Enhancing Neural Network Efficiency: Novel Approaches to Knowledge Distillation

by

Aref Jafari

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

© Aref Jafari 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:	Il-Min Kim Professor, Dept. of Electrical and Computer Engineering, Queen's University
Supervisor:	Ali Ghodsi Professor, Dept. of Statistics and Actuarial Science, University of Waterloo
Internal Members:	Pascal Poupart Professor, Dept. of Computer Science, University of Waterloo
	Yaoliang Yu Associate Professor, Dept. of Computer Science, University of Waterloo
Internal-External Member:	Jun Liu Professor, Dept. of Applied Mathematics, 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

This thesis is based on five research works: four published papers and one accepted paper, authored by myself, my supervisor Prof. Ali Ghodsi, and several co-authors.

For the work 'Annealing Knowledge Distillation' (presented in Chapter 3) I developed the original idea and wrote the code, and ran the experiments together with Pranav Sharma. I wrote the paper with Mehdi Rezagholizadeh and Ali Ghodsi, who also provided feedback and editing. This work was published at EACL 2021 [67].

For the work 'Pro-KD: Progressive Distillation by Following the Footsteps of the Teacher' (presented in Chapter 4), Mehdi Rezagholizadeh and I are designated equal contributors in the published paper. The original idea was developed jointly by Mehdi Rezagholizadeh and myself. I wrote the code together with Puneeth S. M. Saladi and Pranav Sharma, and the experiments were run by Puneeth S. M. Saladi, Pranav Sharma, Ali Saheb Pasand and myself. I wrote the paper with Mehdi Rezagholizadeh and Ali Ghodsi, who also provided feedback and editing. This work was published at COLING 2022 [106].

For the work 'Continuation KD: Improved Knowledge Distillation through the Lens of Continuation Optimization' (presented in Chapter 5) I developed the original idea, wrote the code and ran the experiments. I wrote the paper together with Ivan Kobyzev. Mehdi Rezagholizadeh, Pascal Poupart and Ali Ghodsi provided feedback and editing. This work was published in Findings of ACL: EMNLP 2022 [65].

For the work 'Efficient Citer: Tuning Large Language Models for Enhanced Answer Quality and Verification' (presented in Chapter 6), Marzieh Tahaei and I are designated equal contributors in the published paper. I developed the original idea, and Marzieh Tahaei and I wrote the code and ran the experiments. The training data was collected by Marzieh Tahaei, David Alfonso-Hermelo, Khalil Bibi, Yimeng Wu and myself. The paper was written by Marzieh Tahaei, Ahmad Rashid and myself. Mehdi Rezagholizadeh, Ali Ghodsi, and Boxing Chen provided feedback and editing. This work was published in Findings of ACL: NAACL 2024 [118].

For the work 'Learning from Disagreement: Maximum Divergence Knowledge Distillation' (presented in Chapter 7) I developed the original idea and wrote the code. The experiments were run by Parsa Ashrafi Fashi, Hanieh Asadi Golmankhaneh, Shayan Salehi and myself. I wrote the paper with Mehdi Rezagholizadeh and Ali Ghodsi. Mehdi Rezagholizadeh, Vikram Appia, Emad Barsoum and Ali Ghodsi provided feedback and editing. This work has been accepted at NeurIPS 2026.

Abstract

Deep neural networks have achieved remarkable success across natural language processing and computer vision. However, the growing scale of these models has led to prohibitive computational and memory requirements, making their deployment on resource-limited devices increasingly challenging. Knowledge Distillation (KD) has emerged as a powerful model compression technique that enables a smaller student network to learn from a larger, more expressive teacher model. Despite its effectiveness, classical KD faces several fundamental challenges: (i) the capacity gap problem, where the student struggles to learn from an overly large teacher; (ii) the checkpoint search problem, where the best-trained teacher is not necessarily the optimal one for distillation; (iii) the noisy teacher problem, where unreliable teacher outputs can degrade the student's learning; and (iv) the generative distribution-mismatch problem (of which *exposure bias* is one facet), where students trained on a fixed set of teacher-generated sequences behave very differently when generating their own outputs at inference time. This thesis proposes five novel methods that address these challenges across three complementary paradigms: logit distillation, data distillation, and divergence-aware distillation.

