transformer Layers	2
Transformer Heads	4
Transformer Hidden Sizes	256
Encoder Hidden Size	(256, 256, 256)
Decoder Hidden Size	(256, 256, 256)
Batch Normalization	True
Activation Function	ReLU
Q-Function Learning Rate	3×10^{-4}
SAC and FedAvg	
Hidden Size	(256, 256, 256)
Activation Function	ReLU
Q-Function Learning Rate	3×10^{-4}

Table 6.1: Hyperparameters used by each method explored in our study. Shared parameters apply to all methods.

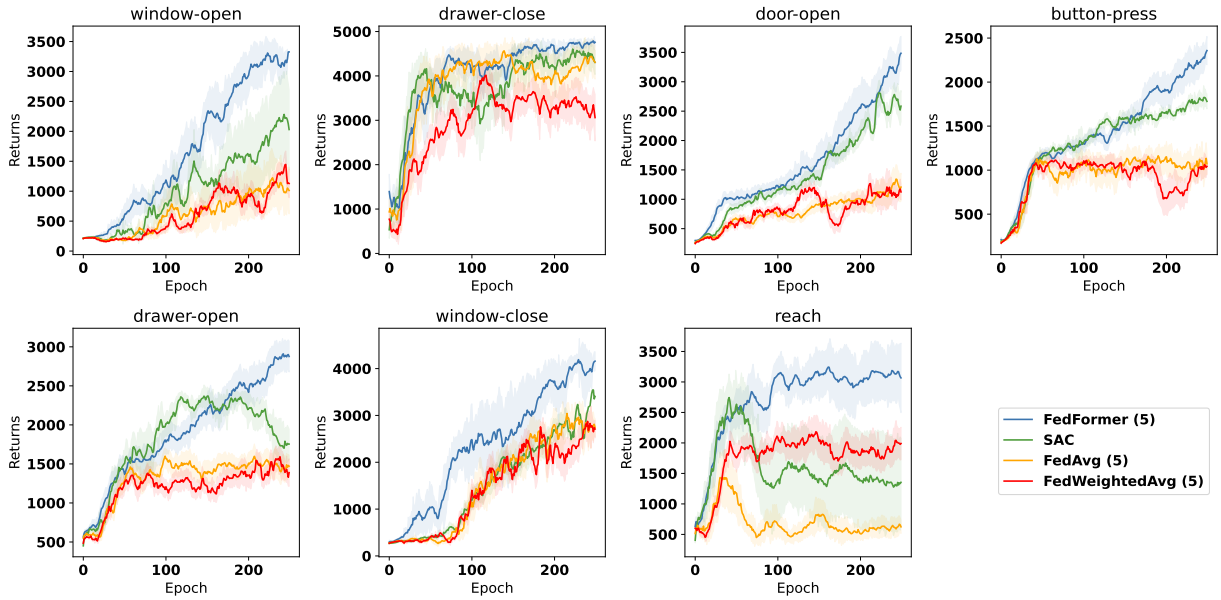


Figure 6.2: Performance on Meta-World tasks

GPUs, depending on availability.

6.2.2 Meta-World Evaluation

In the first set of experiments, we compare the performance of FedFormer (blue) against FedAvg [McMahan et al., 2017] (orange), Weighted FedAvg [Chu et al., 2022] (red) and non-Federated SAC [Haarnoja et al., 2018] (green) on seven Meta-World tasks. Figure 6.2 shows the results, where the X-axis represents training epochs and the Y-axis is the average episodic return. Improvements in the X-axis indicate that the agent can learn faster, and the Y-axis indicates enhanced performance, where the agent outperforms other agents. Further, the separation in standard error (shaded area) between the methods indicates improved performance across 10 evaluated seeds.

FedFormer outperforms FedAvg and Weighted FedAvg and, surprisingly, matches or exceeds non-federated SAC performance across all evaluated tasks. As indicated by the standard error bars, FedFormer consistently outperforms FedAvg across six of the seven tasks evaluated. The only exception is the `drawer-close` task, in which all methods achieved similar returns.

The performance difference between FedFormer and FedAvg increases as training progresses. At the final epoch, FedFormer achieves 2.17x (`door-open`), 1.77x (`drawer-open`), 1.48x (`window-close`), 2.25x (`button-press`), 3.41x (`reach`) and 2.60x (`window-open`) performance gains over FedAvg. The only task on which FedAvg performs comparably to FedFormer is `drawer-close`. We also observe that FedAvg behaves similarly to FedWeightedAvg, except for `window-close`, where FedWeightedAvg achieves significantly stronger performance.

Notably, increases in effectiveness are observed in the `button-press`, `drawer-open`, and `door-open` tasks, in which FedAvg appears to plateau around epoch 50. This is contrasted with FedFormer, which continues to improve performance. We hypothesize that this is due to stifled exploration in FedAvg, a key feature of reinforcement learning that differs from traditional machine learning. When an individual agent achieves substantial gains in FedAvg, it is forced to average its parameters with those of other agents, which may be performing much worse. This is overcome by FedFormer, which enables attentive, selective aggregation among agents based on their learned relevance. The most surprising results are compared with those of the non-federated SAC. In the `door-open`, `reach`, `window-open`, and `window-close` tasks, FedFormer consistently outperforms SAC at each epoch and is robust to standard error. Focusing on the final epoch, we also obtain 1.56x performance gains in the `drawer-open` task, 1.36x performance gains in the `button-press` task, and 1.58x performance gains in the `window-open` task. We achieve performance comparable to SAC on the remaining tasks.

Recall that SAC agents are centralized and train on the entire set of environments without preserving privacy. In contrast, FedFormer adheres to the privacy constraints of federated learning by utilizing a distributed pool of agents. As such, we expected that SAC would serve as the upper bound of performance achievable by FedFormer. We reason two hypotheses to explain FedFormer's superior performance. First, the use of a transformer aggregation network enables FedFormer to leverage agents' individual expertise in a dynamic, context-sensitive manner without requiring a centralized model. As such, FedFormer resembles a mixture-of-experts architecture [Fedus et al., 2022], enabling it to leverage the expertise of distinct experts. Second, the federated nature of FedFormer enables distinct replay buffers for each agent, whereas SAC uses a single central buffer. This may assist in developing specialist behaviours for each agent.

6.2.3 Scalability

Next, we assess the scalability of our method to larger agent sets. To do this, we compare the results of both federation strategies (FedAvg and our FedFormer) when scaled to 5,

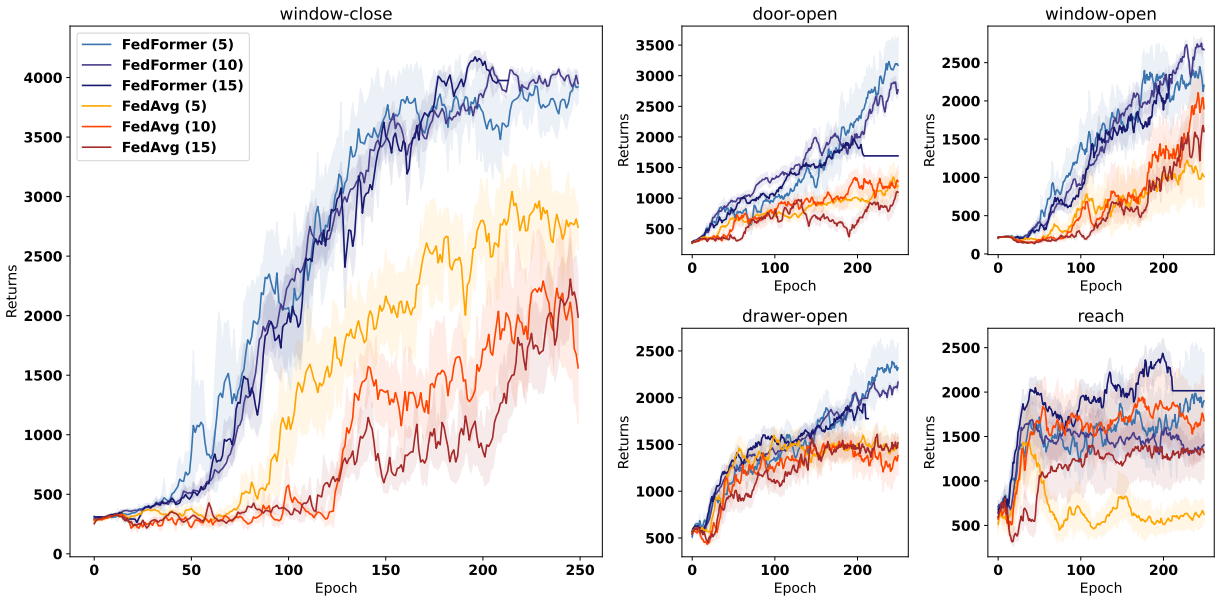


Figure 6.3: Performance when scaling federation methods to 5, 10 and 15 agents

10 and 15 agents (Figure 6.3) [4]. As in our tests in Section 6.2.2, each agent is given five sampled environments for training and five for testing, without replacement. Since Meta-World generates 50 environments per task, some agents will have overlapping training and testing tasks, but never within the same agent. It is assumed that including more agents in federated learning will yield more robust models by providing greater access to training data. However, it is also important to note that the reported performance is the average of *all* agents. Therefore, all agents must perform competently to achieve a high metric in our results. It is essential to note that scaling poses a more challenging problem due to the increased heterogeneity of the environments in which each agent operates.

We find that increasing the number of agents improves performance in FedFormer and worsens performance in FedAvg. Focusing on the final epoch and when utilizing 10 agents, we can see performance improvements of 2.23x (`window-close`), 2.95x (`door-open`), 1.47x (`drawer-open`), 1.83x (`reach`) and 2.70x (`window-open`) compared to FedAvg. We observe similar gains with 15 agents, with performance improvements of 1.98x (`window-close`), 2.75x (`door-open`), 1.62x (`drawer-open`), and 1.88x (`window-open`). We also see that FedAvg fails to meaningfully scale above FedAvg (5), with FedAvg (5) and FedAvg (10)

[4] Although we evaluate our methods up to 15 agents, we can scale up to 512 agents (or 1024 with sparse variants) due to our usage of transformers

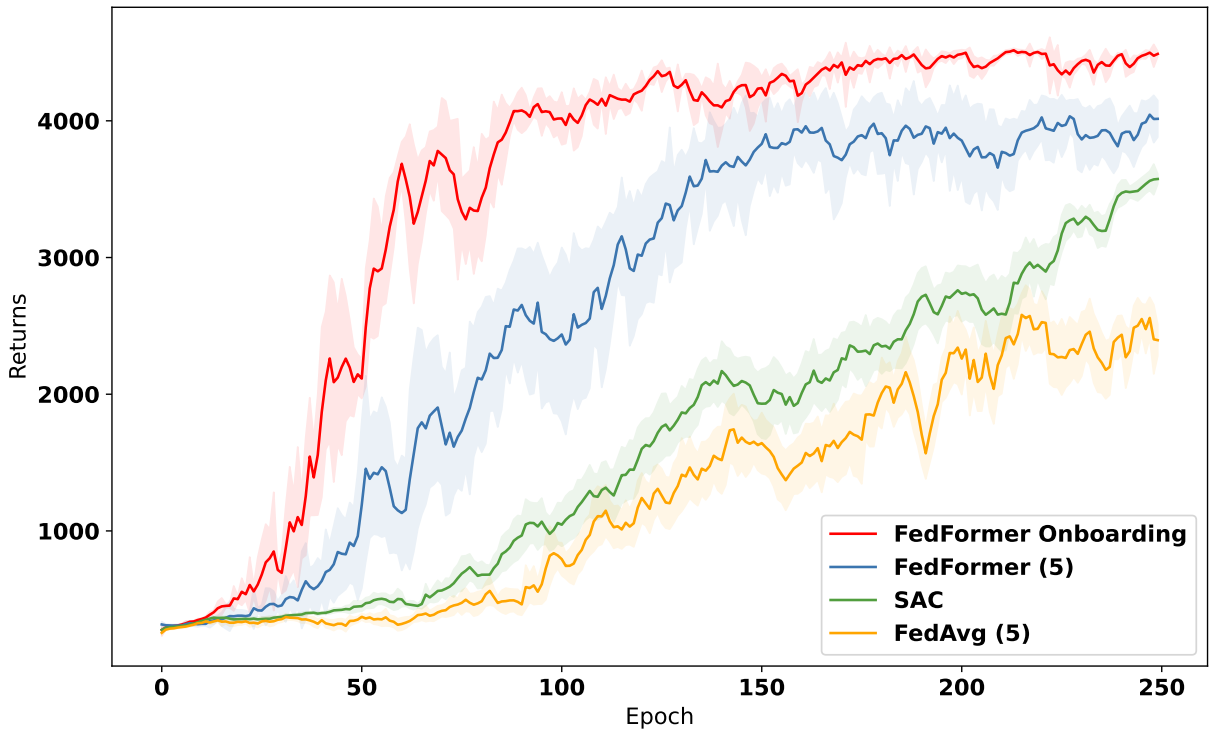


Figure 6.4: Performance of onboarding agents into pre-trained federation networks on the `window-close` task

performing comparably or worse across all tasks. Many of the reported results are also robust to standard error. We hypothesize that FedAvg's poor performance stems from many agents simultaneously exploring the environment, only to be forced into a single representation. In our experiments, we observed that FedWeightedAvg exhibited trends similar to those of FedAvg, with marginal improvements as the number of agents increased.

As a result, we conclude that our federation strategy can efficiently scale to include multiple agents. This result aligns with prior work on transformer models, which has shown their ability to reason over larger sets of embeddings [Devlin et al., 2019].

6.2.4 Agent Onboarding

In real-life use cases, agents are often introduced at random intervals into federation strategies. When an agent joins an established federation network, the network must adapt to

accommodate the new environments introduced by the agent. In the case of FedAvg, this can be a significant setback, as model averaging can introduce noise into converged models when new models originate from distinct heterogeneous environments. In addition, FedAvg can require all agents to restart training to keep up with the new agent's modifications.

We now evaluate the onboarding strategy of FedFormer, which overcomes all these challenges (Section 6.1). To do this, we train a FedFormer network on the `window-close` task and save the resulting converged encoder networks. We then train a single new agent across five never-before-seen environments, using the saved encoder networks from previous agents as the external network set. The main difference here is that the federation does not update the external networks throughout training. In our experiments, we evaluate our methods using the trained networks obtained from each seed of the `window-close` task. The results of this experiment, compared with FedFormer, SAC, and FedAvg, are shown in Figure 6.4.

We see that the onboarded agent (red) drastically benefits from the pre-trained encoder networks. Within the first 60 epochs, the new agent achieves near-equivalent performance to the original FedFormer network after 250 epochs of training, resulting in a $\approx$ 5x efficiency increase. We also achieve 6.5x gains over SAC and 11.23x gains over FedAvg at epoch 60, demonstrating faster convergence. In addition, performance continues to improve beyond this point, obtaining a 1.24x increase over FedFormer at epoch 250. It is important to note that this onboarding process was done without any further training from the other agents.

6.2.5 Ablation Study

To evaluate the utility of attention in our federated network, we conducted additional experiments in which we replaced the Transformer in FedFormer with a three-layer multi-layer perceptron (MLP), which we title FedMLP. To do this, we concatenate the embeddings generated by each encoder network and feed them to the MLP. Then, the MLP's output layer produces a 256-dimensional embedding, matching the dimensionality of the CLS embeddings produced by FedFormer. Just as in FedFormer, this embedding is then decoded by a local decoder network to generate the Q value for that state-action pair (Figure 8.1). This baseline is similar to work by Zhuo et al. [2019], which proposed concatenating the Q-network embeddings of two agents into a global MLP. Our results when scaling this method to 5, 10, and 15 agents are shown in Figure 6.5, following the procedure outlined in Section 6.2.3. It is important to note that scaling poses a more difficult problem due to the increased heterogeneity of the environments to which each agent is exposed.

Across all evaluated tasks, we observe FedFormer outperforming the FedMLP baseline.

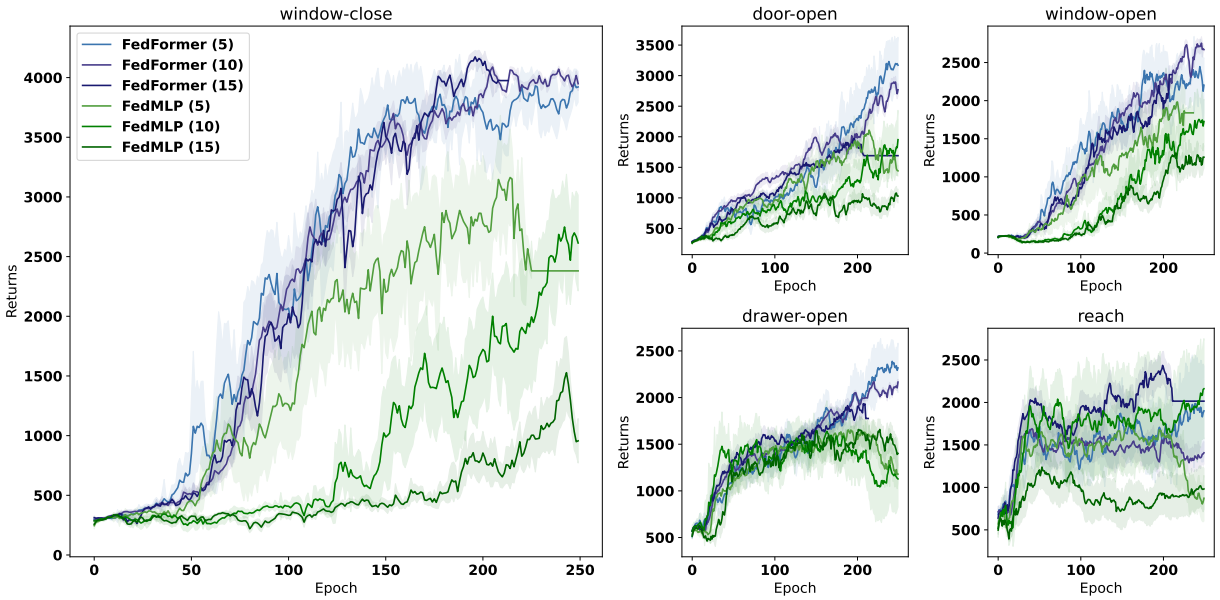


Figure 6.5: Scaling Performance when replacing Transformer Network with MLP Network

This can be most evidently seen in the `window-close` task, where FedFormer (5) often achieves over 2x the performance of FedMLP (5) across each epoch. It is essential to note the significant differences between the two methods when scaled to include larger agent pools. The most pronounced difference is observed on the `window-close` task, where the performance of FedMLP drops by 2.4x when scaled to 10 agents and by 3.8x when scaled to 15 agents. This contrasts with FedFormer, where performance either remains the same or increases with increasing agent count. These results indicate the importance of the attention mechanism in FedFormer, especially for scalability. With attention, it is possible to contextually select and aggregate insights from participating agents in relation to the current agent's needs. This contrasts with FedMLP, which must aggregate insights from all agents, including those that provide irrelevant inputs. The problem of noisy inputs is then further exacerbated with increased agent counts, as seen in Figure 6.5.

6.3 Conclusion

We presented a novel transformer-based federation strategy, FedFormer. Our technique addresses concerns about using federated learning for reinforcement learning, which is

typically implemented using the FedAvg parameter averaging method. Instead, our method uses attention to compute contextual relationships between agents without compromising individual agents' ability to explore the environment.

Our method outperforms FedAvg and single-agent Soft Actor-Critic (SAC), with FedFormer outperforming FedAvg by a factor of 4.4 and SAC by a factor of 1.75. In addition, we demonstrate that FedFormer scales to larger agent counts, achieving gains of 2.4x and 3.8x over FedAvg with 10 and 15 agents, respectively. This contrasts with FedAvg, which has negligible performance gains and, at times, performs worse. Lastly, we demonstrate the ease of agent onboarding in our methods, achieving 5x faster convergence by leveraging pre-trained models without sacrificing the performance of other models. All of this is achieved while adhering to the privacy constraints of federated learning.

The primary limitation of our work is related to computational latency. Previous work in transformer models states a computational limit of ≈ 512 embeddings when computing attention [Devlin et al., 2019]. This is due to the quadratic increase in latency with the number of embeddings. Further, each new agent will add an encoder network to the model. As a result, this limits the number of agents that can participate in the federation network. However, this limit can be addressed by using domain-specific agent sampling strategies, as explored by Ren et al. [2019], Wang et al. [2020b], and Tehrani et al. [2021].

With our advances in FedFormer, several directions for future work are possible. First, since our approach does not rely on averaging parameters, it would be interesting to explore the use of deep learning models to represent the encoder networks [Mnih et al., 2013]. This could allow the federation to tackle more complex tasks involving computer vision and partial observability by leveraging deep encoders, such as convolutional and recurrent neural networks. The use of these models would have been difficult in the past due to instability caused by parameter averaging [Qi et al., 2021].

Another interesting direction for future work is to explore related work in supervised heterogeneous federated learning. This area proposes techniques for grouping and filtering agents, including methods for addressing fault tolerance in Byzantine agents [Gao et al., 2022]. These techniques support learning across a set of heterogeneous, changing environments but often assume that the environments (datasets) of individual agents remain consistent [Gao et al., 2022]. Although this is not typically the case in federated reinforcement learning, these techniques still offer valuable insights. For example, Huang et al. [2022] could be adapted to cluster agents based on their Q-values, and Fang and Ye [2022] could be repurposed to detect byzantine agents. The work of Fan et al. [2021] and Chen et al. [2022b] will also become relevant. The Byzantine filtering layers proposed by Fan et al. [2021] can be applied to input encoder networks; insights from Chen et al. [2022b] on

theoretical guarantees may motivate us to extend beyond our current experimental validation. Since both Fan et al. [2021] and Chen et al. [2022b] are grounded in FedAvg, using FedFormer with these approaches may represent a significant step forward.

Lastly, our work opens the door for advances in existing FedAvg-based architectures. From IoT devices to self-driving cars, we believe our methods can serve as a direct replacement for FedAvg, delivering better performance. Further evaluation of these tasks with the addition of domain-specific adjustments such as those proposed in previous work [Ren et al., 2019, Wang et al., 2020b, Tehrani et al., 2021, Nadiger et al., 2019] would be especially interesting. We hope that our work can be used to improve federated reinforcement learning and to introduce advances in deep learning to this critical field, especially as privacy becomes increasingly difficult to preserve in favour of performance.

Part III

Context of Sequential Recommenders

Chapter 7

Background and Roadmap

7.1 Roadmap

In Part III of this thesis, we investigate the role of context in personalized sequential recommender systems. Specifically, we explore how to bridge the gap between two distinct but complementary sources of context: the implicit, macro-level signals of community behaviour (collaborative filtering) and the explicit, semantic signals of item content and user intent.

We start by defining the relevant background in this chapter. Section 7.2 establishes the foundational paradigms of collaborative and content-based filtering, which respectively leverage community homophily and item attributes to determine relevance. Next, Section 7.3 discusses the evolution of these static methods into sequential recommender systems, which model the temporal dynamics of user interactions, a critical form of context. We then examine the recent trend towards text-only architectures in Section 7.4, where Large Language Models (LLMs) replace traditional collaborative filtering embeddings entirely by representing items as text. Finally, Section 7.5 reviews interactive recommender systems that aim to incorporate user intent through a process called "critiquing", setting the stage for our work on incorporating explicit user intent.

In Chapter 8, we propose Flare (Fusing Language models and collaborative Architectures for Recommender Enhancement), a hybrid sequential recommender that combines community behaviour-based embeddings (collaborative filtering) with content-based information, such as item descriptions, and user intent information. Our solution is uniquely enabled by modifications to the Perceiver architecture [Jaegle et al., 2021], which allow

Flare to condense all these signals into learnable latent representations (introduced in Section 8.1.2).

Within that Chapter, Section 8.1.2 introduces the Flare architecture, Section 8.1.3 introduces the auxiliary contrastive loss used by Flare, and Section 8.1.4 introduces how Flare incorporates user intent. Section 8.2 presents our experimental results, addressing four key questions: the scalability of collaborative filtering baselines (Section 8.2.2), the performance of Flare on large-scale datasets (Section 8.2.3), Flare's ability to follow textual user intent (Section 8.2.4), and Flare's capacity to generalize to out-of-distribution critiques (Section 8.2.5). To evaluate the capability of our models to follow intent, we propose the Cat-nDCG evaluation paradigm in Section 8.2.5. We further validate our approach with a qualitative study in Section 8.2.6 and extensive ablation studies in Section 8.2.7, before summarizing our contributions in Section 8.3.

7.2 Collaborative and Content-Based Filtering

At its theoretical core, the recommendation problem can be formalized as a utility estimation task. Let $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ denote a set of M users and $\mathcal{I} = \{i_1, i_2, ..., i_N\}$ denote the set of N items. The fundamental goal of a recommender system is to learn a utility function $f : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$, where the $r_{u,i} \in \mathbb{R}$ output represents the predicted preference, relevance or probability of interaction between a user and an item. Then, the item recommended to the user will be the item with the optimal utility value (i.e. $i_u^* = argmax_{i \in \mathcal{I}}\ f(u, i)$). The central challenge of this task arises from the sparsity of the interaction space. The Cartesian product $\mathcal{U} \times \mathcal{I}$ is vast, yet the set of observed interactions $\mathcal{O} = \{(u, i) | r_{u,i} \neq 0\}$ is much smaller relative to the set of total possible interactions. This challenge has motivated efforts to develop a more expressive, contextually grounded understanding of users $\mathcal{U}$ and $\mathcal{I}$ to estimate relevance more accurately.

Before the advent of deep learning and sequential modelling, the field of recommender systems was dominated by two paradigms: collaborative filtering (CF) and content-based filtering (CBF). Both approaches aim to define the relevance function f by measuring the pairwise similarity between a user u and an item i, with CF using community homophily-based similarity and CBF using semantic-based similarity. Understanding these foundational paradigms is key, as modern hybrid recommenders are essentially sophisticated syntheses of the foundational ideas of these two paradigms.

7.2.1 Collaborative Filtering

Collaborative filtering (CF) operates on the principle of homophily and collective intelligence. The central hypothesis of CF is that users who have agreed in the past (i.e., shared similar interaction patterns) will agree in the future. For example, if user A has purchased items 1, 2, and 3, and user B has purchased items 1 and 2, a collaborative filtering approach would recommend item 3 to user B. A defining characteristic of pure CF is its agnosticism towards the attributes of the items themselves. It does not require any external context about the items (such as genre, price or textual description). It relies solely on the structure of the user-item interaction matrix $\mathcal{O}$.

CF strategies are broadly categorized into heuristic-based and model-based approaches. Heuristic-based approaches compute similarity metrics directly on the interaction data without learning an underlying generative model. The first step in a heuristic model is to identify a neighbourhood of users, $\mathcal{N}(u)$, most similar to the target user u. This is often created using metrics such as the Pearson Correlation Coefficient or the cosine similarity of item ratings (i.e., Likert $[0-5]$ ratings of preference) between the target user and other users. Then, using this neighbourhood, the predicted rating $\hat{r}_{u,i}$ for an item i is computed as an aggregation of the ratings to the same item by neighbours $v \in \mathcal{N}(u)$. While heuristic approaches are largely intuitive, they are notably sensitive to data sparsity because they rely on ratings from a neighbour set $\mathcal{N}(u)$.

Model-based approaches, in contrast, aim to learn the underlying structural properties of user-item observations $\mathcal{O}$. A seminal model-based CF approach is Matrix Factorization (MF), a variant of Singular Value Decomposition (SVD). Matrix Factorization addresses sparsity by learning projections of users U and items I into a shared, lower-dimensional feature space of dimension d, where $d \ll M, N$. Each user $u \in U$ and item $i \in I$ is associated with a latent vector $p_u \in \mathbb{R}^d$ and $q_i \in \mathbb{R}^d$, respectively. Then, the estimated relevance $\hat{r}_{u,i}$ is approximated by computing the dot product of latent vectors:

$$\hat{r}_{u,i} = q_i^T p_u \tag{7.1}$$

To learn the latent vectors p_u and q_i, the system minimizes the regularized squared error on the set of ground truth observed ratings $\mathcal{O}$:

$$min_{q*,p*} \sum_{(u,i)\in\mathcal{O}} (r_{u,i} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \tag{7.2}$$

where $r_{u,i}$ is the ground truth relevance value and λ is a hyper-parameter controlling the regularization strength. Since the optimization objective of matrix factorization (equation

7.2) is non-convex due to having two unknowns (p_u and q_i), alternating least squares (ALS) is often applied to update parameter weights [Hu et al., 2008]. At a high level, ALS alternates between solving q_i and fixing p_u, then solving p_u and fixing q_i. This alternating process ensures that equation 7.2 decreases until convergence [Hu et al., 2008].

The key strength of matrix factorization lies in its ability to capture transitive relationships. If user A and user B share similar latent embeddings (p_u) and user B rates item X highly, MF can recommend item X to user A even if they share no concrete overlap in viewed items, effectively handling sparsity better than neighbourhood methods. Many works have proposed augmentations to matrix factorization. Koren [2009] incorporates global user biases b_u (i.e. the average rating the user gives to items) and global item biases b_i (i.e. the average rating of the item) when calculating $\hat{r}_{u,i}$. These factors can then be updated over time to reflect evolving user trends, such as fashion trends. He et al. [2017] propose replacing the relevance dot product (Equation 7.1) with a neural network, termed Neural Collaborative Filtering.

7.2.2 Content-Based Filtering

In parallel to collaborative approaches, content-based filtering (CBF) constructs recommendations by comparing items' attributes with a user's profile [Gao et al., 2021]. Unlike CF, which relies on community behaviour, CBF relies on the semantic content of items to determine relevance [1]. For example, if user A rated the movie *Alien* highly, a content-based recommender could retrieve *Terminator*, as the two movies share overlapping characteristics (i.e thriller, action, sci-fi).

A typical workflow for a CBF system involves first creating semantic representations q_i for all items in the corpus, such as TF-IDF embeddings of product descriptions and image embeddings. Then, user representations p_u are created as the average of item embeddings q_i that the user has interacted with. Like CF, the estimated relevance $\hat{r}_{u,i}$ is determined by measuring the embedding similarity between q_i and p_i (equation 7.1).

While CBF cannot capture transitive community relationships, it excels in addressing the cold-start problem. The cold-start problem occurs when the system lacks sufficient data to model a user or item reliably, typically due to limited observations (i.e., ratings) $\mathcal{O}$ [Bobadilla et al., 2012, Hou et al., 2018]. The most common cause of the cold-start problem is the introduction of new items or users, which results in less robust q_i and p_u

[1] In this thesis, we focus on semantics defined in natural language; however, previous work has also recognized content-based semantics as defined in knowledge base properties [Tran and Cohen, 2000]

representations, respectively. Another symptom of the cold-start problem is popularity bias, in which unpopular items are unlikely to receive additional ratings while popular items continue to receive ratings [Hou et al., 2018]. CBF is robust to the *item* cold-start problem as it does not require an item to be rated by others to be recommended. However, CBF is also prone to overspecialization, leading to recommendations that are overly similar to previous interactions. For example, a user who interacts with a red shirt might only be recommended more red shirts.

The limitations of CF and CBF, specifically their inability to model the sequence of user actions and their struggle with cold start items, motivated the development of sequential recommender systems and hybrid architectures.

7.3 Sequential Recommender Systems

Traditional collaborative filtering and content-based filtering methods are often static. User preferences are modelled by aggregating the entire interaction history into a single snapshot representation (e.g. a latent vector p_u). This aggregation process effectively discards the temporal order of interactions.

However, in many real-world scenarios, user behaviour is intrinsically sequential and causal. A user's next action is often driven by their immediate context rather than their overall long-term preferences. For instance, a user who purchases a digital camera is highly likely to subsequently purchase an SD card or a camera bag. This pattern is not a correlation but is grounded in a temporal context. Static models might recommend another camera (because it is similar to the purchased item), whereas a sequential model recognizes the complementary nature of the accessory following the main purchase. Sequential recommender systems (SRS) aim to capture these dynamic and temporal dependencies [Donkers et al., 2017, Hidasi et al., 2016a, Kang and McAuley, 2018, Sun et al., 2019]. Instead of a static user vector p_u, the input to an SRS is the ordered sequence of user interactions $S_u = \{i_1, i_2, ..., i_t\}$, where the objective is to predict the next item i_{t+1}.

Work on sequence modelling for recommendations has traditionally focused on RNNs [Hidasi et al., 2016a, Donkers et al., 2017] (introduced in Section 2.1) and, more recently, on transformers (introduced in Section 2.3). Two dominant transformer-based architectures have emerged as state-of-the-art in collaborative filtering sequential recommender systems: SASRec [Kang and McAuley, 2018] and Bert4Rec [Sun et al., 2019]. In both of these architectures, the input to the transformer is the sequence S_u consisting of dense embeddings of items (akin to q_i), in place of word embeddings.

The difference between SASRec and Bert4Rec lies in their attention patterns and loss functions. SASRec [Kang and McAuley, 2018] enforces a causal (unidirectional) attention mask (akin to a generative model), whereas Bert4Rec [Sun et al., 2019] adopts a bidirectional attention mask. The authors of SASRec [Kang and McAuley, 2018] argue that the causal mask of SASRec enables a more natural training process, where predictions are strictly based on previous items. Conversely, the authors of Bert4Rec [Sun et al., 2019] argue that bidirectional masking increases the amount of information available for creating rich item representations. The other difference between SASRec and Bert4Rec lies in their loss function. Standard SASRec typically uses a binary cross-entropy loss with negative sampling, where the model is tasked with predicting whether an item in the sequence is correct (1) or incorrect (0) [Kang and McAuley, 2018]. Conversely, Bert4Rec uses standard cross-entropy (softmax) over the entire item vocabulary, tasking the model to select the next item from the complete item vocabulary [Sun et al., 2019].

To improve recommendation quality, follow-up efforts investigate hybrid models that augment item ID sequences with corresponding item context sequences (e.g., textual descriptions, categories, attributes). For example, HybridBERT4Rec [Channarong et al., 2022] extends Bert4Rec with a separate tower that models item context, using the set of users that consume that item as input. S^3Rec [Zhou et al., 2020] proposes self-supervised pre-training tasks to align items and attributes by utilizing mutual information maximization. CaFe [Li et al., 2022] explicitly models coarse-grained, or intent-based, sequences along with dense fine-grained sequences to better capture short-term vs long-term preferences. CARCA [Rashed et al., 2022] combines item sequences, associated context, and item attributes, connected via cross-attention, to better correlate older and more recent items in the user profile. U-BERT [Qiu et al., 2021] trains user and content encoders in a self-supervised pre-training stage using content-rich signals, followed by fine-tuning to focus on content-deficient domains.

7.4 Text-only Large Language Model Recommenders

Large Language Models (LLMs) have emerged recently as a promising avenue for recommender systems. LLMs are transformer models pre-trained on large-scale text corpora, enabling them to acquire world knowledge, develop reasoning capabilities, and enhance natural language understanding. Notably, LLMs have also demonstrated the ability to generalize across many tasks by framing them as text-generation tasks [Raffel et al., 2020]. These capabilities have encouraged their adoption in recommender systems, where works replace explicit collaborative-filtering-based item representations with natural-language

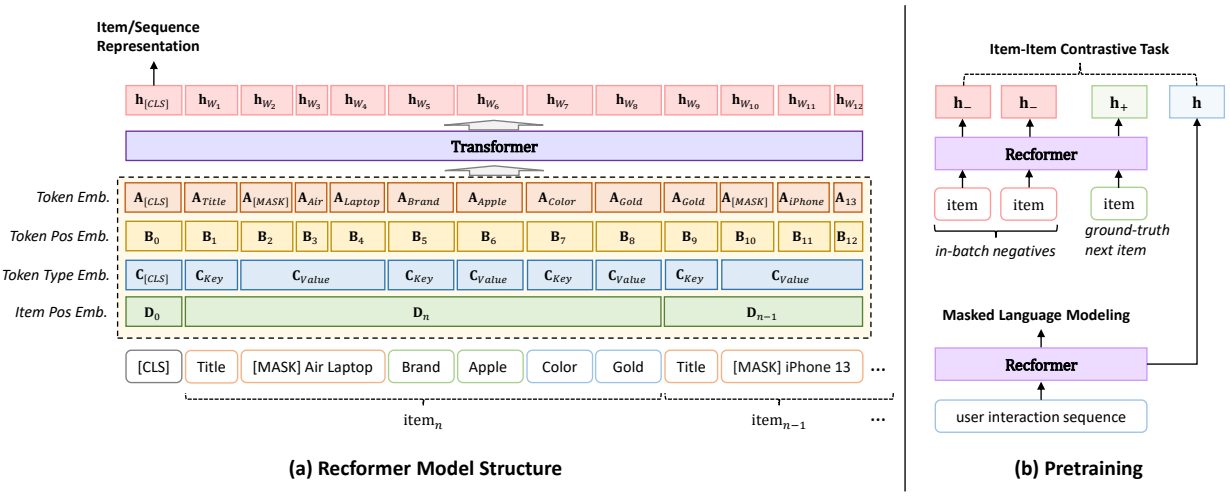


Figure 7.1: Architecture of Recformer. Recformer represents items as text-based key-value pairs (i.e. Brand: Apple, Colour: Gold). Figure taken from [Li et al., 2023]

textual representations. Under this paradigm, the LLM text-generation task is mapped to generate text for recommended items.

For example, Recformer [Li et al., 2023] proposes an item representation comprising learned embeddings of text-based key-value pairs (figure 7.1). TIGER [Rajput et al., 2023] investigates a semantic ID representation for items, along with a T5-style [Raffel et al., 2020] training and generative approach for retrieval. UniSRec [Hou et al., 2022] learns transferable item representations based on item description, along with sequence representations with contrastive tasks. UniSRec can also incorporate IDs during fine-tuning.

Recent efforts explore incorporating collaborative filtering embeddings directly into a large-language model (LLM), motivated by the intuition that the LLM's language semantics facilitate item-context interpretation. P5 [Geng et al., 2022] represents all user-item signals, along with associated context, as natural language sequences that are input to an encoder/decoder model (figure 7.2). LC-Rec [Zheng et al., 2024] uses vector quantization for ID representation and LLM pre-training tasks to align items and language. CLLM4Rec [Zhu et al., 2024] adds new user and item tokens to the LLM, along with a training head to model the collaborative filtering signal. Hua et al. [2023] study methods for effectively adding item IDs to the LLM token vocabulary. User-LLM [Ning et al., 2025] encodes user history into user embeddings and incorporates them into an LLM via cross-attention. Harte et al. [2023] experiment with extracting item embeddings from a pretrained LLM to

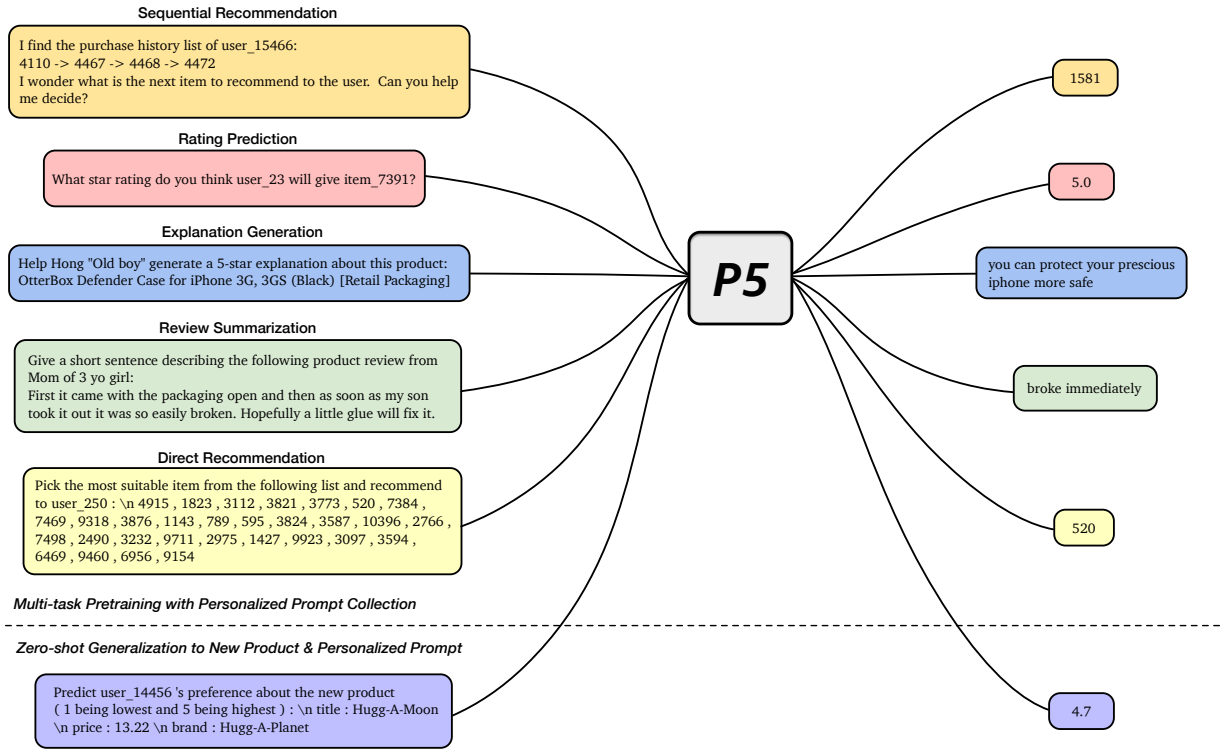


Figure 7.2: Text-to-text recommendation task of P5. Figure taken from Geng et al. [2022]

make recommendations directly, as well as to initialize Bert4Rec's item embeddings. CALRec [Li et al., 2024] proposes a two-stage LLM finetuning approach using a combination of contrastive and language modelling losses.