In the logit distillation paradigm, we first propose Annealing-KD, which tackles the capacity gap by applying a dynamic temperature function exclusively to the teacher's logits, constructing an entropy-based curriculum that gradually exposes the student to increasingly sharp targets. Building on known properties of the temperature-scaled softmax, we give a self-contained analysis showing that the entropy of the teacher's temperature-scaled distribution increases monotonically with temperature (strictly, whenever the logits are not all equal), which provides a principled mechanism for controlling the sharpness of the supervisory signal. Second, we propose Progressive Knowledge Distillation (Pro-KD), which addresses both the capacity gap and the checkpoint search problem by having the student follow the teacher's entire training trajectory rather than distilling from a single, fully converged checkpoint. Combined with an adaptive temperature schedule, Pro-KD creates a doubly smooth curriculum that eliminates the need for costly checkpoint searches. Third, we propose Continuation-KD, which generalizes Annealing-KD by replacing its rigid two-stage structure with a single scheduled objective inspired by the continuation method from optimization theory. Continuation-KD's mechanism is an annealed hinge-loss margin: once the student's residual to the scaled teacher falls below the margin, the distillation term is exactly flat there and produces no gradient, so noise-level differences from the teacher are not transferred to the student. We give two exact identities for the annealed objective (one for the gradient, one for the loss Hessian) that isolate the teacher's contribution in a single term scaled linearly by the annealing factor and make precise what the hinge removes.

We are explicit that the step from this structure to improved optimization and generalization is not established by our experiments. All three methods consistently outperform the vanilla-KD and TAKD baselines on the GLUE benchmark, CIFAR-10, and CIFAR-100.

Moving beyond white-box distillation, we apply the data distillation paradigm to citation-grounded question answering. Using ChatGPT as a black-box teacher, we synthesize a training corpus of citation-enriched answers and fine-tune two compact student models: FIDCiter (3B parameters) and LLaMACiter (13B parameters). LLaMACiter exceeds the teacher on human-rated informativeness across all four evaluation sets and FIDCiter on two of the four, and FIDCiter matches or exceeds it on in-domain citation quality, while both use much shorter instruction templates; the teacher retains an edge on citation quality out of domain.

Finally, we propose Maximum Divergence Knowledge Distillation (MDKD) for autoregressive language model distillation. MDKD uses a rejection-sampling mechanism to preferentially select tokens where the student underestimates the teacher, constructing training sequences concentrated in regions where the student underweights teacher-supported tokens. We prove that the stochastic acceptance branch samples exactly from the positive teacher–student residual, with acceptance probability equal to their total variation distance. This active, divergence-aware approach targets the distributional-coverage aspect of the training–inference mismatch in sequence-level distillation; because its training prefixes are teacher-generated, direct on-policy mitigation of exposure bias is left to future work.

Collectively, the five proposed methods trace a progression from passive distillation with static objectives to active, adaptive paradigms that tailor the supervisory signal to the student's current learning state. By addressing the capacity gap, checkpoint search, noisy teacher, and generative distribution-mismatch problems through complementary mechanisms, this thesis advances the design of efficient, high-performing models that bridge the gap between large-scale deep learning and real-world deployment constraints.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Ali Ghodsi, for his guidance, support and encouragement throughout my PhD. His insight and openness to new ideas, and the trust he placed in my research, were invaluable to this work. Beyond the technical guidance, he played an important role in shaping my vision of research. I am sincerely grateful for the opportunity to learn from him.

I would also like to thank my co-authors, and in particular Mehdi Rezagholizadeh, for our collaborations, his ideas and feedback, and the many insightful discussions that helped shape the work in this thesis.

I am grateful to the School of Computer Science at the University of Waterloo for providing the environment and resources that made it possible for me to pursue my PhD.

Dedication

This thesis is dedicated to my family.

To my mother, whose resilience and sacrifices made everything possible.

To my father, whose guidance and encouragement have stayed with me.

To my brother, Ali, for his constant companionship.

To my uncle, Reza, my role model, who gave me strength in the hardest of times.