7.5 Interactive Recommender Systems and Critiquing

In Chapter 8, we also explore methods for incorporating the macro-context of user intent to guide the recommendation process. Earlier work in this space focuses on eliciting attribute preferences from users, typically at the feature level, through a process called critiquing [Burke et al., 1996, Tversky and Simonson, 1993, Tou et al., 1982, Smyth and McGinty, 2003, Pu and Faltings, 2004, Luo et al., 2020a, Shimazu, 2002]. An example of a critique would be asking the user whether they prefer red or black shirts, akin to a salesperson proactively asking questions to determine what items to show the user. Early critiquing works elicited feedback through suggested pre-designed tweaks [Burke et al., 1996] or through language-driven dialogue systems [Shimazu, 2002, Thompson et al., 2004, Grasch et al., 2013].

Recent work [Wu et al., 2019, Luo et al., 2020b,a] explores systems that jointly learn latent representations for users, items, and key-phrases, the latter of which serve as critiques that can influence the recommendations through operations in the latent space. Others explore the interpretation of attributes [Nema et al., 2021, Balog et al., 2021] to support critiquing. We view hybrid models that associate item sequences with textual context signals as natural extensions of recent critiquable recommenders. In particular, we believe that the intrinsic world knowledge provided by LLMs offers a promising avenue for incorporating critiquing signals into a hybrid recommender.

Chapter 8

FLARE: Fusing Language Models and Collaborative Architectures for Recommender Enhancement

Modern web experiences depend on recommender systems to help users discover personalized and relevant content. As the volume of available items, from streaming media to e-commerce, far exceeds the cognitive processing capacity of any individual user, the need for automated, intelligent filtering has become increasingly important. Recommender Systems (RS) aim to address this challenge. Research in recommender systems has evolved from simple heuristic filters into complex neural systems capable of modelling dynamic user preferences with high precision.

Of particular interest are systems that incorporate the sequential order of interactions as an extra source of context [Donkers et al., 2017, Hidasi et al., 2016a, Kang and McAuley, 2018, Sun et al., 2019]. For instance, a user who buys a cell phone will likely be interested in purchasing a case for it. Note that this pattern is directly grounded in the temporal order of interactions. Without temporal dynamics, a recommender system might recommend another phone because it is similar to the item the user purchased. Sequential recommenders are often implemented by modelling sequences of collaborative filtering embeddings (Section 7.2.1), which we refer to as "item-ID only".

With the advent of large language models, some methods have proposed hybrid recommenders that also model sequential context (e.g., textual descriptions, categories, attributes) for items [Hidasi et al., 2016b, Huang et al., 2019, Zhang et al., 2019]. These models suggest that local item context (without macro-level user grounding) is itself in-

formative, with the potential to improve overall recommendation quality and address the cold-start problem for new items. With the advent of transformer-based large language models, more recent work has focused on forgoing collaborative filtering embeddings entirely and concentrating solely on free-form text and attributes [Li et al., 2023, 2024, Zhou et al., 2020, Rashed et al., 2022, Qiu et al., 2021, Ning et al., 2025, Rajput et al., 2023]. These text-only works extend the thesis of sequential context by assuming that *only* local item context is required to make informative recommendations. Indeed, these approaches demonstrate that using text improves performance over item-ID-only baselines, such as Bert4Rec [Sun et al., 2019], which is widely used to justify these new approaches.

However, these comparisons often overlook a significant disparity in model scale and evaluation contexts. Text-based methods typically leverage Large Language Models (LLMs) with millions (or even billions) of parameters. At the same time, Bert4Rec ID-only baselines are frequently diminutive, often using the default configurations proposed in the original paper with only two layers and 64 dimensions[1], and therefore remain significantly undertuned. Furthermore, evaluations are commonly conducted on relatively small datasets (≈40k items); this setting inherently favours LLMs, which can rely on rich pre-trained semantic knowledge, while disadvantaging ID-based models that require larger volumes of interaction data to learn robust embeddings from scratch. The gap in the item scale also raises questions about whether text-only approaches remain practical and feasible in a more realistic industrial setting.

Our work starts by re-evaluating the potential of macro-level item-ID information. Our results from extensive tuning and scaling of Bert4Rec, referred to as Bert4Rec*, consistently exceed those of standard Bert4Rec, with some datasets demonstrating relative improvements of up to +65% recall@10 (measuring whether the item appears in the top 10 items) and +31% nDCG@10 (measuring the predicted rank of the correct item). These findings suggest that ID-only models are more competitive than previously assumed, indicating value in macro-level user similarity.

In summary, collaborative filtering item embeddings leverage macro-level user similarity and are more competitive than previously assumed, but are susceptible to the cold-start problem and shifting user needs. On the other hand, text-only models overcome the cold-start problem and can benefit from rich semantic information, but ignore macro-level user similarity. Together with in-depth evaluation, the complementary strengths and weaknesses of these approaches motivate the central research question of this chapter: "How

[1]This configuration can be seen as the default in RecBole, the industry standard library for recommender systems baselines, which many of our referenced papers use verbatim: https://github.com/RUCAIBox/RecBole/blob/master/recbole/properties/model/BERT4Rec.yaml

can we effectively integrate the benefits of LLM-powered local item context embeddings with macro-level collaborative filtering embeddings?"

To explore this question, this chapter presents Flare, a simple hybrid recommender that extends Bert4Rec by incorporating a modern text encoder using a Perceiver [Jaegle et al., 2021] network. Rather than directly using collaborative filtering item embeddings, Flare uses a Perceiver network to intelligently distill local text-based item context embeddings and a macro-level collaborative filtering embedding into a single learnable latent embedding. This integration enables efficient synthesis of text-based item context and item embeddings, providing a richer representation for recommendations. Flare outperforms state-of-the-art models on certain datasets, indicating that combining IDs and text unlocks additional benefits.

To empirically validate these benefits, we conduct an extensive evaluation across multiple categories in the Amazon Product Reviews dataset. We also assess the applicability of our work on the significantly larger Clothing, Shoes, and Jewelry category, which comprises 376k unique items. Our findings indicate that Flare's effectiveness and performance trends are consistent across both small and large-scale settings, demonstrating the scalability and generalizability of our hybrid architecture in more realistic scenarios.

Beyond evaluating historical data, a key capability for a recommender is also its ability to adapt to shifting user intent. Standard ranking metrics, while essential for assessing historical prediction accuracy, are insufficient for this purpose as they do not measure a model's responsiveness to explicit, real-time feedback. To address this methodological gap, we propose and formalize critiquing [Chen and Pu, 2012, Gao et al., 2021, Jannach et al., 2021] as an evaluation task for text-aware recommenders. This paradigm assesses a model's capacity to integrate user-intent instructions and local item context to guide its predictions. We do this by using the category of the next item, as given in natural language, and measuring the overlap between the command and the retrieved items. We evaluate this overlap through our proposed metric, Cat-nDCG, and present baselines for this task on the Amazon Product Reviews dataset.

In this chapter, we make the following contributions:

- A empirical re-evaluation of ID-only baselines. We demonstrate that a well-tuned and scaled Bert4Rec* significantly outperforms prior results, challenging the prevailing trend toward text-only recommenders. This finding establishes the continued value of macro-level collaborative signals.
- The proposal of Flare, a novel hybrid architecture. Flare directly addresses our central research question by effectively integrating local, text-based item context

with macro-level, collaborative filtering embeddings. It uses a Perceiver Transformer network to distill these signals into a single learnable latent embedding, achieving superior performance compared to models that rely on either context in isolation.

- The establishment of new, robust baselines on a large-vocabulary dataset. By evaluating our models on a dataset with 376k items, we address a significant limitation in the literature concerning model scalability. Our results validate the practical applicability of the hybrid approach in more realistic, industrial-scale settings and provide a new benchmark for future research.
- The formalization of critiquing as an evaluation paradigm. We introduce a comprehensive methodology and baselines for this task. Critiquing offers a new way to measure a model's ability to adapt to the crucial macro-context of explicit, real-time user intent, a capability that standard ranking metrics fail to capture.

8.1 Method

8.1.1 Task Definition

We formulate the recommendation task as follows. A dataset consists of user sequences, with each sequence $U = \{i_1, i_2, ..., i_n\}$ containing items $i \in \mathcal{I}$. Each item i is a tuple (ID, T), where ID is the unique identifier of the item, and $T = \{t^1, t^2, ..., t^m\}$ is a natural language description of the item consisting of text tokens t. Each description T comprises textual content (e.g. item title, category, brand, price, etc). The task is to predict i_{n+1} given U.

8.1.2 Flare Model Architecture

Flare builds upon the foundational Bert4Rec architecture [Sun et al., 2019], augmenting the collaborative filtering backbone with rich item context features (Figure 8.1). As with standard sequential recommendation models, Flare adapts the Masked Language Modelling (MLM) task for recommendations [Sun et al., 2019]. During training, a random subset of items in the user's interaction sequence is replaced with a special mask token $\langle mask \rangle$. The model's objective is to reconstruct these masked items by reasoning about the surrounding context.

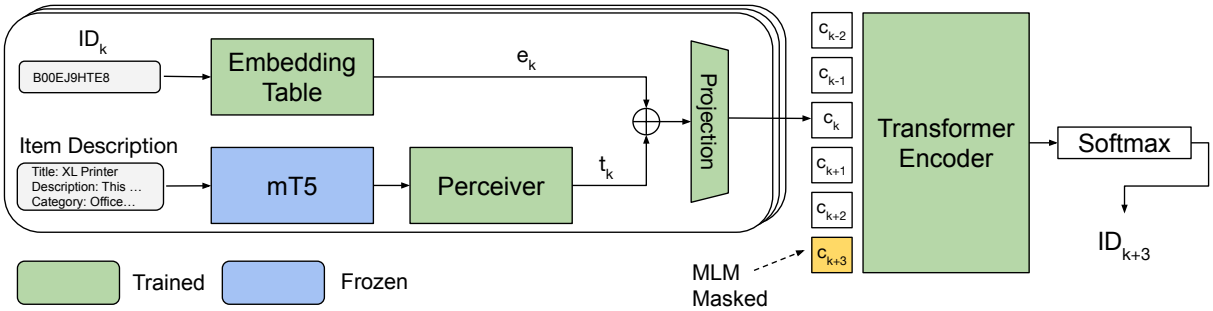


Figure 8.1: Sequential recommender model architecture with text encoder augmentation. Text tokens are encoded with a frozen (blue) language model (mT5), and the associated embeddings are resampled with the Perceiver. Components in green boxes are trained using masked language modelling (MLM).

For example, an input sequence of item IDs $[ID_1, ID_2, ID_3, ID_4, ID_5]$ might be transformed via random masking into $[ID_1, \langle mask \rangle_1, ID_3, \langle mask \rangle_2, ID_5]$. The training labels for this sequence would be $\langle mask \rangle_1 = ID_2$ and $\langle mask \rangle_2 = ID_4$. Following the item embedding lookup, the input to the initial transformer layer is represented as:

$$[e_1, e_{\langle mask \rangle}, e_3, e_{\langle mask \rangle}, e_5]$$

where e_i represents the learnable ID embedding for item ID_i, and $e_{\langle mask \rangle}$ denotes the shared mask embedding. The transformer layers process the masked sequence, and the outputs for the masked positions are fed into a softmax over the item vocabulary. The loss is calculated as the average negative log-likelihood:

$$\mathcal{L}_{MLM} = \frac{1}{|U^m|} \sum_{ID_m \in U^m} P(ID_m = ID_m^* | U') \tag{8.1}$$

where U^m is the set of masked items, ID_m^* is the ground truth item, and U' is the masked input sequence.

To integrate macro-level semantic context, Flare modifies the input representation. Instead of relying solely on the ID embedding e_i, we augment it with a context embedding t_i derived from the item's textual metadata (e.g., title, description). The input sequence becomes:

$$[c_1, e_{\langle mask \rangle}, c_3, e_{\langle mask \rangle}, c_5]$$

where the combined representation c_i is the vector sum of the ID embedding e_i and the projected context embedding t_i.

Context Encoding via Perceiver

A core architectural challenge in hybrid recommendation is mapping variable-length textual descriptions to a fixed-size embedding space compatible with item IDs. To achieve this, we employ a frozen, pre-trained Large Language Model (LLM) encoder (specifically, mT5 Base [Xue et al., 2021]) followed by a Perceiver [Jaegle et al., 2021] network.

The Perceiver is a specialized Transformer architecture designed to efficiently process high-dimensional, multimodal inputs. Unlike standard Transformers, which computationally scale quadratically with the input sequence length, the Perceiver scales linearly. It achieves this by introducing a small set of learnable *latent vectors* that act as queries in a cross-attention mechanism. Given the variable-length output sequence from the LLM encoder, the Perceiver's latent vectors iteratively query this input to extract the most salient semantic information. This process acts as a learned information bottleneck, distilling the unbounded textual data into a fixed-size, dense representation.

Using a Perceiver offers two distinct advantages over simple mean-pooling or truncation:

1. **Information Selection:** The cross-attention mechanism allows the model to learn which parts of a description are relevant for recommendation (e.g., focusing on "wireless" and "noise-cancelling" while ignoring filler words), rather than treating all tokens equally.

2. **Multimodal Flexibility:** The Perceiver can naturally handle inputs from multiple sources by simply concatenating them along the sequence dimension. This capability is critical to Flare's critiquing mechanism (Section 8.2.4), as it enables us to fuse item descriptions with user critique strings into a single joint encoding without modifying the downstream architecture.

Finally, the Perceiver's output is linearly projected to match the dimensionality of the item ID embeddings (e_i) and summed, enabling the main Transformer backbone to process a unified representation that combines collaborative and semantic signals.

8.1.3 ID-Text Contrastive Learning

We introduce an auxiliary contrastive loss to align item ID embeddings with item-context embeddings. This approach is motivated by the observation that the less frequent, or "tail" items, which have limited representation in the training data, can instead rely on their contextual representations. Conversely, frequently occurring items can benefit from

collaborative filtering signals that can be learned by training a sequential recommender. Aligning ID and context embeddings places both collaborative filtering (CF) and contextual signals in the same latent space, potentially allowing context to fill information gaps in CF embeddings. More precisely, we employ the InfoNCE loss function [Oord et al., 2018] as detailed below:

$$\mathcal{L}_c = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\frac{e_i\cdot t_i}{\tau}-m)}{\exp(\frac{e_i\cdot t_i}{\tau}-m)+\sum_{j=1,j\neq i}^{N}\exp(\frac{e_i\cdot t_j}{\tau})} \tag{8.2}$$

where τ is the softmax temperature, and we also include a softmax additive margin term m to each positive pair to increase the separation between positive and negative pairs, as proposed by Yang et al. [2019].

8.1.4 Critiquing

A core component of hybrid recommenders is their ability to effectively leverage both textual content semantics and collaborative filtering. However, bridging the gap between these two modalities remains challenging, potentially leaving the downstream sequential recommender unable to fully exploit the information in each modality.

We propose a novel masking and evaluation paradigm that uses critiquing to assess whether hybrid recommenders appropriately leverage both modalities. This consists of adding a textual instruction S to each item i in the user's sequence U, providing a broad hint to guide the model to the next recommended item. That is, each item i in the user sequence U becomes $i = (ID, T, S)$, where S is the critique instruction. For masked locations during training and inference, we only mask the ID and item text T, leaving S unmasked, which the model can utilize as a hint towards the target item (i.e., $i = ([\langle mask\rangle, \langle mask\rangle, S))$.

The key to this task is that hints are textual, therefore requiring recommenders to adapt to a text modality by relating the critique to the other signals in the user's sequence. Note that this task formulation has many practical applications in recommender systems, such as refining recommendations by leveraging the user's explicit natural-language query as a critique or by using the natural language description of the webpage subsection the user is currently navigating. In this work, because Amazon Product Reviews does not provide user queries, we use the item's product category as a proxy for the user intent critique string (section 8.2.4). In essence, critiquing strings can help steer a model's predictions to incorporate the user's current context.

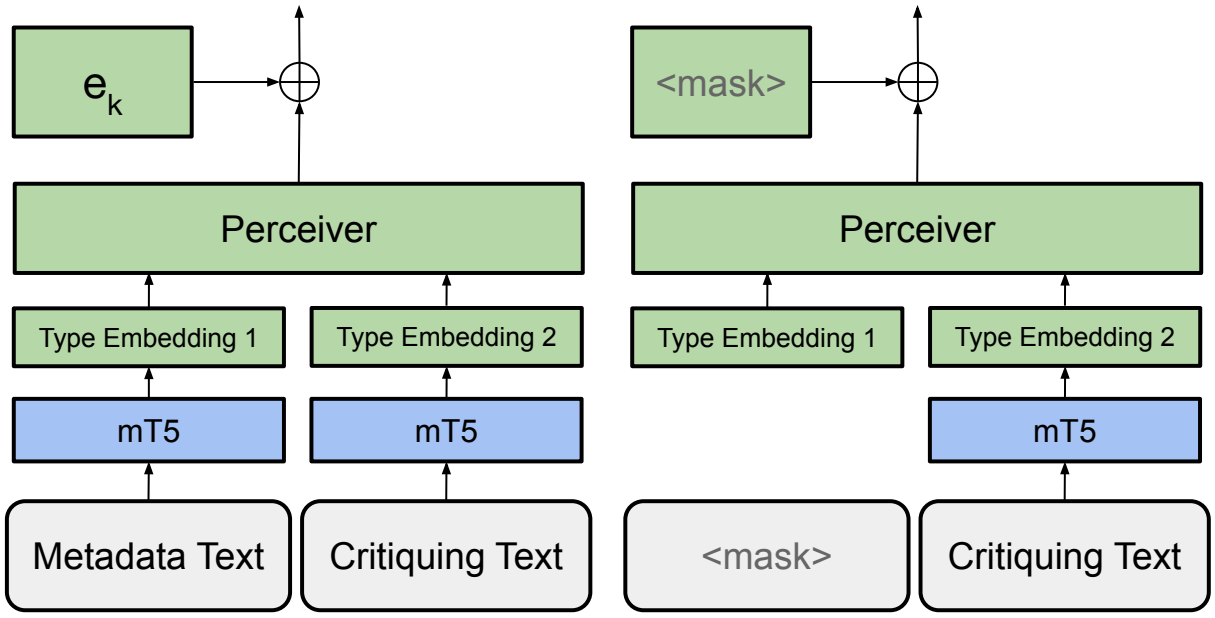


Figure 8.2: Flare adapted to incorporate recommendation critiques. Critiques are textual descriptions incorporated as a separate modality into the Perceiver to guide the model's predictions.

Flare's architecture is uniquely suited to including critique text thanks to the Perceiver network's flexibility. As described in Section 8.1.2, the Perceiver fuses multi-modal inputs by concatenating them along the sequence dimension. We treat the critique text as a distinct input modality, separate from the item description, allowing the model to learn the semantic interplay between "what the item is" (T) and "what the user wants" (S).

Mechanically, we encode both the item description T_k and the critique string S_k using the same frozen LLM encoder. To allow the model to distinguish between these modalities, we add learnable segment embeddings (θ_T and θ_S) to the respective output tokens. The Perceiver then ingests the concatenated sequence of description and critique tokens. The context embedding t_k and the final augmented representation c_k are computed as:

$$T'_k = \text{Encoder}(T_k) + \theta_T \tag{8.3}$$

$$S'_k = \text{Encoder}(S_k) + \theta_S \tag{8.4}$$

$$t_k = \text{Linear}\left(\text{Perceiver}\left([S'_k \ ; \ T'_k]\right)\right) \tag{8.5}$$

$$c_k = t_k + e_k \tag{8.6}$$

where [;] denotes concatenation, e_k is the learnable ID embedding, and the Linear layer projects the Perceiver's output to the hidden dimension of the backbone Transformer. By integrating the critique at the fusion layer, Flare dynamically modulates the item's representation based on the explicit user intent before the sequential backbone processes it.

When masking items for training or during inference, we replace T'_k and e_k at the desired positions with $\langle mask \rangle$ embeddings, leaving the model to utilize S' as a direction instruction for predictions (Figure 8.2). By treating critique text as a separate input modality, we hypothesize that the model can learn to associate critique text with item context. Note that this input is optional; if no critique is provided, the model processes the item description T'_k in isolation.

8.1.5 Training

We train Flare using the MLM loss by masking items in the user's sequence with 15% probability, ensuring that at least one item is masked. For the total loss, we use a weighted sum of MLM loss and the ID-Text contrastive loss:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{MLM} + (1 - \alpha)\mathcal{L}_c \tag{8.7}$$

controlled by the hyper-parameter α, which we found $\alpha = 0.5$ yielded optimal results. For the contrastive loss, we use a softmax temperature $\tau = 0.3$ and softmax additive margin term $m = 0.2$.

	Recformer			CALRec		
Dataset	# Users	# Items	I/U	# Users	# Items	I/U
Games	55,132	17,389	9.00	50,940	17,383	8.97
Office	101,357	27,932	7.88	44,736	27,482	7.87
Scientific	11,011	5,327	6.98	9,461	5,282	7.04
Music	27,525	10,611	8.40	25,577	10,599	8.39
Arts	56,110	22,855	8.77	47,197	22,828	8.72
Pets	47,569	37,970	8.84	43,135	37,712	8.82
Clothing	-	-	-	1,219,592	376,843	9.20

Table 8.1: Statistics for the Recformer- and CALRec-processed Amazon Product Reviews dataset. I/U is the average number of items per user, or average sequence length. Note that the Clothing dataset preprocessing follows the same steps as CALRec

8.2 Experimental Results

This section investigates the following research questions:

- **RQ1**: Do the previously reported Bert4Rec baseline models have potential for further performance gains through tuning?
- **RQ2**: Can hybrid models like Flare perform well on larger item count datasets?
- **RQ3**: Can Flare relate textual critique commands to sequential user signals?
- **RQ4**: Can Flare adapt to critique commands that are out of sequence?

8.2.1 Experimental Setup

Datasets

To evaluate the effectiveness of Flare, we utilize datasets from the Amazon Product Reviews corpus [Ni et al., 2019]. This extensive collection comprises 233.1 million reviews of Amazon Products spanning May 1996 to October 2018 across 29 distinct categories. The dataset provides rich metadata for both product descriptions (title, description, category, and price) and user reviews (text, timestamp, rating, and summary) (Figure 8.3). We

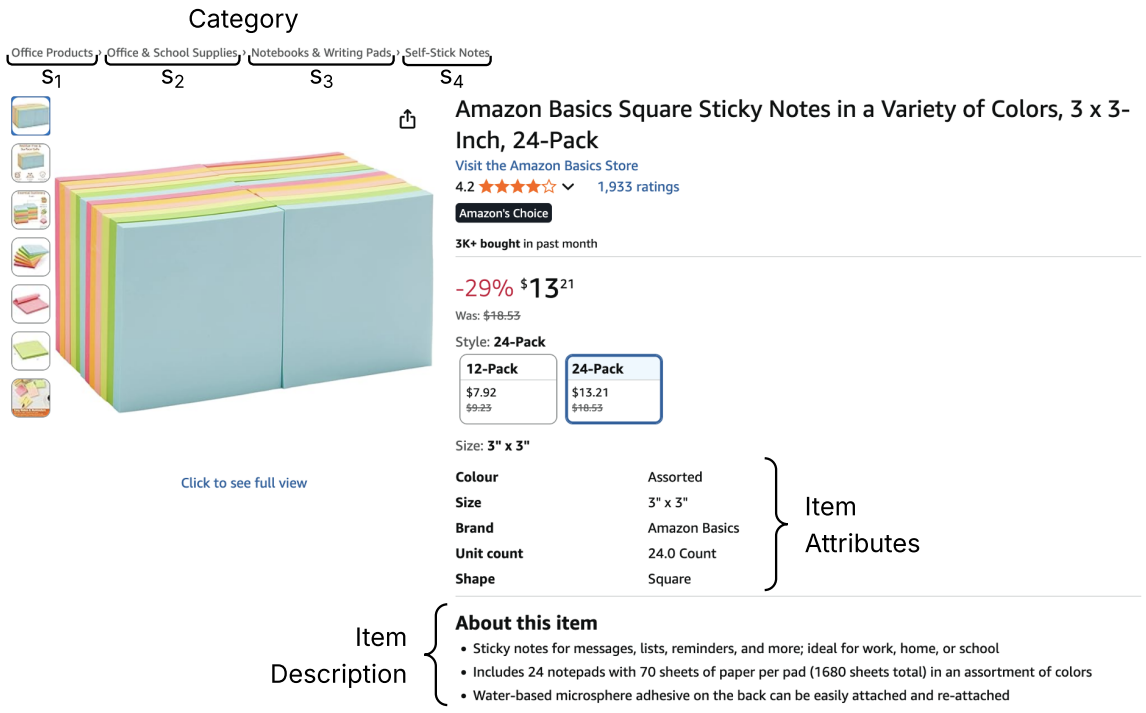


Figure 8.3: Annotated example of a product in the Amazon Product Reviews dataset. Note the category string of "Office Products - Office & School Supplies - Notebooks & Writing Pads - Self-Stick Notes".

construct sequential recommendation tasks by grouping reviews by user and sorting them chronologically, and task the model with predicting the next product the user will review (i.e., the next item) in the sequence.

We benchmark against Recformer [Li et al., 2023] and CALRec [Li et al., 2024], which report the current state-of-the-art results. Both methods evaluate performance on the same six subsets of the Amazon corpus: "Video Games", "Office Supplies", "Industrial and Scientific", "Arts, Crafts and Sewing", "Pet Supplies", and "Musical Instruments.". To meaningfully compare our results with those methods, we generated two versions of these datasets, each following its unique preprocessing procedure.

The primary distinction between the two preprocessing pipelines lies in how they handle redundant interactions. Recformer does not perform deduplication, meaning that identical interaction events may appear multiple times within a user's sequence [Li et al., 2023]. It also manages sequence length by trimming user histories to a maximum of 50 items. Conversely, the CALRec implements a stricter cleaning process by deduplicating user sequences [Li et al., 2024]. We implement deduplication by removing consecutive items from the same user that share identical Item IDs (ASINs), review texts, ratings, and timestamps. The motivation for this deduplication is to mitigate the influence of trivial recommendation strategies, such as simply recommending the most recently interacted item. To manage sequence length, CALRec filters out user sequences exceeding 50 items, reducing overall data volume [Li et al., 2024]. At a high level, CALRec pre-processing is a stricter, more challenging version of the Amazon Product Reviews dataset compared to Recformer.

Finally, to evaluate the scalability of our approaches, we also establish new baselines on the larger "Clothing, Shoes and Jewelry" subset. This subset contains $\approx 378,000$ unique items, roughly $9\times$ the scale of the largest subset used by our text-based recommendation baselines. For this large-scale evaluation, we adopt the robust CALRec preprocessing protocol, applying the same deduplication and filtering criteria to ensure high data quality. Comprehensive statistics for all datasets and processing variants are provided in Table 8.1.

Baselines

We evaluate Flare against three distinct classes of state-of-the-art sequential recommenders: ID-only, text-only, and hybrid architectures. As a representative of ID-only approaches, we employ Bert4Rec [Sun et al., 2019], which utilizes a transformer backbone trained via BERT-style [Devlin et al., 2019] masked language modelling on item sequences. To ensure a rigorous comparison, we created updated Bert4Rec baselines derived from extensive

Parameter	Games	Office	Scientific	Music	Arts	Pets
# Layers	2	2	4	8	2	2
# Heads	2	2	16	8	8	16
Model Dims	64	64	512	256	768	1024
Hidden Dims	256	256	2048	1024	3072	4096
Learning Rate	1e-3	1e-3	1e-4	1e-5	1e-4	1e-4
Batch	1	1	16	16	16	32
Steps	50K	50K	50K	50K	50K	10K

Table 8.2: Hyperparameters used by Bert4Rec for each Recformer dataset

hyperparameter tuning, which we denote as Bert4Rec*. We emphasize that Bert4Rec* is architecturally identical to Bert4Rec.

For methods incorporating textual semantics, we benchmark against both text-only and hybrid baselines. In the text-only category, we compare against Recformer [Li et al., 2023] and CALRec [Li et al., 2024], which rely exclusively on text-based representations of user history and currently hold state-of-the-art results. For hybrid (Text + ID) models, we include S^3Rec [Zhou et al., 2020], which uses mutual information maximization to align item and attribute representations, and UniSRec [Hou et al., 2022], which combines text-based representations with ID fine-tuning. Finally, we note a distinction in training methodology: while Recformer and CALRec employ a two-stage paradigm of pre-training across multiple datasets followed by fine-tuning [Li et al., 2024, 2023], Flare is trained only on the target dataset.

Evaluation Settings

Following prior work [Li et al., 2023, Zheng et al., 2024, Sun et al., 2019], we adopt the leave-one-out evaluation strategy. We reserve the user's penultimate and last items for validation and testing, respectively, and use the remaining items for training. Since training sequences can be of varying length (Table 8.1), we pack multiple training sequences into a single input example such that each packed example is as close to the same size as possible. Then, during training, we restrict attention to just the items within the same sequence. This maximizes hardware throughput and avoids wasteful computation caused by excessive padding tokens. As a result, the effective batch size during training can vary from step to step, depending on sequence length.

For our experiments, we use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, following

Parameter	Games	Office	Scientific	Music	Arts	Pets
# Layers	4	4	2	4	2	4
# Heads	16	16	8	16	8	16
Model Dims	1024	1024	768	1024	768	1024
Hidden Dims	4096	4096	3072	4096	3072	3072
Learning Rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Batch	32	32	32	32	32	32
Steps	10K	10K	5K	5K	10K	10K

Table 8.3: Hyperparameters used by Bert4Rec for each CALRec dataset

Parameter	Games	Office	Scientific	Music	Arts	Pets
# Layers	8	2	2	2	4	2
# Heads	16	4	8	2	4	8
Model Dims	1024	768	256	1024	512	256
Hidden Dims	4096	3072	1024	4096	2048	1024
Learning Rate	1e-4	1e-4	1e-4	1e-5	1e-4	1e-4
Batch	2	16	8	2	8	16
Steps	5K	5K	5K	25K	10K	25K
Perceiver Heads	16	16	8	8	2	8
Perceiver Layers	8	2	2	8	2	2
Perceiver Latents	2	8	2	4	2	2
Weight Decay	1e-3	1e-2	1e-3	1e-3	1e-3	1e-3

Table 8.4: Hyperparameters used by Flare for each Recformer dataset

Parameter	Games	Office	Scientific	Music	Arts	Pets
# Layers	8	4	2	2	4	4
# Heads	4	4	2	2	8	4
Model Dims	256	128	1280	128	128	512
Hidden Dims	1024	512	5120	512	512	2048
Learning Rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Batch	32	32	64	32	32	16
Steps	50K	10K	10K	10K	10K	10K
Perceiver Heads	8	8	8	8	8	8
Perceiver Layers	4	4	4	4	4	4
Perceiver Latents	4	4	4	4	4	4
Weight Decay	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3

Table 8.5: Hyperparameters used by Flare for each CALRec dataset

standard practice [Reddi et al., 2018]. We also performed a hyperparameter sweep over the number of transformer layers and heads; model and hidden dimensions; learning rate; batch size; total step count; weight decay; use of contrastive loss; number of Perceiver layers; and number of heads and latents for the Perceiver network. Tables 8.2 and 8.3 for Bert4Rec and tables 8.4 and 8.5 for Flare summarize the best parameters for the Recformer and CalRec datasets, respectively.

Metrics

To evaluate the effectiveness of our models, we employ three standard top-k ranking metrics: recall@k, normalized discounted cumulative gain (nDCG@k), and mean reciprocal rank (MRR). In contrast to standard retrieval metrics, which only consider whether the top-ranked item matches the correct item, top-k metrics assess whether an item appears among the top k recommendations. For recall and nDCG, we report results with $k = 10$, unless otherwise specified.

Recall@*k* measures the proportion of relevant items successfully retrieved within the top-k recommendations. In the context of next-item prediction, where there is a single ground-truth item for each user u, this is equivalent to Hit Ratio (HR@k) and is defined as:

$$\text{Recall@}k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{1}(\text{rank}_u \leq k) \tag{8.8}$$

where $\mathcal{U}$ is the set of users, rank_u is the rank of the ground-truth item in the recommendation list for user u, and $\mathbb{1}(\cdot)$ is the indicator function.

nDCG@k evaluates the ranking quality by assigning higher scores to relevant items that appear earlier in the recommendation list [Järvelin and Kekäläinen, 2002, 2017]. It is calculated as the Discounted Cumulative Gain (DCG) normalized by the Ideal DCG (IDCG):

$$\text{nDCG@}k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\text{DCG}_u@k}{\text{IDCG}_u@k}, \quad \text{DCG}_u@k = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)} \tag{8.9}$$

where rel_i represents the binary relevance of the item at position i (1 if it matches the ground truth, 0 otherwise).

MRR (Mean Reciprocal Rank) measures the average reciprocal rank of the first relevant item in the recommendation list, focusing on the model's ability to place the correct item as high as possible:

$$\text{MRR} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{\text{rank}_u} \tag{8.10}$$

For example, if the correct item appears at rank 3, the reciprocal rank for that given item is $1/3 = 0.333$. The final MRR is computed as the average of the reciprocal ranks of the predicted item for each user in the test set.

8.2.2 Flare Performance on Existing Benchmarks

To assess the efficacy of Flare, we first evaluate our models against the current state of the art on standard, commonly used datasets. We compare both our tuned ID-only baseline, Bert4Rec*, and our hybrid model, Flare, against reported reference results from Recformer [Li et al., 2023] and CALRec [Li et al., 2024]. We ensure that we follow an identical evaluation methodology and data preprocessing (section 8.2.1).

Tables 8.7 and 8.6 present the performance results using the metrics defined in section 8.2.1. **Bold** values indicate the best result across all models, and underlined values indicate the second best. The "%ID" column denotes the relative performance improvement of Flare over our optimized Bert4Rec* baseline.

Our investigation begins with an analysis of Bert4Rec*. Here, we sought to investigate whether ID-only models are significantly underparameterized. We find that our tuned baseline is remarkably competitive. First, we find that Bert4Rec* substantially outperforms the original Bert4Rec baselines reported in prior literature. We observe that these

Dataset	Metrics	Bert4Rec	UniSRec	Recformer	S^3Rec	Bert4Rec*	Flare	%ID
Pets	Recall@10	0.077	0.093	**0.116**	0.104	0.101	<u>0.112</u>	+10.89%
	nDCG@10	0.060	0.070	**0.097**	0.074	0.081	<u>0.092</u>	+13.58%
	MRR	0.059	0.065	**0.094**	0.071	0.077	<u>0.088</u>	+14.29%
Office	Recall@10	0.121	0.126	0.140	0.119	<u>0.141</u>	**0.148**	+4.96%
	nDCG@10	0.097	0.092	0.114	0.091	<u>0.115</u>	**0.121**	+5.22%
	MRR	0.093	0.085	<u>0.109</u>	0.096	<u>0.109</u>	**0.115**	+5.50%
Arts	Recall@10	0.124	0.133	**0.161**	0.140	0.144	<u>0.153</u>	+6.25%
	nDCG@10	0.094	0.089	**0.125**	0.103	0.111	<u>0.121</u>	+9.01%
	MRR	0.090	0.080	**0.119**	0.106	0.103	<u>0.114</u>	+10.68%
Games	Recall@10	0.103	0.092	0.104	0.088	<u>0.110</u>	**0.117**	+6.36%
	nDCG@10	0.063	0.058	0.068	0.053	<u>0.072</u>	**0.075**	+4.17%
	MRR	0.059	0.055	0.065	0.050	<u>0.067</u>	**0.069**	+2.99%
Music	Recall@10	0.097	<u>0.112</u>	0.105	0.111	0.103	**0.112**	+8.74%
	nDCG@10	0.071	0.079	<u>0.083</u>	0.079	0.078	**0.085**	+8.97%
	MRR	0.068	0.074	<u>0.081</u>	0.076	0.073	**0.081**	+10.96%
Scientific	Recall@10	0.106	<u>0.126</u>	**0.145**	0.080	0.113	0.119	+5.31%
	nDCG@10	0.079	<u>0.086</u>	**0.103**	0.045	0.080	0.084	+5.00%
	MRR	0.076	<u>0.079</u>	**0.095**	0.039	0.074	0.076	+2.70%

Table 8.6: Flare performance against prior baselines on the RecFormer-processed [Li et al., 2023] datasets. Baseline results are quoted from those papers. Bert4Rec* is our tuned Bert4Rec, which outperforms previously reported Bert4Rec baselines. The %ID column is the difference between Flare and Bert4Rec*. The best result is **bolded**; the second best is <u>underlined</u>.

Dataset	Metrics	Bert4Rec	Recformer	S^3Rec	UniSRec†	CALRec	Bert4Rec*	Flare	%ID
Pets	Recall@10	0.050	0.083	0.065	0.097	0.094	<u>0.100</u>	**0.109**	+9.00%
	nDCG@10	0.038	0.059	0.039	0.057	0.074	<u>0.076</u>	**0.082**	+7.89%
	MRR	0.037	0.055	0.035	0.050	0.070	<u>0.073</u>	**0.078**	+6.84%
Office	Recall@10	0.071	0.106	0.094	<u>0.122</u>	0.121	0.117	**0.124**	+5.98%
	nDCG@10	0.055	0.069	0.061	0.071	**0.098**	<u>0.094</u>	**0.098**	+4.26%
	MRR	0.052	0.060	0.054	0.060	**0.093**	<u>0.089</u>	**0.093**	+4.49%
Arts	Recall@10	0.080	0.110	0.105	**0.132**	0.114	0.107	<u>0.115</u>	+7.48%
	nDCG@10	0.055	0.066	0.060	0.073	**0.086**	0.079	<u>0.082</u>	+3.80%
	MRR	0.051	0.058	0.052	0.062	**0.082**	0.074	<u>0.077</u>	+4.05%
Games	Recall@10	0.065	0.072	0.088	**0.107**	<u>0.099</u>	0.078	0.098	+25.64%
	nDCG@10	0.033	0.038	0.042	0.050	**0.059**	0.044	<u>0.054</u>	+22.73%
	MRR	0.031	0.035	0.037	0.043	**0.051**	0.039	<u>0.048</u>	+23.07%
Music	Recall@10	0.092	0.094	0.105	**0.126**	0.116	0.109	<u>0.118</u>	+8.26%
	nDCG@10	0.068	0.060	0.061	0.071	**0.091**	0.085	<u>0.088</u>	+3.53%
	MRR	0.066	0.054	0.053	0.061	**0.086**	0.081	<u>0.084</u>	+3.70%
Scientific	Recall@10	0.064	0.106	0.073	**0.118**	<u>0.112</u>	0.083	0.093	+12.05%
	nDCG@10	0.041	<u>0.064</u>	0.041	<u>0.064</u>	**0.079**	0.055	0.060	+9.09%
	MRR	0.039	<u>0.057</u>	0.038	0.056	**0.073**	0.050	0.055	+10.00%

Table 8.7: Flare performance against prior baselines on the CALRec-processed [Li et al., 2024] datasets. Bert4Rec* is our tuned Bert4Rec, which outperforms previously reported Bert4Rec baselines. The %ID column is the difference between Flare and Bert4Rec*. The best result is **bolded**; the second best is <u>underlined</u>.

improvements are most profound on datasets with larger item vocabularies. For instance, on the CALRec Music dataset (10k items), Bert4Rec* achieves a +18% improvement in Recall@10 and +25% in nDCG@10 over the original baseline. However, on the significantly larger CALRec Pets dataset (37k items), these gains increase to +100% for both recall and nDCG@10 and to +97% for MRR. We observe similar trends in the Recformer-processed datasets. This strongly reinforces our hypothesis that standard ID-only baselines are frequently under-parameterized. This suggests that scaling ID-only can more accurately incorporate macro-level collaborative context.

Second, somewhat surprisingly, we also find that Bert4Rec* occasionally outperforms much larger text-only models that rely on LLMs. On the RecFormer-processed "Office" and "Games" datasets, our ID-only model outperforms the LLM-based RecFormer model. Specifically, in Recformer Games, Bert4Rec* achieves +5.8% recall@10 and +5.9% nDCG@10 over Recformer. Furthermore, Bert4Rec* generally outperforms other hybrid baselines (UniSRec and S^3Rec) across Recformer datasets, except for Music. On the Office dataset, for example, it achieves recall@10 improvements of 11.9% and 26% over UniSRec and S^3Rec, respectively. These findings empirically validate the continued importance of collaborative filtering signals.