And to my wonderful wife, Ayda, whose love and support have sustained me every step of the way.

Table of Contents

Examining Committee ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vii

Dedication viii

List of Figures xvi

List of Tables xviii

1 Introduction 1

1.1 Motivation 1

1.2 Problem Definition 3

1.2.1 Classical Knowledge Distillation 3

1.2.2 Problem 1: The Capacity Gap 4

1.2.3 Problem 2: The Checkpoint Search Problem 5

1.2.4 Problem 3: The Noisy Teacher Problem 5

1.2.5 Problem 4: Generative Distribution Mismatch . . . 6
1.3 The Landscape of Knowledge Distillation . . . 7
1.3.1 Logit Distillation . . . 7
1.3.2 Intermediate-Layer Distillation . . . 9
1.3.3 Data Distillation . . . 11
1.3.4 Divergence-Aware Distillation . . . 12
1.4 Thesis Contributions . . . 14
1.4.1 Advancing Logit Distillation via Adaptive Optimization . . . 14
1.4.2 Data Distillation for Citation-Grounded Question Answering . . . 15
1.4.3 Active Divergence Distillation for Generative Models . . . 15
1.5 Thesis Organization . . . 16

2 Background and Related Work **17**
2.1 Foundations of Knowledge Distillation . . . 17
2.1.1 Dark Knowledge and Temperature Scaling . . . 17
2.1.2 The KD Objective Function . . . 19
2.1.3 Limitations of the Classical Formulation . . . 19
2.2 The Capacity Gap . . . 19
2.2.1 Theoretical Foundations: Generalized Distillation and VC Dimension 20
2.2.2 Teacher–Assistant Knowledge Distillation (TAKD) . . . 21
2.2.3 Beyond the Parametric Capacity Gap . . . 22
2.3 Theoretical Preliminaries . . . 22
2.3.1 Information Theory and Target Complexity . . . 23
2.3.2 Optimization Topology and Loss Landscapes . . . 23
2.3.3 Spectral Bias and the Frequency Principle . . . 24
2.3.4 Continuation Methods and Graduated Non-Convexity . . . 25
2.4 Distillation in the Generative Era . . . 25
2.4.1 White-Box vs. Black-Box Distillation . . . 26

2.4.2 Data Distillation and Synthetic Instruction Tuning 26
2.4.3 Exposure Bias in Autoregressive Generation 27
2.5 Active and Divergence-Based Distillation 27
2.5.1 The Limitation of Static Datasets 28
2.5.2 Continuous-Domain Precursors: Backward Knowledge Distillation . 28
2.5.3 Transition to the Discrete Generative Domain 29
2.6 Summary 29

3 Annealing Knowledge Distillation **31**
3.1 Overview 31
3.2 Related Work 32
3.2.1 The Capacity Gap and Existing Solutions 32
3.2.2 Annealing in Knowledge Distillation 33
3.3 Method: Annealing Knowledge Distillation 34
3.3.1 Example 36
3.3.2 Rationale Behind Annealing KD: An Information-Theoretic Curriculum 37
3.4 Experiments 45
3.4.1 Datasets 45
3.4.2 Experimental Setup for Image Classification Tasks 46
3.4.3 Experimental setup for GLUE tasks 46
3.4.4 GLUE Results 47
3.5 Scaling to the Generative Era: Upcycling Hybrid LLMs 49
3.5.1 Task Definition: Hybrid Model Upcycling 50
3.5.2 Experimental Setup 50
3.5.3 Results and Discussion 51
3.6 Discussion 52
3.7 Conclusion and Future Work 53