Our results also indicate strong additive performance for Flare, which builds on the strong foundation set by Bert4Rec* by combining collaborative signals with semantic text. We find that Flare consistently outperforms Bert4Rec*. Across all datasets, we observe relative improvements ranging from $\approx +4\%$ to $\approx +25\%$ (denoted by the %ID column). The most significant gains are observed in the CALRec Games dataset, where Flare achieves +25.64% recall@10, +22.73% nDCG@10, and +23.07% MRR over the ID-only baseline. We note that our performance on CALRec games also correlates with strong performance from other hybrid systems, suggesting that this domain particularly benefits from the complementary insights provided by textual metadata.

We also find that Flare achieves state-of-the-art performance across five categories: the Recformer-processed Office, Games, and Music datasets, as well as the CALRec-processed Pets and Office datasets. Across the remaining five datasets, Flare consistently achieves second-best performance. Notably, we find that Flare extends the lead over Recformer to +12.5% recall@10 and +10.3% nDCG@10 on the Recformer Games dataset.

The only domain where Flare exhibits diminished performance is the Scientific dataset. We theorize that this is attributable to the extreme data sparsity in this domain, which contains only $\approx$ 5k items (7.4$\times$ fewer than in Pets). We note that successful text-only competitors Recformer and CALRec employ a two-phase training paradigm, where the underlying language model is pre-trained on a massive corpus of examples from diverse

		Bert4Rec*				Flare			
Metrics	Recformer	Small	Base	Large	%	Small	Base	Large	%
Recall@1	-	0.079	0.127	0.128	-	0.123	<u>0.129</u>	**0.130**	-
Recall@5	-	0.106	<u>0.138</u>	<u>0.138</u>	-	0.137	**0.142**	**0.142**	-
Recall@10	0.135	0.112	0.143	0.143	+5.93%	0.143	**0.148**	<u>0.147</u>	+9.63%
nDCG@10	0.123	0.096	0.134	<u>0.135</u>	+9.76%	0.132	**0.138**	**0.138**	+12.20%
MRR	0.120	0.091	<u>0.133</u>	<u>0.133</u>	+10.83%	0.130	**0.136**	**0.136**	+13.33%

Table 8.8: Performance of Flare and Bert4Rec* (tuned Bert4Rec) scaling on "Clothing", a subset with over 370k items. Flare achieves improved performance when increasing the model size. The % columns show the performance improvement of the best model over Recformer.

domains before being fine-tuned on the target dataset. This allows them to leverage transfer learning to mitigate data scarcity. In contrast, Flare is trained from scratch on the target dataset. This result highlights the dependence of our current approach on sufficient in-domain training data and motivates exploring pre-training strategies for Flare, as discussed in Section 9.1 of this thesis, to better handle low-resource domains.

8.2.3 Performance on Large Datasets

Many production applications of recommender systems face vocabulary sizes that are orders of magnitude larger than those in the datasets above. As such, a question arises whether performance gains scale with larger vocabulary datasets. To study this problem, we use the "Clothing, Shoes and Jewelry" category from Amazon Product Reviews (376k Items) and report the performance of Flare models using different model sizes (Table 8.8). Given the larger candidate set, we also include recall@1 and recall@5.

We selected three representative model sizes reported as (i) "Small" with 2 layers, 2 heads, 64 model dimensions, and 256 hidden dimensions; (ii) "Base" with 8 layers, 16 heads, 768 model dimensions, and 3072 hidden dimensions; and (iii) "Large" with 32 layers, 32 heads, 768 model dimensions, and 3072 hidden dimensions. For all Flare models, the Perceiver model size is 8 layers, 16 heads, and 8 latents. The results show that increasing the model size improves performance, up to a point of diminishing returns. We benchmarked Flare against two baselines: Bert4Rec* and Recformer. Comparing the Flare models with the Bert4Rec* models enables us to verify that adding text is useful, independent of model size.

The Recformer paper did not evaluate performance on the Clothing dataset. Thus, we downloaded Recformer's published pretrained checkpoints and executed the fine-tuning pipeline. Before fine-tuning Recformer on our Clothing dataset, we confirmed our ability to successfully reproduce Recformer's published results on the "Scientific" dataset, within -0.5% and +3% across runs. Due to the much larger scale of the "Clothing" dataset, we reduced the batch size from 16 sequences to 8 to accommodate memory constraints.

For their small-dataset evaluation, Recformer pretrains on a disjoint set of datasets from the one used for fine-tuning. Recformer's pretrained checkpoint includes "Clothing", so fine-tuning that checkpoint on "Clothing" would depart from Recformer's evaluation methodology. Although we expect that, because the model sees the "Clothing" dataset multiple times, performance in this setting is "optimistic" and thus serves as a more challenging baseline. Even so, the most performant Flare model outperforms on all metrics and establishes a new baseline for this dataset (Table 8.8).

8.2.4 Critiquing

We evaluate Flare's ability to utilize textual signals using the critiquing task described below. We use the "Office" and "Clothing" datasets from Amazon Product Reviews, following the preprocessing steps as described for the "Clothing" dataset (Section 8.2.1). We formulate Amazon Product Reviews for this task by using each item's "category" attribute as the critique. Each provided category string consists of a hierarchical ordering of categories, with each category term progressively more precise, delimited by "-". For example, the product depicted in figure 8.3 would have the hierarchical category string "Office Products - Office & School Supplies - Notebooks & Writing Pads - Self-Stick Notes". This characteristic enables us to evaluate two different levels of critiquing:

- **Precise critiquing:** Critiquing with a specific category using the first four item categories (Ex: *Office Products - Office and School Supplies - Printer Ink & Toner - Inkjet Printer Ink*)
- **Broad critiquing:** Critiquing with a broad category using the first two item categories (Ex: *Office Products - Office and School Supplies*)

By including these text-based hints, we expect that a model that effectively leverages text semantics will achieve higher performance under precise critiquing, slightly lower performance under broad critiquing, and the worst performance without critiquing.

Dataset	Metrics	Precise	Broad	None
Office	Recall@1	**0.213**	0.138	0.124
	Recall@5	**0.355**	0.180	0.158
	Recall@10	**0.437**	0.203	0.173
	nDCG@10	**0.313**	0.168	0.147
	MRR	**0.286**	0.161	0.141
Clothing	Recall@1	**0.167**	0.135	0.129
	Recall@5	**0.229**	0.151	0.142
	Recall@10	**0.272**	0.161	0.148
	nDCG@10	**0.212**	0.147	0.138
	MRR	**0.202**	0.144	0.136

Table 8.9: Flare's performance when utilizing precise (4 categories) and broad (2 categories) textual critiquing. Flare achieves stronger performance under both forms of critiquing, demonstrating that it can leverage textual critiques, though it shows some room for improvement in broad critiquing.

Results

Results on precise and broad critiquing on the "Office Products" and "Clothing" datasets are shown in Table 8.9. Indeed, precise critiquing performance is higher than broad critiquing, and both outperform not using the critique text. Compared with the baseline without critiquing, precise critiquing achieved a recall@10 that was 153% and 84% higher on the "Office Products" and "Clothing" datasets, respectively. The vastly improved performance achieved with critiquing suggests that Flare can leverage textual instruction.

Notably, we observe that broad critiquing only improves performance by +17% and +9% recall@10 on the same datasets, a much smaller increase than that achieved with precise critiquing. We hypothesize that the reduced performance gain is due to the lack of specificity of the critiquing string, offering a challenge for future work. Section 8.2.6 details further qualitative analysis of critiquing outputs.

8.2.5 Out-of-Domain Critiquing

We now examine whether critiquing strings can steer the model toward out-of-domain categories from the original user sequence. To this end, we construct a variant of the

Office test set by randomly mutating the last item's critiquing category to another valid critiquing category in the dataset. We then evaluate our best pre-trained Flare model trained under precise critiquing on that modified dataset. Using category mutations, we evaluate Flare's ability to follow the critique, even if it is increasingly out of context, given the user's history. For example, rather than using the ground-truth shopping scenario for a desk chair, supported by a user sequence, we perturb the final critique and query the model to recommend a large industrial-grade TV. In this case, an effective model would handle the critique request despite being an increasingly distinct domain of the original, unmodified user sequence.

Category nDCG Metric

Since there are no longer ground-truth items for this task, we propose Cat-nDCG, a variant of nDCG, to quantify how well the model's predictions align with the mutated critiquing string.

We define Cat-nDCG as follows. Let S' be the "target" critiquing string given as input to the model. Let K be the set of retrieved items sorted by confidence, and $S_k = \{s_1, s_2, s_3, s_4\}$ be the set of hierarchically ordered categories associated with item $k \in K$. We then define a retrieved item k's relevance rel_k as

$$rel_k = |S_k \cap S'| \tag{8.11}$$

the number of categories in the predicted item k overlapping with the critiquing string S'. Because the categories in S and S' are hierarchical, an item that matches s_j in S' will also have matched all $s_i, i < j$. For instance, a retrieved item matching all four categories will have a relevance of 4, whereas an item only matching two parent categories (s_1, s_2 but not s_3, s_4) will have a relevance of 2. We then apply nDCG [Järvelin and Kekäläinen, 2002, 2017] as usual using these relevance values (equation 8.9). As a result, Cat-nDCG evaluates rank-wise whether the model retrieves items matching the given critique category.

Results

To construct the input critique S', we take an original category string $C_k = \{c_1, c_2, c_3, c_4\}$ for an item k. We consider the candidate set C_k^j obtained by randomly mutating each component category $c_l \in C_k$ with $j \leq l \leq 4$, such that the resulting string maps to a valid category in the dataset. We then randomly sample a critiquing string from this set, denoted S'. To ensure that the mutated categories are of high quality, we require that the

Mutated Categories	Cat-nDCG@10
Level 4 (s_4)	0.944
Level 3 (s_3, s_4)	0.917
Level 2 (s_2, s_3, s_4)	0.890

Table 8.10: Performance of Flare's category alignment when mutating the critiquing category string to another valid category on Office. Flare's predictions are nearly entirely within the desired category, even when the target category differs significantly from the user's prior actions.

chosen category S' contain at least five items in the dataset. As we decrease j, the mutated categories move farther from the original string C_k, thereby becoming increasingly difficult to adapt.

Our results setting $j = 4, 3, 2$, referred to as Level j, is shown in Table 8.10. Starting with Level 4 mutations, Flare achieves a Cat-nDCG@10 of 0.944, indicating that most retrieved items follow the critiquing string and are highly ranked. This finding suggests that Flare can generalize well to out-of-user distribution categories. Notably, when increasing to Level 2 mutations, Flare's Cat-nDCG@10 score drops to 0.890. This may be due to a conflict between the user's historical signals and the desired category value, suggesting an area for future work.

8.2.6 Qualitative Study of Critiquing

This section highlights several examples of recommendations generated through critiquing. Our study focuses on the user retrieved in Table 8.11, who originates from the Office Products category. The target held out item is the 5th item, a box of "T50 Heavy Duty Staples". Starting with non-critiquing approaches (table 8.12), we see that neither Bert4Rec* nor Flare can retrieve the correct item. Inspecting the retrieved items closer, we see Bert4Rec* does not retrieve any staples or staplers, while Flare can achieve partial relevance with rank 3 with the "Arrow" staples. We now compare to Flare with critiquing (table 8.13), where we include the target item's category ("Office & School Supplies - Staplers & Punches") as a critique instruction. Here, we see that the correct item is retrieved at rank 1, with all other items within the same category. Our finding aligns with our Cat-nDCG results in section 8.2.5, indicating strong instruction-following capability.

In the previous example, the target category can be identified in the user history as the most recently interacted item (rank 4). To analyze whether Flare's ability to follow

User Input History		
Seq.	Title	Product Category
1	Spectrum Diversified Scroll Mail and Letter Organizer, Wall Mount, Black	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets, Racks & Holders
2	Spectrum Diversified Scroll Mail and Letter Organizer, Wall Mount, Black	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets, Racks & Holders
3	Canon PGI-250/CLI-251 Combo Pack	Office & School Supplies - Printer Ink & Toner - Inkjet Printer Ink
4	Arrow Fastener 506 Genuine T50 3/8-Inch Staples, 1250-Pack	Office & School Supplies - Staplers & Punches - Staples
5*	T50 Heavy Duty Staples [Set of 2]	Office & School Supplies - Staplers & Punches

Table 8.11: Example user trajectory in the "Office Products" category. The 5th (*) item, the box of T50 staples, is held out and is the target next item. Note the preference for black.

critiquing instructions is sensitive to the user history, we compare outputs generated using a category outside the user history ("PDAs, Handhelds & Accessories - PDA & Handheld Accessories") and one from a different position in the user history ("Forms, Recordkeeping, & Money Handling - Key Cabinets & Holders"). Our findings in table 8.14 demonstrate that Flare is robust in both cases, indicated by the retrieved items in both cases, which all follow the respective critiquing instruction.

Finally, we examine Flare's ability to interpret varying levels of intent by analyzing recommendations generated from broad and precise critiques (Section 8.2.4). For the broad critique, we provide only the coarse, top-level category ("Office & School Supplies"). For precise critiquing, we focus on disambiguating between "Staples" and "Staplers", which both fall under the same training-time category, "Staplers & Punches". To test whether Flare can leverage additional detail to disambiguate between these cases, we manually construct a precise query ("Office & School Supplies - Staplers & Punches - Staples") that provides higher granularity than the category strings seen during training ("Office & School Supplies - Staplers & Punches). This setup aims to test whether Flare can generalize beyond memorized training critiques and leverage fine-grained semantic cues to distinguish between closely related items.

Example outputs using broad and precise critiquing are provided in table 8.15. Start-

Recommendations using Bert4Rec*		
Rank	Title	Product Category
1	Scotch Thermal Laminator 2 Roller System (TL901C)	Office Electronics - Presentation Products - Laminators
2	Spectrum Diversified Scroll Mail and Letter Organizer, Wall Mount, Black	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets, Racks & Holders
3	Scotch Thermal Laminating Pouches, 8.9 x 11.4-Inches, 3 mil thick, 20-Pack	Office Electronics - Presentation Products - Laminating Supplies
4	Quartet Dry Erase Board, Whiteboard / White Board, Magnetic, [...]	Office & School Supplies - Presentation Boards - Dry Erase Boards
5	Sakura Pigma 30062 Micron Blister Card Ink Pen Set, Black, 6CT Set	Office & School Supplies - Writing & Correction Supplies - Pens & Refills - Calligraphy Pens
Recommendations using Flare		
Rank	Title	Product Category
1	Sharpie 37001 Permanent Markers, Ultra Fine Point, Black, 12 Count	Office & School Supplies - Writing & Correction Supplies - Markers & Highlighters - Permanent Markers
2	Canon PGI-250XL Pigment Black Ink, Twin Value Pack, Compatible with [...]	Office & School Supplies - Printer Ink & Toner - Inkjet Printer Ink
3	Arrow Fastener 506 Genuine T50 3/8-Inch Staples, 1250-Pack	Office & School Supplies - Staplers & Punches - Staples
4	AT&T CL2940 Corded Phone with Caller ID/Call waiting, [...]	Office Electronics - Telephones & Accessories - Landline Phones - Corded Telephones
5	Pilot G2 Retractable Premium Gel Ink Roller Ball Pens Fine Pt (.7) Dozen Box Red [...]	Office & School Supplies - Writing & Correction Supplies - Pens & Refills - Rollerball Pens - Gel Ink Rollerball Pens

Table 8.12: Ranked recommendations generated for the user in Table 8.11 using Bert4Rec* and Flare. Both approaches fail to retrieve the correct item (T50 Staplers).

Recommendations using Flare, with the target category as a critique: *Office & School Supplies - Staplers & Punches*		
Rank	Title	Product Category
1	T50 Heavy Duty Staples [Set of 2]	Office & School Supplies - Staplers & Punches
2	Onotio Heavy Duty 100 Sheet High Capacity Office Desk Stapler [...]	Office & School Supplies - Staplers & Punches
3	Swingline Stapler, Optima 70, Desktop Stapler, 70 Sheet Capacity [...]	Office & School Supplies - Staplers & Punches
4	1 X Flat Clinch Staples Mini Box of 1000 by MAX No.10	Office & School Supplies - Staplers & Punches
5	Swingline Automatic Stapler, Breeze, 20 Sheet Capacity, Battery Powered	Office & School Supplies - Staplers & Punches

Table 8.13: Top 5 recommendations generated for the user in Table 8.11, using the category of the correct item as a critique instruction. The correct item is retrieved at rank 1, with all other items retrieved within the critique category. Titles shortened are denoted with [...]

ing with broad critiques, we find that all Flare-retrieved items fall within the top-level category specified by the respective critique. Similarly, we find that Flare can leverage the additional details in the precise critiquing string to retrieve only staples. This is particularly notable given that both critique strings were unseen during training, demonstrating Flare's semantic understanding of them.

8.2.7 Ablation Study

Next, we conducted ablation studies to demonstrate the impact of the components of our proposed model. The results are shown in Table 8.16, which were performed on the Clothing dataset using a Base model with 8 layers, 16 heads, 768 model dimensions, and 3072 hidden dimensions.

To evaluate the impact of the Perceiver model, we experiment with using the EOS token embedding directly from a text sentence. While we find that the effects of Perceiver are less than expected, it enables future applications involving multi-modal inputs. We also find that bidirectional masking and contrastive loss improve Flare's performance, with removing these features resulting in $\approx -8\%$ and $\approx -6\%$ drops in Recall@5, respectively. In particular, the largest decrease in MRR was observed by removing bidirectional masking,

Recommendations with alternate in-history category as critique: ***Forms, Recordkeeping & Money Handling - Key Cabinets & Holders***		
Rank	Title	Product Category
1	Spectrum Diversified Scroll Mail and Letter Organizer, Wall Mount, Black	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets & Holders
2	Lucky Line Designer Pattern Key Identifier Caps, 4 Pack (16304)	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets & Holders
3	KeyGuard SL-8548-U Dual Access Combination Key Cabinet [...]	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets & Holders
4	Master Lock Small Lock Box with 20 Key Capacity, 7131D, Beige	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets & Holders
5	Security Large Magnetic Hide-A-Key Holder for Over-Sized Keys [...]	Office & School Supplies - Forms, Recordkeeping & Money Handling - Key Cabinets & Holders
Recommendations with outside-history category as critique: ***PDAs, Handhelds & Accessories - PDA & Handheld Accessories***		
Rank	Title	Category
1	Kyocera Kona Black (Virgin Mobile)	Office Electronics - PDAs, Handhelds & Accessories - PDA & Handheld Accessories
2	iPad Pro 9.7 Case, Apple iPad Pro 9.7 Folio Case, [...] Black	Office Electronics - PDAs, Handhelds & Accessories - PDA & Handheld Accessories
3	Saunders 00497 RhinoSkin Aluminum Hardcase for Palm Tungsten TX/T5	Office Electronics - PDAs, Handhelds & Accessories - PDA & Handheld Accessories
4	ZAGG Folio Case with Backlit Bluetooth Keyboard [...] Purple-Black [...]	Office Electronics - PDAs, Handhelds & Accessories - PDA & Handheld Accessories
5	HQRP Replacement Battery Palm Tungsten E2 PDA + Screwdriver	Office Electronics - PDAs, Handhelds & Accessories - PDA & Handheld Accessories

Table 8.14: Top 5 recommendations generated for the user in table 8.11, using a product category from the user history (in-history critique) and using one outside the user's history (outside-history critique). Both commands are followed, and all retrieved items are within the target category, demonstrating Flare's robust ability to follow instructions. Titles shortened are denoted with [...].

Recommendations with the target item's broadened category as a critique: *Office & School Supplies*		
Rank	Title	Category
1	Canon CLI-8M Magenta Ink Tank	Office & School Supplies - Printer Ink & Toner - Inkjet Printer Ink
2	Swingline Stapler, Optima 25, Full Size Desktop Stapler, 25 Sheet Capacity	Office & School Supplies - Staplers & Punches - Stapler
3	Mead #10 Envelopes, Press-It Seal-It, White, 50/Box (75024)	Office & School Supplies - Envelopes, Mailers & Shipping Supplies - Envelopes
4	Columbian #10 Envelopes, Security Tinted,4-1/8" x 9-1/2", White, 500 [...]	Office & School Supplies - Envelopes, Mailers & Shipping Supplies - Envelopes
5	Five Star Filler Paper, Graph Ruled Paper, 100 Sheets, 11"x8'1/2" [...]	Office & School Supplies - Binders & Binding Systems - Binder Accessories
Recommendations with the target items's category made more specific: *Office & School Supplies - Staplers & Punches - Staples*		
Rank	Title	Category
1	Arrow Fastener 506 Genuine T50 3/8-Inch **Staples**, 1250-Pack	Office & School Supplies - Staplers & Punches
2	Swingline **Staples**, S.F. 4, Premium, 1/4" Length, 210/Strip, 5000/Box, 1 Box (35450)	Office & School Supplies - Staplers & Punches
3	Swingline **Staples**, Standard, 1/4" Length, 210/Strip, 5000/Box, 5 Pack	Office & School Supplies - Staplers & Punches
4	Swingline **Staples**, Standard, 1/4" Length, 210/Strip, 5000/Box, 1 Box	Office & School Supplies - Staplers & Punches
5	Arrow Fastener 224 Genuine P22 1/4-Inch **Staples**, 5,050-Pack	Office & School Supplies - Staplers & Punches

Table 8.15: Top 5 recommendations generated for the user in table 8.11, comparing using broadened categories and more precise categories as critiquing commands. Note that the precise critique was not seen during training, but Flare leveraged the more specific information to retrieve relevant items.

Method	Recall@5	Recall@10	nDCG@10	MRR
Flare	0.142	0.148	0.138	0.136
w/o Text	0.138	0.143	0.134	0.133
w/o Perceiver	0.140	0.145	0.135	0.133
w/o Bidirectional Masking	0.131	0.134	0.128	0.126
w/o Contrastive Loss	0.134	0.145	0.136	0.134
w/o Duplicates	0.140	0.146	0.136	0.134

Table 8.16: Flare ablation study using the Clothing dataset.

despite the evaluation always masking the last item. We suspect that bidirectional masking reinforces the model's ability to leverage signals across the sequence. Furthermore, we note that bidirectional masking can be a more demanding training task, as multiple items may be masked in a sequence rather than just the last item. Finally, we evaluate the impact of removing duplicates from the user's sequence and find that the model's performance does not decrease significantly. This provides confidence that the model's performance is unlikely to be attributable to copying previous items in the sequence.

8.3 Conclusion and Future Work

We introduce Flare, a hybrid recommender model that integrates item IDs with contextual text. In Bert4Rec*, we tune Bert4Rec and demonstrate that the tuned model outperforms previous Bert4Rec baselines and state-of-the-art models on two datasets. Flare shows additional performance gains over the new Bert4Rec* baseline, suggesting a benefit to hybrid recommenders. We also extend existing benchmarks by introducing a baseline for larger-item-count domains, where Flare outperforms the Bert4Rec* and Recformer baselines. We believe these results may serve as new baselines for future work.

Our results suggest that hybrid methods that incorporate both direct item representations and item contexts, such as text, are a promising approach for improving recommendation quality. Regarding text, one area for future work is to investigate the impact of different types of language models and their fine-tuning strategies on recommendation performance when used to encode text metadata. Another consideration is the substantial memory cost of ID-embedding tables at scale. Extending models to handle large vocabularies beyond the datasets used in this chapter may be considerably challenging. We plan to focus on the impact of different vocabulary sizes and to investigate new techniques for

handling larger item counts.

Finally, we propose using critiquing as an evaluation methodology to assess how well a model interprets and responds to textual critique strings. Though using the category feature in the Amazon Product Reviews dataset as a proxy for a user-supplied critique is arguably contrived, we view it as a modest step towards natural-language critiquing that might occur in a conversational recommender. We plan to expand the methodology to incorporate additional natural-language critiques in future work.

Part IV

Putting Everything into Context

Chapter 9

Common Bonds and Future Directions

In this chapter, we explore common directions for future work that apply to all of the proposed solutions introduced in this thesis. In particular, we focus on two common directions: the need for data that covers all required modalities of context (section 9.1) and the computational intensity of processing those modalities (section 9.2). In both directions, we propose potential paths uniquely enabled by our solutions' ability to tap into the macro-level context.

9.1 Addressing Data Sparsity through Pre-Training

A significant limitation of our proposed solutions is their reliance on fine-tuning to learn how to interpret their domain-specific macro-context (i.e. learning to capture context). In Discussion Transformers (chapter 4) and FedFormer (chapter 6), the ability to incorporate context is enabled by fine-tuning a macro-level transformer model. Similarly, in Flare (chapter 8), macro-level context is learned by refining item collaborative filtering embeddings across user sequences, ensuring that they are complementary with text-based content embeddings.

An inherent consequence of fine-tuning models to interpret context is that the availability of high-quality data is a limiting factor. For instance, in Discussion Transformers (chapter 4), we trained our models on human-annotated hate-speech examples. However,

social media platform owners are intrinsically motivated to remove such examples of abusive behaviour that appear on their platforms. This, in turn, causes a cat-and-mouse game in which high-quality labelled data is increasingly deleted, reducing the size of available datasets for training. Further, even if data is available, expert-level human annotation remains a time-consuming and expensive process. Datasets like our proposed HatefulDiscussions (section 4.1.3), with $\approx 27,000$ labelled discussions, are much smaller than those in other domains and exemplify this scarcity, potentially hindering the mDT's ability to learn to interpret complex conversational dynamics effectively.

Data scarcity also affects hybrid recommenders like Flare (chapter 8). Recommendation datasets are often sparse, with the number of observed user-item interactions per item being much smaller than the total set of possible interactions. Indeed, in Flare, we observed that our performance was weakest in the Scientific domain, containing $\approx$ 5 000 items and $\approx$ 10 000 users, compared with our best domain, Pets, containing $\approx$ 38 000 items and $\approx$ 47 569 users (section 8.2.2). We hypothesize that this drop in performance was due to insufficient training on unpopular items, leading to under-refined macro-context information when processing sequences containing those items (the "cold start" problem) [Li et al., 2023, Hou et al., 2018, Bobadilla et al., 2012].

Our insights across the three core chapters of this thesis (Discussion Transformers in chapter 4, FedFormer in chapter 6, and Flare in chapter 8) highlight the challenges of creating robust mechanisms to incorporate context in low-data environments. In this section, we explore the use of unsupervised pre-training (section 2.3.2) to address the common issue of data sparsity.

9.1.1 Pre-Training for Discussion Transformers

Background: Pre-training for Graph Neural Networks

Pre-training has been explored to enhance Graph Neural Networks (GNNs), with applications primarily in chemistry and recommender systems. While generalizable pre-training tasks for natural language models are somewhat intuitive (i.e. masked language modelling to learn language semantics), designing a task for Graph Neural Networks is more nebulous. Indeed, many early works on GNN pre-training exhibited negative transfer, in which pre-training harmed performance on the downstream task [Rosenstein et al., 2005]. A key theory of negative transfer is the misconception that semantic-domain similarity between pre-training and the downstream task implies transferability. For example, Hu et al. [2020a] found that pre-training by predicting general chemical properties using the

ChEMBL dataset [Zdrazil et al., 2024] ($\approx 456,000$ molecules) resulted in negative transfer on two different downstream tasks (Clintox and HIV from MoleculeNet [Wu et al., 2018]). Similarly, Donabauer and Kruschwitz [2024] attempted to pre-train a GNN for fake news detection by first predicting a node's number of retweets, but found that this pre-training task led to negative transfer.

Due to the risk of negative transfer, many recent works have focused on domain-agnostic generative and contrastive pre-training tasks, building on ideas from NLP (section 2.3.2). Generative tasks in GNNs task the model to reconstruct masked nodes or edges from the original graph, analogous to masked language modelling in NLP (section 2.3.2). Here, node masking requires the model to reconstruct masked nodes by minimizing an L2 loss between the reconstructed and original representations, akin to a variational autoencoder (VAE) Hu et al. [2020b], Kipf and Welling [2016]. Conversely, edge-level masking removes edges from the original graph and requires the model to classify whether two nodes are intended to be connected. Models that are successful at these tasks must learn to create compressed representations of nodes that capture essential structural and semantic properties. In other words, focus primarily on refining individual node and edge features, often learning expressive local structures [Wang et al., 2024].

On the other hand, contrastive tasks focus on refining graph-level skills by tasking the model to recognize whole-graph permutations that originate from the same input graph. A seminal paper on graph-level permutations is GraphCL [You et al., 2020], which introduces four types of graph permutations: node dropping, edge perturbation, attribute masking, and subgraph sampling. Node dropping randomly removes a fraction of nodes from the graph. Edge perturbation randomly adds or removes a fraction of edges. Attribute masking randomly masks a fraction of node attributes/features. Finally, subgraph sampling selects a subgraph from the original graph by performing a random walk. Applying two of these augmentations to the same graph yields two correlated views, treated as positive pairs. A GNN encoder is then trained to produce similar embeddings for these two views (the positive pair) and dissimilar embeddings for views generated from other graphs in the batch (negative pairs), typically using a contrastive loss function like InfoNCE (equation 2.9) [You et al., 2020]. As such, graph contrastive tasks can be viewed as analogous to next-sentence prediction tasks in NLP, with graph permutations replacing sentences.

Indeed, recent work by Wang et al. [2024] has demonstrated that generative and contrastive tasks are complementary, capturing different aspects of graph information. This has motivated hybrid pre-training strategies that combine local generative tasks and global contrastive tasks. In Wang et al. [2024], the authors incorporate both tasks by using two output-prediction networks atop the same GNN, one for each task, thereby enabling GC-MAE to learn rich local details and discriminative global representations simultaneously.

Ideas for Discussion Transformer Pre-training

Section 9.1.1 outlines many guiding principles for graph pre-training. In particular, we defined the risk for negative transfer in GNNs. Indeed, the spectre of negative transfer also looms over discussion transformers, as we encountered in our preliminary studies exploring pre-training on Reddit discussions to predict subreddit clusters. However, insights from domain-agnostic generative and contrastive graph pre-training tasks inspire us to envision potential pre-training tasks for Graph Transformers applied to social media discussions.

Starting with generative tasks, it would be possible to extend node- and edge-masking pre-training to discussion transformers. Edge-level masking would require a discussion transformer to classify whether a comment is a reply to another comment, thereby emphasizing conversational awareness. Conversely, node-level masking would require a discussion transformer to reconstruct masked comments within a discussion tree, leveraging the context of surrounding discussions. However, both tasks face domain-specific challenges in social media discussions. On edge-level "is a reply" classification, it is feasible that a given reply for one comment might be equally possible for another. For example, the reply "I agree" would apply to any comment expressing an opinion. A potential workaround for this issue is to retain all edges in the graph, thereby removing the edge-masking component from the original formulation [You et al., 2020]. This would simplify the task of classifying whether two comments are physically in response to each other in the input discussion graph. While this would be a trivial task using a graph neural network, as node representations are inherently a function of their neighbours [Veličković et al., 2017, Fey and Lenssen, 2019, Hamilton et al., 2017, Chen et al., 2018], the same cannot be said for graph transformers, which rely on learned positional encodings to encode structure [Ying et al., 2021]. As a result, the edge-level task would refine the discussion transformer positional encodings to capture the discussion's structure more expressively.

In node-level comment reconstruction tasks, it is vital to recognize the difference in information density between multimodal text embeddings and atom embeddings. Using RDKit, an industry-standard library for retrieving chemical properties and subsequently used by many works that implement node-masking pre-training for molecular GNNs, an atom is often described by its atomic number, chirality tag, and the type and direction of its bonds, totalling four features each with a constrained set of possible values [1]. Comments, on the other hand, are comprised of unbounded language and carry virtually limitless semantic possibilities, especially when you consider the possibility of multimodality with images. As

[1] We make careful note that while individual atoms have limited descriptors, molecules as a whole have many more descriptive features that more accurately describe the role of individual atoms. For the purposes of describing node reconstruction, we focus on comparing the features of individual atoms

a result, we hypothesize that reconstructing a dense comment representation would be many orders of magnitude more difficult than reconstructing a sparse atom embedding. That said, more robust and capable graph transformers may be able to reconstruct dense comment representations, which we leave for future work.

Next, we turn our attention to contrastive tasks. Likewise, we can define logical extensions of GraphCL [You et al., 2020] contrastive pre-training tasks for social discussion pre-training. To construct discussion permutations, we can trivially adopt node dropping and edge permutations from GraphCL to comments and reply connections, respectively. Because we use dense comment embeddings to seed node representations, attribute masking is unsuitable for our use case. Subgraph sampling also requires more creative adaptations to discussion transformers, since discussions are organized as hierarchical trees rather than rotation-invariant graphs. Rather than using a random walk to sample subgraphs, a more discussion-aware approach is to sample branches in the discussion stemming from the root post (i.e., different discussions that originate from the same common point). The root post could optionally be removed from each sampled branch to prevent the answer from leaking.

One additional factor in constructing a pre-training task is the data used for unsupervised training, which we believe also offers opportunities for creative, discussion-aware solutions. While one approach would be to capture all discussions on a social platform in a single dataset, akin to large-scale LLM pre-training [Radford et al., 2018, Raffel et al., 2020], we believe that careful data curation could also enable Discussion Transformers to learn diverse community norms. For instance, to succeed on discussion-level contrastive tasks, we hypothesize that Discussion Transformer models must understand the values and social norms of the community in which the conversation took place. Therefore, by ensuring a representative, diverse sample of discussions from a wide range of communities, we can build a final model that is sensitive to multiple viewpoints and reduces bias. For example, ensuring that representative conversations from both sides of the political spectrum are included during pre-training may enable the model to capture signals that help it comprehend political discourse. We believe that learning from a wide variety of community norms can, in turn, benefit downstream tasks, such as hate speech detection.

9.1.2 Pre-Training for Hybrid Recommenders

Background: Pre-training for Sequential Recommenders

Next, we explore applications of pre-training to enhance sequential recommender systems. Pre-training has been widely adopted in sequential recommender systems as a self-supervised learning paradigm to address data sparsity and the cold-start problem. Similar

to NLP pre-training, pre-training tasks for sequential recommenders can be broadly categorized as either generative, which learn to reconstruct user interactions, or contrastive, which learn to discriminate between similar and dissimilar views of user sequences.

Seminal generative tasks were directly inspired by the masked language modelling task (MLM) in NLP (section 2.3.2). An influential example is the Masked Item Prediction (MIP) task, introduced in BERT4Rec [Sun et al., 2019], which we also adapted in Flare (see chapter 8). This approach randomly masks a percentage of items in a user's interaction history and trains a bidirectional Transformer model to predict the original items using both past and future context. Note that this differs from strictly training to predict the next item for a user, as explored in SASRec [Kang and McAuley, 2018]. For example, given a user history consisting of items $[a_1, a_2, a_3, a_4]$, MIP could create training examples $[a_1, mask, a_3, a_4]$ and $[a_1, a_2, mask, a_4]$, and task the model to predict the masked item in each sequence. By learning to reconstruct the sequence from bidirectional context, the model develops a rich, contextualized understanding of item relationships, which it then uses to predict the future next item (a_5) during inference.

By contrast, contrastive tasks learn representations by discriminating between augmented views of user interaction sequences. Similar to GraphCL for GNNs [You et al., 2020], CL4SRec (Contrastive Learning for Sequential Recommendation) proposes applying stochastic data augmentations to create two correlated "views" of the same sequence [Xie et al., 2022]. Examples of stochastic data augmentations include random cropping, masking, or item reordering. The model is then trained with a contrastive loss, such as the InfoNCE loss (equation 2.9, Oord et al. [2018]), to pull the representations of positive views together in the embedding space while pushing them apart from negative pairs (i.e., augmented sequences from other users).

Perhaps most relevant to our work in Flare (chapter 8) is work that uses LLMs to guide both paradigms toward more semantically intelligent objectives. In the generative domain, the focus is shifting from reconstructing item IDs to predicting more semantically nuanced outputs. For example, Latent Relation Discovery (LRD) [Yang et al., 2024] moves away from predicting masked IDs and instead predicts latent, unobserved relationships between items. In essence, LRD aims to align the probability of relationship plausibility generated by an LLM from item text with that generated directly from item ID embeddings. Another relevant direction is P5 (Pretrain, Personalized Prompt & Predict Paradigm) [Geng et al., 2022], which formulates recommendation tasks as text-generation problems inspired by T5 [Raffel et al., 2020], which does the same for text classification. Through this formulation, the authors pretrain on a wide range of recommendation tasks, including rating prediction, review summarization, direct recommendation, and sequential recommendation [Geng et al., 2022].

Similarly, LLMs are being used to mitigate the semantic blindness of synthetic augmentations in contrastive tasks. Due to the in-batch nature of InfoNCE contrastive losses, the richness of the learning signals depends on the difficulty of the examples selected in the batch [Oord et al., 2018]. As a result, poor selection of negatives can degrade performance. To this end, SRA-CL leverages LLMs' semantic awareness to generate more challenging augmentations and contrastive pairs [Cui et al., 2025]. Semantic-aware augmentations include leveraging semantic similarity between items to select informative replacements and selecting users with semantically related histories for intra-user contrastive learning [Cui et al., 2025].

Ideas for Hybrid Recommender Pre-Training

The principles of pre-training in sequential recommendation offer a clear path to address the data scarcity challenges identified for Flare. As we hypothesized earlier in this section, the performance degradation observed in low-data domains is likely due to under-refined macro-level collaborative filtering embeddings for less popular items. By leveraging large, unlabeled corpora of user interaction data, we can pre-train Flare to be more robust before fine-tuning on a specific downstream domain. Indeed, most of the tasks explored in the previous section can be directly applied to Flare. For instance, it would be possible to apply sequence-level contrastive tasks, such as those proposed in CL4Srec and SRA-CL, to refine the robustness of macro-level ID embeddings [Cui et al., 2025, Xie et al., 2022].

Perhaps most interesting for future work would be to explore pre-training tasks that leverage Flare's unique ability to process user intent critique commands, in addition to collaborative filtering and content signals. That is, leveraging that items in Flare are represented as $item_i = (ID, text, critique)$ tuples, where the critique string is a semantic instruction linking $item_i \rightarrow item_{i+1}$. We hypothesize that tasks that manipulate critique strings can elicit more user-aware, semantically rich representations of items and better adherence to critique instructions. Towards this goal, we propose two pre-training tasks that utilize critique information, extending generative and contrastive tasks, respectively.

Starting with generative tasks, a potential application of generative pre-training is reconstructing critique strings rather than item IDs. Given a user's full interaction history and a target item the user interacted with, the model would be tasked with predicting or reconstructing the critique text that likely motivated that choice. For example, given a user history of "[laptop, mouse]" and a target item of webcam, the model would be trained to reconstruct the plausible critique "office supplies for video calls." Similar to LRD [Yang et al., 2024], this task moves beyond simple item-item relationships and requires the model to learn the latent semantic intent that links a user's macro-level history to a specific future

choice. This task could be formulated via text generation, as explored in REGEN [Sayana et al., 2025], or by having the model select from a set of possible critique strings.

Next, we can design a contrastive pre-training task that leverages critiques to create contrastive views. Specifically, we can use a pre-trained Large Language Model (LLM) to generate semantically aware substitutions for critique strings, constructing the positive and negative views of a user sequence required for the contrastive objective. For instance, a positive pair would consist of two views of the same user sequence with semantically equivalent yet textually different critiques, such as a paraphrase generated by an LLM (e.g., "cheap running shoes" becoming "affordable athletic footwear"). Conversely, hard negative pairs would be constructed using critiques that are semantically related but distinct (e.g., "expensive hiking boots") or antonyms. The contrastive objective would then train Flare to produce similar representations for the positive pairs while maximizing the distance to the negative pairs. Such a task would explicitly pre-train the model to be sensitive to the nuances of natural language, thereby improving its ability to differentiate between subtly different user requests during fine-tuning and inference.

9.2 Addressing Latency by Leveraging the Context of Context

In this thesis, we have argued that including additional macro-context can yield several benefits, such as reducing the risk of ambiguous and misinformed decisions. However, it is also essential to recognize that including auxiliary information increases the amount of data a model must process to make a decision. As a result, many of the methods proposed in this thesis incur additional latency compared to their non-contextual counterparts. For example, in Chapter 4, we proposed discussion transformers that add context by encoding all other comments in discussions, including their images, and aggregating them using a graph transformer model. Similarly, in chapter 6, we proposed aggregating q-value representations of federated agents using a transformer model.

Despite the promise of macro-context, the additional computational overhead may limit its adoption among platform owners and practitioners. In this section, we explore two possible directions for using metadata about context (i.e., the context of context) to accelerate model inference and reduce memory requirements. First, we explore the use of temporal metadata to enable causal attention and KV caching, with applications to Discussion Transformers (chapter 4) and Flare (chapter 8). Then, we explore applications of using pre-emptive context-based gating to reduce the input size of FedFormer's federation trans-

former (chapter 6), drawing inspiration from the mixture-of-experts architecture [Fedus et al., 2022].

9.2.1 Casual KV Caching Through Context Metadata

A primary challenge in deploying macro-context aware Transformer models is the high computational cost of bidirectional re-inference in dynamic, interactive scenarios. This issue is particularly salient in two domains explored in this thesis: the real-time moderation of online discussions (Discussion Transformers in chapter 4) and interactive recommendation sessions (Flare in chapter 8). For Discussion Transformers, every additional comment requires a full re-computation of the entire discussion graph. Similarly, for Flare, each user-critique iteration (e.g., a user browsing different product categories) would require recomputing their historical sequence to incorporate the query.

In both cases, the model's reliance on non-causal, all-to-all attention is a double-edged sword. On the one hand, all-to-all attention enables the model to continuously reassess prior predictions in light of new evidence, such as a community's post hoc reactions to a comment (e.g., replies), or to recontextualize items toward new intents, thereby motivating its use in this thesis. On the other hand, recomputation incurs increasingly prohibitive latency costs as the input sequence increases, making real-time computation infeasible at scale.

To address this challenge, we can draw inspiration from a similar problem solved in generative natural language processing. In Large Language Models (LLMs), text generation is performed auto-regressively, where tokens are generated one at a time [Vaswani et al., 2017]. This left-to-right nature of text generation is exploited to accelerate inference through causal attention and Key-Value (KV) caching. In a causal setup, a token can only attend to tokens that appeared before it. In the causal self-attention mechanism [Vaswani et al., 2017], each token produces a Query (Q), a Key (K), and a Value (V) vector. The current token's Query is used to score its relevance against the Keys of all previous tokens, and these scores are used to compute a weighted sum of the Values from those previous tokens, yielding a new token representation. However, since the Keys and Values of past tokens do not change when a new token is generated in a causal setup, they can be pre-computed and stored in a cache. Therefore, at each new step, the model only needs to compute the Q, K, and V vectors for the single new token and compare its Q against the entire history of cached Ks and Vs [Vaswani et al., 2017]. This avoids the costly re-computation of Keys and Values for the whole sequence at every step.

A promising direction for future work is to adapt this causal paradigm to our proposed

approaches by incorporating temporal information. This would involve modifying the attention mechanism to be strictly causal, such that each element attends only to elements created chronologically earlier. With this temporal constraint, KV caching can be applied directly. In discussion transformers, this would involve constraining attention such that each comment attends only to previous comments. Then, as new comments arrive, their representations can be computed efficiently by reusing the cached Key and Value tensors from prior comments, thereby drastically reducing inference latency. Similarly, in Flare, the Ks and Vs of the user sequence can be cached so that recomputation of new critique strings would only require updating the Qs for the latest item.

The causal approach, however, entails a significant trade-off: it would achieve much faster inference but lose the ability to perform holistic, post hoc analysis of future content, such as replies. An important avenue for future research will be to quantify this speed-versus-accuracy trade-off and, perhaps, explore hybrid architectures that leverage caching for initial, rapid assessments while periodically performing a complete, holistic re-evaluation.

9.2.2 Mixture-of-Federated Agents to Accelerate Federation Transformers

Another challenge facing macro-context-aware transformers is that the computational complexity of their attention mechanism scales quadratically with sequence length [Vaswani et al., 2017]. While manageable for smaller sequences, this $\mathcal{O}(n^2)$ complexity incurs significant latency as the number of embeddings, such as those containing macro-level context, increases. In particular, this bottleneck can limit FedFormer's applicability to scenarios involving hundreds or thousands of agents, such as large-scale IoT networks or fleets of autonomous vehicles. Each agent would need to compute attention scores against every other agent in the network at every time step, leading to prohibitive latency and computational costs as the federation grows.

To overcome this scaling challenge, a promising direction is to adapt the Mixture-of-Experts (MoE) paradigm, which has proven highly effective in scaling Large Language Models to trillions of parameters while maintaining computational efficiency [Fedus et al., 2022]. The core principle of MoE is to replace a single, dense network with a collection of smaller "expert" sub-networks and a lightweight "gating network" or router. For a given input, the gating network dynamically selects a sparse subset of the most relevant experts (typically the top one or two) to process it, based on the input itself [Fedus et al., 2022]. This decouples the model's total size from its per-inference computational cost, since only

a small fraction of its parameters are activated for any given input. In practice, MoE is applied per block to decompose the computationally expensive feedforward layer into a set of smaller expert layers [Fedus et al., 2022].

We propose adapting this paradigm to FedFormer by re-framing the inter-agent federation process as an expert selection problem. In this new architecture, each agent in the federation would be treated as an "expert" on its own local environment. A new, small gating network would be added to each agent. At each time step, an agent would first compute an embedding of its current state or observation (the initial q-value embedding). This embedding would then be passed to the local gating network, which would output a set of scores indicating the predicted relevance of each other agent in the federation to the agent's specific context. Instead of attending to all other agents, the agent would attend only to the sparse set of top-k agents selected by its gating network. Note that our application differs from standard MoE, which is applied on a per-layer basis, by instead gating at the input level [Fedus et al., 2022].

This MoE-based approach to federation would yield two benefits. First, it would break the quadratic scaling barrier, enabling FedFormer to scale to large federations with thousands of agents while maintaining a low, constant per-agent computational cost. Second, it could lead to superior performance by encouraging specialization and reducing noise. The gating network could learn sophisticated, context-dependent routing policies, for instance, learning that an agent navigating a dense urban environment should primarily consult other "city experts" while completely ignoring irrelevant information from "highway experts." This would create a more dynamic, efficient, and intelligent federation in which agents selectively collaborate with the most relevant peers. A key research challenge for this new paradigm is developing effective methods for training the input-level gating network, for instance, to ensure that the gating network captures insights from new agents. Still, its potential to enable truly scalable, adaptive federated intelligence makes it a critical direction for future work.

Chapter 10

Conclusion

This thesis began with a simple premise: that the ability to make sound judgments is closely tied to understanding context. Consider the meaning of the word "fire". Whether it is an alarm, a command, or a termination notice can only be determined by considering its surrounding context. We argued that the ability for AI models to consider macro-level context poses a fundamental challenge, particularly when making complex decisions in context-rich environments. This led to the guiding theory of this thesis, that the key to designing more robust and adaptive AI systems lies in their ability to understand this underutilized macro-level context. We demonstrated the potential for macro-level context by successfully designing novel transformer-based architectures for three context-rich environments: social media analysis (part I), federated learning (part II), and sequential recommender systems (part III).

In this chapter, we will first synthesize the key contributions and findings that support our central thesis, connecting the common threads that run through our work. We will then discuss the broader implications of this contextual paradigm, acknowledge the limitations of our research, and finally, offer a vision for the future of context-aware AI.

10.1 Key Findings and Contributions of this Thesis

The key contributions of this thesis primarily focus on three thematic lenses: a new paradigm for contextual modelling, architectural innovations with transformers, and critical methodological innovations to enable the research community.

10.1.1 New Paradigm for Contextual Modelling

A core contribution of this thesis was a new perspective on how context is defined and modelled across domains. At the outset of this thesis, we detailed how context is often viewed through a local lens. That is, the unit of context to be aggregated is defined as the words within a document or the pixels within an image. To broaden this lens, we proposed redefining the unit of context within a macro-context perspective for each domain we studied.

Starting in Discussion Transformers (chapter 4), our key insight was to extend the unit of context beyond the content of an individual comment (micro) to the entire discussion graph (macro). In turn, this enabled a more holistic understanding of conversational dynamics. Our insights are guided by recognizing that the meaning of a comment is deeply rooted in the context in which it was uttered. For instance, a comment such as "Ew, that's gross" can be benign in response to unsavoury-looking food, or hateful in response to protected characteristics such as sexuality. By defining context on the macro conversational level, we enable analysis that can disambiguate between these two cases, reducing the risk of ambiguous decision-making. Further, discussion-level analysis opens the door to incorporating social norms, as the model can learn how conversation dynamics shift across communities, a capability impossible under the previous micro comment-only paradigm.

Next, in FedFormer (chapter 6), we refactored the unit of context from blind averaging of model weights (FedAvg, McMahan et al. [2017]) to a context-aware, selective federation approach. We define context as the macro-level relationships among heterogeneous agents. By learning these relationships, we enable context-dependent federation, in which an agent can access the insights of other agents based on their current relevance. For instance, in the hypothetical deployment of FedFormer in self-driving cars, a car agent can learn which other agents have insights most relevant to the current environment, such as highway or rural driving experts. In turn, we enabled faster convergence, particularly during the onboarding of new agents to the federation network (section 6.2.4).

Finally, in Flare (chapter 8), our key insight was to extend the unit of context beyond local textual semantics to include the macro-context of explicit user intent (via critique instructions) and global item representations via collaborative-filtering embeddings. By bridging local and global semantics, we enable user agency by directly steering the recommendation process based on their current intent without sacrificing recommendation quality (section 8.2.4).

10.1.2 New Context-Aware Architectures based on the Transformer

Our paradigm shift toward incorporating macro-context was enabled by innovations that adapted the Transformer architecture. In particular, we leveraged the attention mechanism at the heart of Transformers to adaptively incorporate insights from macro-level sources. We did this through our non-trivial innovations, building on Graph Transformers [Ying et al., 2021] and Perceiver Transformers [Jaegle et al., 2021], as well as our novel application of transformers as a contextual federation mechanism.

A core contribution of Discussion Transformers (chapter 4) was our novel application of Graph Transformers [Ying et al., 2021], which had previously been used in chemistry, to model the hierarchical structure of conversations. With Discussion Transformers, we aimed to incorporate not only the content of other comments but also their organization. Our key insight for enabling this approach was recognizing the conceptual similarities between molecules and conversations, where comments and their connections can be represented as atoms and bonds. This framework led us to adapt the expressive power of Graph Transformers from modelling molecular relationships between atoms to modelling semantic relationships between comments. We introduced hierarchical positional encodings (section 4.1.2) and rich multimodal node representations (section 4.1.2) to better adapt this molecular framework to discussions.

Next, our work on FedFormer (chapter 6) introduced a new use case for the Transformer's attention mechanism, repurposing it as a dynamic, contextual federation mechanism. The central weakness of traditional Federated Averaging that FedFormer addresses is its static, context-blind aggregation. This limitation is particularly detrimental in heterogeneous environments, where not all participating agents are equally relevant to one another at all times. In such settings, simple averaging of models can yield a suboptimal, one-size-fits-all agent that performs worse than specialized agents would have individually. To address this, we adapt the Transformer architecture by treating the federated learning process as a contextual aggregation problem (Section 6.1). In our framework, the current agent's state serves as the Query, while the other agents' federated interpretations of that same state serve as the Keys and Values. The attention mechanism then computes a new state encoding as a weighted sum of the other agents' insights, with the weights dynamically determined by the relevance of each peer's "Key" to the current agent's "Query." This transforms federation from a simple averaging process into a sophisticated, on-the-fly contextual aggregation, enabling each agent to selectively draw knowledge from the most relevant peers for its immediate context.

Finally, in Flare (Chapter 8), we addressed the architectural challenge of fusing multi-

ple modalities, each representing a different source of context, within a hybrid sequential recommender. The goal was to integrate two fundamentally different types of information: the sparse, high-cardinality embeddings from collaborative filtering and the dense, semantic embeddings from textual context. Notably, the textual context encompasses not only static item descriptions but also dynamic, user-provided critiques, which represent an explicit form of macro-level intent (section 8.2.4). All of this had to be achieved while preserving the sequential nature of the user's interaction history. Recognizing that naive methods such as simple addition or concatenation are suboptimal because they risk losing the fine-grained intricacies of each modality, our architectural innovation was to adapt the Perceiver network as a sophisticated fusion module (section 8.1.2). The Perceiver is uniquely suited for this task, as it is designed to distill large, multimodal inputs into a learnable latent representation [Jaegle et al., 2021]. We use this mechanism as a learned information bottleneck, forcing the model to synthesize distinct collaborative and semantic signals into a single, unified, and context-aware representation for each item in the sequence.

10.1.3 Methodological and Empirical Contributions

Beyond proposing novel architectures, this thesis made a series of methodological and empirical contributions, providing new datasets, baselines, and evaluation paradigms to enable further exploration of macro-context models. These contributions aim to bridge the methodological gap between non-contextual and macro-contextual approaches and to offer avenues for evaluating the impact of context.

A foundational contribution of our study of online discourse was the curation of the HatefulDiscussions dataset (section 4.1.3). Before this work, hate speech detection research was predominantly constrained by datasets that consisted almost exclusively of isolated, text-only comments. This data scarcity effectively rendered the larger macro-context of conversational structure and visual media invisible to researchers. By aggregating thousands of complete, multimodal discussion trees, HatefulDiscussions provides a robust testbed for holistic analysis of abusive behaviours. Notably, our curation of HatefulDiscussion includes not only a broad spectrum of hateful behaviour (ranging from person-directed abuse to slurs to affiliation abuse), but also instances of conduct where context is necessary for disambiguation (table 4.2). This can be best seen in our incorporation of non-derogatory slurs, historically hateful slurs which have been re-appropriated by minority groups, which often lead to false positives by text-only comment models (as seen through our extensive qualitative study in section 3.5). We believe that HatefulDiscussions can not only advance Discussion Transformers but also do so equitably, covering

the diverse spectrum of hateful behaviour.

In recommender systems, we conducted a rigorous re-evaluation of ID-only baselines to investigate their performance at larger model scales. Our study was motivated by the recent trend towards text-only architectures, which often leverage Large Language Models (LLMs) and forgo collaborative filtering embeddings entirely. However, we observed a growing disparity in model capacity between these two approaches. Namely, we observe that as the scale of LLMs continues to increase (often reaching multiple billion parameters), these text-only methods now contain several orders of magnitude more parameters than the diminutive ID-only baselines used to justify their performance. This motivated our contribution of Bert4Rec*, a thoroughly tuned and scaled version of the Bert4Rec architecture [Sun et al., 2019], designed to be more comparable in size to contemporary models. Our empirical findings demonstrate that ID-only model performance improves dramatically as model capacity increases, achieving significantly higher performance than previous ID-only benchmarks and even establishing competitive results with modern text-only baselines (section 8.2.2). Beyond motivating our work with Flare, our contribution of Bert4Rec* can serve as a robust ID-only baseline for future work, enabling researchers exploring hybrid recommenders to perform a more principled ablation study measuring the actual additive value of incorporating text.

Another significant contribution of this thesis is the establishment of critiquing as a new evaluation paradigm explicitly designed for hybrid recommenders (section 8.2.4). In developing Flare, we incorporated macro-level user intent as a distinct input modality, with the explicit goal of enabling user-guided recommendations. However, validating this capability revealed the limitations of standard retrieval metrics such as recall and precision. These metrics are inherently rigid, evaluating success solely on whether a predicted item exactly matches the ground truth next item in a sequence. Consequently, they fail to account for semantic proximity (e.g., red vs. blue Adidas shoes) or to verify whether the model effectively used the provided intent, rather than merely extrapolating from other features. To address this methodological gap, we proposed permuted critiquing as a formal evaluation task, quantified through our novel Cat-nDCG metric (section 8.2.4). At a high level, this approach leverages the hierarchical category structure of the Amazon Product Reviews dataset to generate permuted user intent strings. We then adapted the standard nDCG formulation to define relevance not as a binary ground-truth match, but as a graded function of the overlap between a retrieved item's categories and the specific intent string (ranging from 1 to 4). The result is a nuanced evaluation framework that shifts the focus from passive prediction to active adherence, offering a robust measure of a model's capacity to understand and act upon the macro-context of explicit, dynamic user intent.

10.2 Limitations

While this thesis presents a significant conceptual and architectural step towards macro-context aware models, it is essential to acknowledge the limitations of our current work. These limitations do not undermine the findings but rather illuminate clear and necessary pathways for future research.

A primary limitation of the architectures proposed in this thesis is computational scalability. The non-causal, all-to-all attention mechanisms central to Discussion Transformers, FedFormer, and Flare incur a computational cost that scales quadratically with the size of the context window (i.e., the number of comments, agents, or items) [Vaswani et al., 2017]. While effective for the scale of data examined in our experiments, this complexity creates a bottleneck for real-time deployment in massive-scale systems. To address this, future work must focus on efficiency. As discussed in section 9.2.2, we propose adapting the gating mechanism of Mixture-of-Experts (MoE) to FedFormer to break the quadratic scaling barrier in large agent networks [Fedus et al., 2022]. Similarly, for time-sensitive applications like real-time discussion moderation and interactive hybrid recommendation, implementing causal attention to enable Key-Value (KV) caching for Discussion Transformers and Flare offers a promising avenue to achieve low-latency inference without sacrificing the benefits of Transformer-based modelling (section 9.2.1). We note that context in these domains is intrinsically temporal, offering a natural avenue for causal masking grounded in the contextual information it uses.

Second, our approach to modelling macro-context is heavily dependent on data availability and quality. While our models have demonstrated effective performance, their potential is currently constrained by the reliance on supervised learning to capture context. This necessitates labelled datasets that explicitly contain the necessary macro-contextual elements, such as complete discussion trees or aligned user-critique pairs. This requirement becomes a significant bottleneck in data-sparse domains such as hate speech detection, where high-quality datasets that include all requisite modalities—text, images, and conversational structure—are rare and, due to platform moderation, increasingly shrinking. To address this, section 9.1 outlined a vision for extending unsupervised pre-training to effectively learn macro-context before fine-tuning. By employing a general framework of complementary generative and contrastive tasks, models can learn how to understand these complex contextual structures without relying solely on scarce labelled data. Furthermore, we emphasize the potential of this pre-training paradigm to capture a diverse range of behaviours and norms, extending beyond the narrow distribution often present in task-specific fine-tuning data. By carefully curating diverse pre-training curricula, we can create models that are not only more robust to data sparsity in downstream tasks but also

more equitable and representative of diverse user populations.

Third, although we validated our models on rigorous and challenging benchmarks, the question of generalizability across diverse platforms remains an open frontier. Our architectures were optimized and evaluated within specific structural contexts: the branching discussion trees of Reddit, the e-commerce domain of Amazon [Ni et al., 2019], and the robotic manipulation tasks of the Meta-World benchmark [Yu et al., 2020]. We note that while our proposed architectures for modelling macro-context are transferable, their successful application to other domains may require additional innovations to address unique environmental characteristics. For instance, social discourse manifests in varied shapes, from the linear, short-form nature of X (formerly Twitter) to the visual-first interactions of TikTok, each presenting distinct challenges and interaction patterns. Similarly, agentic federation in long-horizon domains, such as autonomous driving, introduces temporal complexities that differ significantly from those in episodic manipulation and may require specific adaptations to the FedFormer architecture. Future work should therefore extend the macro-context paradigm to these domains by exploring the domain-specific innovations required to capture their respective contextual nuances effectively.

Finally, we acknowledge the gap between the specific macro-context we explored and the sheer complexity of the real world. Our implementation of "critiquing" in Flare, for instance, relied on structured category data (e.g., "Office Products - Printers") to simulate user-provided instructions. While this served as a valid proxy for establishing the methodology and enabled our instructional-following evaluation paradigm, natural language human intent is often far messier, nuanced, and contradictory. Users may not know precisely what they want, or they may express it in vague language. Future research must move beyond structured proxies to handle truly open-ended natural language interactions, developing systems capable of engaging in a dialogue to clarify ambiguous intent [Hebert et al., 2024b, Sayana et al., 2025].

10.3 The Bigger Picture

Given the increasing impact of Artificial Intelligence, it is crucial not only to conduct research that advances our field but also to invest in understanding the benefits our models can deliver to individuals. In Chapter 4, we discussed how this work could improve users' lives on social networks and other online discussion boards. We also see potential for social good in FedFormer (Chapter 6), which enables contextual yet privacy-preserving AI systems, and in Flare (Chapter 8), which empowers users to express their intent to recommender systems.

For all of these applications, we believe they offer guidance for platform owners to adjust their systems so that these benefits can be realized. For instance, we are encouraged that our work on discussion transformers has been recognized in research led by Reddit on enabling community-driven models [Kairam et al., 2024]. Our work also influences organizations that use federated learning, guiding them in building robust, data-hungry systems without sharing private data. Perhaps the most exciting prospect lies in translating the principles of this thesis to high-stakes domains like personalized precision medicine [Ashley, 2016]. In this context, the macro-context lens is vital. Rather than analyzing isolated diagnostics, our architectures could be adapted to synthesize a comprehensive picture of a patient, including medical imaging [Carbone et al., 2025], laboratory results, and clinical notes, into a holistic and actionable representation of patient health.

In our work, we also note our dedication to designing effective datasets and codebases[1] for future model validation. We are also committed to ensuring that the broader research community can benefit from these resources. An entire thread of our work to date has involved collaborating with undergraduate research assistants to continuously refine and extend the datasets we have developed, ensuring they are sufficiently broad and representative of online behaviours (as discussed in section 9.1.1). These datasets are publicly available for use by the broader research community. In addition to datasets, extensive work has been done to refine and document the codebases for the topics discussed in this thesis. We believe that these efforts can enable future work adapting our proposed contextual paradigms.

10.4 Concluding Thoughts

In this thesis, we sought to develop AI systems that can perceive the "forest" alongside the "trees", to understand the broader conversation, capture relationships between agents and incorporate user intent. Towards this goal, we focused on the Transformer architecture. We asked ourselves, "*Can we effectively ground Transformer architectures to make decisions using broad macro-level context?*" We conclude that this thesis demonstrated the potential for this paradigm in the affirmative, enabled by our novel Transformer architectures such as discussion-aware Graph Transformers and user intent Perceiver networks.

Looking ahead, we believe this direction is vital to the future of artificial intelligence. Future AI models will not be defined by their ability to process data, but by their capacity

[1]The codebase and datasets for mDT (https://github.com/liamhebert/MDT-2) and FedFormer (https://github.com/liamhebert/FedFormer) are open-sourced. Due to proprietary code, Flare cannot be open-sourced.

to develop a deeper understanding of the context in which that data exists. This thesis represents a step towards achieving this goal, in which AI systems are not merely predictive but also perceptive, adaptive, and truly context-aware.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 2022.

Euan A Ashley. Towards precision medicine. *Nature Reviews Genetics*, 17(9):507–522, 2016.

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. Affective and contextual embedding for sarcasm detection. In *Proceedings of the 28th international conference on computational linguistics*, 2020.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*, 2015.

Krisztian Balog, Filip Radlinski, and Alexandros Karatzoglou. On interpretation and measurement of soft attributes for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fon, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael Bronstein, Amra Delić, et al. Privacy-aware recommender systems challenge on twitter's home timeline. *arXiv*, 2020.

Melissa Block. The clear and present danger of trump's enduring 'big lie'. *NPR*, 2021.

JesúS Bobadilla, Fernando Ortega, Antonio Hernando, and Jesús Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, 2012.

Eloi Brassard-Gourdeau and Richard Khoury. Using sentiment information for preemptive detection of harmful comments in online conversations. *Canadian Conference on Artificial Intelligence*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020.

Fanyu Bu and Xin Wang. A smart agriculture iot system based on deep reinforcement learning. *Future Generation Computer Systems*, 2019.

Robin D Burke, Kristian J Hammond, and Benjamin C Young. Knowledge-based navigation of complex information spaces. In *Proceedings of the national conference on artificial intelligence*, 1996.

Rui Cao, Roy Ka-Wei Lee, Wen-Haw Chong, and Jing Jiang. Prompting for multimodal hateful meme classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 321–332. Association for Computational Linguistics, 2022.

Kathryn Carbone, Liam Hebert, Robin Cohen, and Lukasz Golab. Visual medical entity linking with velcro. In *Machine Learning for Health 2025*, 2025.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25. Association for Computational Linguistics, 2021.

Chanapa Channarong, Chawisa Paosirikul, Saranya Maneeroj, and Atsuhiro Takasu. Hybridbert4rec: a hybrid (content-based filtering and collaborative filtering) recommender system based on bert. *IEEE Access*, 2022.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018.

Li Chen and Pearl Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 2012.

Weilong Chen, Chenghao Huang, Weimin Yuan, Xiaolu Chen, Wenhao Hu, Xinran Zhang, and Yanru Zhang. Title-and-tag contrastive vision-and-language transformer for social media popularity prediction. In *Proceedings of the 30th ACM International Conference on Multimedia*, page 7008–7012. Association for Computing Machinery, 2022a.

Yiding Chen, Xuezhou Zhang, Kaiqing Zhang, Mengdi Wang, and Xiaojin Zhu. Byzantine-robust online and offline distributed reinforcement learning. *arXiv preprint arXiv:2206.00165*, 2022b.

Jianfeng Chi, Ujjwal Karn, Hongyuan Zhan, Eric Smith, Javier Rando, Yiming Zhang, Kate Plawiak, Zacharie Delpierre Coudert, Kartikeya Upasani, and Mahesh Pasupuleti. Llama guard 3 vision: Safeguarding human-ai image understanding conversations. *arXiv preprint arXiv:2411.10414*, 2024.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Yunfei Chu, Zhinong Wei, Xicheng Fang, Sheng Chen, and Yizhou Zhou. A multiagent federated reinforcement learning approach for plug-in electric vehicle fleet charging coordination in a residential community. *IEEE Access*, 2022.

Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. The echo chamber effect on social media. *National Academy of Sciences*, 2021.

Garage Contributors. Garage: A toolkit for reproducible reinforcement learning research, 2019.

Ziqiang Cui, Yunpeng Weng, Xing Tang, Xiaokun Zhang, Dugang Liu, Shiwei Li, Peiyang Liu, Bowei He, Weihong Luo, Xiuqiang He, et al. Semantic retrieval augmented contrastive learning for sequential recommendation. *arXiv preprint arXiv:2503.04162*, 2025.

Mithun Das, Binny Mathew, Punyajoy Saha, Pawan Goyal, and Animesh Mukherjee. Hate speech in online social media. *SIGWEB Newsl.*, 2020.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. Data bootstrapping approaches to improve low resource abusive language detection for indic languages. *arXiv*, 2022.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019.

Gregor Donabauer and Udo Kruschwitz. Challenges in pre-training graph neural networks for context-based fake news detection: An evaluation of current strategies and resource limitations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024.

Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, 2017.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 1990.

Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. Fault-tolerant federated reinforcement learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 2021.

Xiuwen Fang and Mang Ye. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10072–10081, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv*, 2019.

Jessica N Fish, Lauren B McInroy, Megan S Paceley, Natasha D Williams, Sara Henderson, Deborah S Levine, and Rachel N Edsall. "i'm kinda stuck at home with unsupportive parents right now": Lgbtq youths' experiences with covid-19 and the importance of online support. *Journal of Adolescent Health*, 2020.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 2016.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, Jun. 2018.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 2018.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten De Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI open*, 2021.

Dashan Gao, Xin Yao, and Qiang Yang. A survey on heterogeneous federated learning. *arXiv preprint arXiv:2210.04505*, 2022.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, 2022.

Hongyu Gong, Alberto Valido, Katherine M Ingram, Giulia Fanti, Suma Bhat, and Dorothy L Espelage. Abusive language detection in heterogeneous contexts: Dataset collection and the role of supervised attention. In *Proceedings of the AAAI Conference on Artificial Intelligence (AISI Track)*, 2021.

Peter Grasch, Alexander Felfernig, and Florian Reinfrank. Recomment: Towards critiquing-based recommendation with speech interaction. In *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 2018.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 2017.

Yanlin Han and Piotr Gmytrasiewicz. Ipomdp-net: A deep neural network for partially observable multi-agent planning using interactive pomdps. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019.

Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 2017.

Liam Hebert, Elizabeth Eddy, Will Harrington, Lauryn Marchand, Jason d'Eon, and Sageev Oore. Artbeat–deep convolutional networks for emotional inference to enhance art with music. In *Companion Publication of the 2021 International Conference on Multimodal Interaction*, 2021.

Liam Hebert, Lukasz Golab, and Robin Cohen. Predicting hateful discussions on reddit using graph transformer networks and communal context. In *2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2022a.

Liam Hebert, Raheleh Makki, Shubhanshu Mishra, Hamidreza Saghir, Anusha Kamath, and Yuval Merhav. Robust candidate generation for entity linking on short social media texts. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022) at COLING*, 2022b.

Liam Hebert, Hong Yi Chen, Robin Cohen, and Lukasz Golab. Qualitative analysis of a graph transformer approach to addressing hate speech: Adapting to dynamically changing content. *Workshop for AI for Social Good at AAAI*, 2023a.

Liam Hebert, Lukasz Golab, Pascal Poupart, and Robin Cohen. Fedformer: Contextual federation with attention in reinforcement learning. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023b.

Liam Hebert, Gaurav Sahu, Yuxuan Guo, Nanda Kishore Sreenivas, Lukasz Golab, and Robin Cohen. Multi-modal discussion transformer: Integrating text, images and graph

transformers to detect hate speech on social media. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024a.

Liam Hebert, Krishna Sayana, Ambarish Jash, Alexandros Karatzoglou, Sukhdeep Sodhi, Sumanth Doddapaneni, Yanli Cai, and Dima Kuzmin. Persoma: Personalized soft prompt adapter architecture for personalized language prompting. In *Workshop on Generative AI for Recommender Systems and Personalization at KDD*, Aug 2024b.

Liam Hebert, Marialena Kyriakidi, Hubert Pham, Krishna Sayana, James Pine, Sukhdeep Sodhi, and Ambarish Jash. Flare: Fusing language models and collaborative architectures for recommender enhancement. In *Workshop for Large Language Model Using Multi-modal Data for User Modeling at WWW*, Apr 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations*, 2016a.

Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 2016b.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Lei Hou, Xue Pan, and Kecheng Liu. Balancing the popularity bias of object similarities for personalised recommendation. *The European Physical Journal B*, 2018.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022.

W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020a.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 2008.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020b.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. How to index item ids for recommendation foundation models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 2023.

Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019.

Qianjia Huang, Diana Inkpen, Jianhong Zhang, and David Van Bruwaene. Cyberbullying intervention based on convolutional neural networks. In *First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics, 2018.

Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10143–10153, 2022.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2019.

Herbert Jaeger. The "echo state"' approach to analysing and training recurrent neural networks: Technical report. *German National Research Center for Information Technology GMD Technical Report*, 2001.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*. PMLR, 2021.

Rajkumar S Jagdale, Vishal S Shirsat, and Sachin N Deshmukh. Sentiment analysis on product reviews using machine learning techniques. *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*, 2019.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 2021.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 2002.

Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum*. ACM New York, NY, USA, 2017.

Michael I Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 1986.

Sanjay Kairam, Michael S Bernstein, Amy S Bruckman, Stevie Chancellor, Eshwar Chandrasekharan, Munmun De Choudhury, Casey Fiesler, Hanlin Li, Nicholas Proferes, Manoel Horta Ribeiro, et al. Community-driven models for research on social platforms. In *Companion Publication of the 2024 Conference on Computer-Supported Cooperative Work and Social Computing*, 2024.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes, 2020.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. ISSN: 2640-3498.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 1999.

Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

Jana Kurrek, Haji Mohammad Saleem, and Derek Ruths. Towards a comprehensive taxonomy and large-scale annotated corpus for online slur usage. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 138–149. Association for Computational Linguistics, 2020.

Kevin J Lang, Alex H Waibel, and Geoffrey E Hinton. A time-delay neural network architecture for isolated word recognition. *Neural networks*, 1990.

Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian McAuley. Coarse-to-fine sparse sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Yaoyiran Li, Xiang Zhai, Moustafa Alzantot, Keyi Yu, Ivan Vulić, Anna Korhonen, and Mohamed Hammad. Calrec: Contrastive alignment of generative llms for sequential recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024.

Bin Liang, Chenwei Lou, Xiang Li, Min Yang, Lin Gui, Yulan He, Wenjie Pei, and Ruifeng Xu. Multi-modal sarcasm detection via cross-modal graph convolutional network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1767–1777. Association for Computational Linguistics, 2022.

Hyun-Kyo Lim, Ju-Bong Kim, Joo-Seong Heo, and Youn-Hee Han. Federated reinforcement learning for training control policies on multiple iot devices. *Sensors*, 20(5), 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 2, pages 13–23. Curran Associates Inc., 2019.

Leanna Lucero. Safe spaces in online places: social media and lgbtq youth. *Multicultural Education Review*, 9(2):117–128, 2017.

Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. Latent linear critiquing for conversational recommender systems. In *Proceedings of The Web Conference 2020*, 2020a.

Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. Deep critiquing for vae-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020b.

Hanjia Lyu and Jiebo Luo. Understanding political polarization via jointly modeling users, connections and multimodal contents on heterogeneous graphs. In *Proceedings of the 30th ACM International Conference on Multimedia*. Association for Computing Machinery, 2022.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. Modelling the dynamic joint policy of teammates with attention multi-agent ddpg. *arXiv preprint arXiv:1811.07029*, 2018.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Meta. Meta reports first quarter 2023 results. Meta Investor Relations., 2023.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. Abusive Language Detection with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2145–2150. Association for Computational Linguistics, 2019.

Shubhanshu Mishra, Aman Saini, Raheleh Makki, Sneha Mehta, Aria Haghighi, and Ali Mollahosseini. Tweetnerd-end to end entity linking benchmark for tweets. *Advances in Neural Information Processing Systems*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mehdi Mohammadi, Ala Al-Fuqaha, Mohsen Guizani, and Jun-Seok Oh. Semisupervised deep reinforcement learning in support of iot and smart city services. *IEEE Internet of Things Journal*, 2017.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 2022.

Niklas Muennighoff. Vilio: State-of-the-art visio-linguistic models applied to hateful memes. *arXiv preprint arXiv:2012.07788*, 2020.

Chetan Nadiger, Anil Kumar, and Sherine Abdelhak. Federated reinforcement learning for fast personalization. In *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2019.

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14200–14213. Curran Associates, Inc., 2021.

Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. Disentangling preference representations for recommendation critiquing with ß-vae. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-llm: Efficient llm contextualization with user embeddings. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alexandre Parmentier and Robin Cohen. Learning user reputation on reddit. In *Web Intelligence (WI)*, pages 242–247, 2019.

Alexandre Parmentier, Jason P'ng, Xiang Tan, and Robin Cohen. Learning reddit user reputation using graphical attention networks. In *Future Technologies Conference (FTC) 2020*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 2019.

Paloma Piot, Patricia Martín-Rodilla, and Javier Parapar. Metahate: A dataset for unifying efforts on hate speech detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2024.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart Van Merrienboer, Kyunghyun Cho, and Yoshua Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. *arXiv preprint arXiv:1409.1257*, 2014.

Pearl Pu and Boi Faltings. Decision tradeoff using example-critiquing and constraint programming. *Constraints*, 2004.

Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. Federated reinforcement learning: Techniques, applications, and open challenges. *arXiv preprint arXiv:2108.11887*, 2021.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764. Association for Computational Linguistics, 2019.

Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. U-bert: Pre-training user representations for improved recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 2020.

RAIL-Berkeley and RLKit contributors. Rlkit: Reinforcement learning framework and algorithms implemented in pytorch, 2021.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 2023.

Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. Context and attribute-aware sequential recommendation via cross-attention. In *Proceedings of the 16th ACM conference on recommender systems*, 2022.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Jianji Ren, Haichao Wang, Tingting Hou, Shuai Zheng, and Chaosheng Tang. Federated learning-based computation offloading optimization in edge computing-supported internet of things. *IEEE Access*, pages 69194–69201, 2019.

Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *Neural Information Processing Systems*, 2005.

Gaurav Sahu and Olga Vechtomova. Adaptive fusion techniques for multimodal data. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

Gaurav Sahu, Robin Cohen, and Olga Vechtomova. Towards a multi-agent system for online hate speech detection. *Second Workshop on Autonomous Agents for Social Good (AASG), AAMAS, 2021*, 2021.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 29(19): 70–76, 2017.

Krishna Sayana, Raghavendra Vasudeva, Yuri Vasilevski, Kun Su, Liam Hebert, James Pine, Hubert Pham, Ambarish Jash, and Sukhdeep Sodhi. Beyond retrieval: Generating

narratives in conversational recommender systems. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

Rachel M. Schmitz and Emily Kazyak. Masculinities in cyberspace: An analysis of portrayals of manhood in men's rights activist websites. *Social Sciences*, 2016.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Hideo Shimazu. Expertclerk: a conversational case-based reasoning tool fordeveloping salesclerk agents in e-commerce webshops. *Artificial Intelligence Review*, 2002.

Apoorva Singh, Soumyodeep Dey, Anamitra Singha, and Sriparna Saha. Sentiment and emotion-aware multi-modal complaint identification. *Proceedings of the AAAI Conference on Artificial Intelligence (AISI Track)*, 2022.

Barry Smyth and Lorraine McGinty. An analysis of feedback strategies in conversational recommenders. In *Proceedings of the 14th Irish Artificial Intelligence and Cognitive Science Conference*. Citeseer, 2003.

Cailum MK Stienstra, Liam Hebert, Patrick Thomas, Alexander Haack, Jason Guo, and W Scott Hopkins. Graphormer-ir: Graph transformers predict experimental ir spectra using highly specialized attention. *Journal of chemical information and modeling*, 2024.

Cailum MK Stienstra, Teun van Wieringen, Liam Hebert, Patrick Thomas, Kas J Houthuijs, Giel Berden, Jos Oomens, Jonathan Martens, and W Scott Hopkins. A machine-learned "chemical intuition" to overcome spectroscopic data scarcity. *Journal of chemical information and modeling*, 2025.

Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 2000.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Kun Su, Krishna Sayana, Hubert Pham, James Pine, Yuri Vasilevski, Raghavendra Vasudeva, Marialena Kyriakidi, Liam Hebert, Ambarish Jash, Anushya Subbiah, et al. Regen: A dataset and benchmarks with natural language critiques and narratives. *arXiv preprint arXiv:2503.11924*, 2025.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 2016.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 2014.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 1999.

YunPeng Tan, Fangyu Liu, BoWei Li, Zheng Zhang, and Bo Zhang. An efficient multi-view multimodal data processing framework for social media popularity prediction. In *Proceedings of the 30th ACM International Conference on Multimedia*. Association for Computing Machinery, 2022.

Peyman Tehrani, Francesco Restuccia, and Marco Levorato. Federated deep reinforcement learning for the distributed control of nextg wireless networks. In *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2021.

Cynthia A Thompson, Mehmet H Goker, and Pat Langley. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 2004.

Lin Tian, Xiuzhen Zhang, and Jey Han Lau. DUCK: Rumour detection on social media by modelling user and comment propagation networks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022.

Frederich N Tou, Michael D Williams, Richard Fikes, D Austin Henderson Jr, and Thomas W Malone. Rabbit: An intelligent database assistant. In *AAAI*, 1982.

Thomas Tran and Robin Cohen. Hybrid recommender systems for electronic commerce. In *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press*, 2000.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Amos Tversky and Itamar Simonson. Context-dependent preferences. *Management science*, 1993.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

Janikke Solstad Vedeler, Terje Olsen, and John Eriksen. Hate speech harms: a social justice discussion of disabled norwegians' experiences. *Disability & Society*, 34(3):368–383, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv*, 2017.

Riza Velioglu and Jewgeni Rose. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge. *arXiv preprint arXiv:2012.12975*, 2020.

Bertie Vidgen, Dong Nguyen, Helen Margetts, Patricia Rossini, and Rebekah Tromble. Introducing CAD: the contextual abuse dataset. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021a.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021b.

Sebastian Wachs, Manuel Gámez-Guadix, and Michelle F Wright. Online hate speech victimization and depressive symptoms among adolescents: The protective role of resilience. *Cyberpsychology, Behavior, and Social Networking*, 2022.

Isaac Waller and Ashton Anderson. Quantifying social organization and political polarization in online platforms. *Nature*, 2021.

Xiaofei Wang, Ruibin Li, Chenyang Wang, Xiuhua Li, Tarik Taleb, and Victor CM Leung. Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. *IEEE Journal on Selected Areas in Communications*, 2020a.

Xiaofei Wang, Chenyang Wang, Xiuhua Li, Victor CM Leung, and Tarik Taleb. Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching. *IEEE Internet of Things Journal*, 2020b.

Yuxiang Wang, Xiao Yan, Chuang Hu, Quanqing Xu, Chuanhui Yang, Fangcheng Fu, Wentao Zhang, Hao Wang, Bo Du, and Jiawei Jiang. Generative and contrastive paradigms are complementary for graph self-supervised learning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024.

Christopher John Cornish Hellaby Watkins et al. Learning from delayed rewards. 1989.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 2018.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2022.

Xiong Xiong, Kan Zheng, Lei Lei, and Lu Hou. Resource allocation based on deep reinforcement learning in iot edge computing. *IEEE Journal on Selected Areas in Communications*, 2020.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. Sequential recommendation with latent relations based on large language model. In *Proceedings of the 47th International ACM SIGIR conference on research and development in information retrieval*, 2024.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. In

Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 2020.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*. PMLR, 2020.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420. Association for Computational Linguistics, 2019.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen De Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, et al. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic acids research*, 2024.

Wenjun Zeng, Dana Kurniawan, Ryan Mullins, Yuchi Liu, Tamoghna Saha, Dirichi Ike-Njoku, Jindong Gu, Yiwen Song, Cai Xu, Jingjing Zhou, et al. Shieldgemma 2: Robust and tractable image content moderation. *arXiv preprint arXiv:2504.01081*, 2025.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, 2021.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, 2019.

Zixuan Zhang, Yuning Jiang, Yuanming Shi, Ye Shi, and Wei Chen. Federated reinforcement learning for real-time electric vehicle charging and discharging control. In *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020.

Ron Zhu. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *arXiv preprint arXiv:2012.08290*, 2020.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*, 2024.

Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. Federated deep reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.