4 Progressive Knowledge Distillation **55**

4.1 Overview . . . 55

4.2 Introduction . . . 56

4.3 Related Work . . . 57

4.3.1 Capacity-Gap Problem in KD . . . 58

4.3.2 Checkpoint Search Problem in KD . . . 58

4.4 Methodology: Pro-KD . . . 59

4.4.1 Why Does Pro-KD work? . . . 62

4.5 Experiments and Results . . . 63

4.5.1 Experimental Setup for Image Classification Tasks . . . 63

4.5.2 Experimental Setup for Natural Language Understanding Tasks on the GLUE Benchmark . . . 65

4.5.3 Experimental Setup for Question Answering . . . 67

4.5.4 Ablation Studies and Further Analysis . . . 68

4.6 Conclusion . . . 68

5 Continuation Knowledge Distillation **70**

5.1 Overview . . . 70

5.2 Introduction . . . 71

5.3 Related Works . . . 72

5.3.1 Knowledge Distillation and the Capacity Gap . . . 72

5.3.2 Continuation Optimization . . . 72

5.4 Background: From Annealing-KD to Continuation-KD . . . 73

5.5 Methodology . . . 73

5.5.1 The Structure of the Annealed Objective and the Hinge Mechanism 73

5.5.2 Loss Formulation and Training Procedure . . . 78

5.6 Experiments . . . 82

5.6.1 Hardware Details . . . 82

5.6.2 Image Classification . . . 83
5.6.3 Natural Language Understanding . . . 85
5.7 Analysis . . . 87
5.8 Conclusion . . . 89

6 Data Distillation: Efficient Citer **92**
6.1 Overview . . . 92
6.2 Introduction . . . 93
6.3 Related work . . . 94
6.4 Training . . . 95
6.4.1 Data Distillation Pipeline . . . 95
6.4.2 Student Models . . . 96
6.5 Evaluation Criteria . . . 97
6.5.1 Human evaluation . . . 97
6.6 Results . . . 98
6.7 Conclusion . . . 100

7 Maximum Divergence Knowledge Distillation **101**
7.1 Overview . . . 101
7.2 Introduction . . . 102
7.3 Related Work . . . 104
7.3.1 Divergence-aware distillation in continuous domains . . . 104
7.3.2 Generalized knowledge distillation . . . 104
7.3.3 Speculative knowledge distillation . . . 105
7.3.4 Divergence-aware sequence distillation . . . 105
7.3.5 Active, hard-example, and adversarial distillation . . . 105
7.3.6 Discussion . . . 106
7.4 Methodology . . . 106

7.4.1 Preliminaries . . . 106
7.4.2 The divergence-maximization principle . . . 107
7.4.3 Divergence-aware token sampling . . . 107
7.4.4 Theoretical analysis . . . 108
7.4.5 Training objective . . . 111
7.4.6 Practical considerations . . . 112
7.4.7 Training algorithm . . . 113
7.5 Experiments . . . 113
7.5.1 Setup . . . 114
7.5.2 Tasks . . . 115
7.5.3 Main Results . . . 116
7.5.4 Divergence Gain Analysis . . . 116
7.5.5 Interpreting the Acceptance Rate . . . 117
7.5.6 In-Domain Transfer and Regularization . . . 119
7.5.7 Scaling to Larger Teachers . . . 120
7.5.8 Hyperparameter Analysis . . . 120
7.5.9 Discussion . . . 122

8 Conclusion and Future Work **126**
8.1 Summary of Contributions . . . 126
8.1.1 Logit Distillation: Addressing the Capacity Gap, Checkpoint Search, and Noisy Teacher Problems . . . 126
8.1.2 Data Distillation: Citation-Grounded Question Answering . . . 128
8.1.3 Divergence-Aware Distillation: Active Knowledge Transfer for Generative Models . . . 128
8.2 Unifying Themes . . . 129
8.3 Limitations . . . 129
8.4 Future Research Directions . . . 131

References **133**

APPENDICES **150**

A Annealing Knowledge Distillation **151**

A.1 Experimental parameters of the image classification tasks 151

A.2 BERT Experiments 151

A.3 Multi-Seed Study of Annealing-KD 155

B Progressive Knowledge Distillation **157**

B.1 Why Does Pro-KD Work? 157

B.2 Hyper-parameters 159

C Continuation Knowledge Distillation **161**

C.1 Derivation of the Annealed MSE Hessian 161

D Data distillation: Efficient Citer **164**

D.1 Training details 164

D.2 Inference parameters 164

D.3 Data collection prompt 165

D.4 Automatic evaluation metric from ALCE Benchmark 165

D.5 Example of our human evaluation 166

E Maximum Divergence Knowledge Distillation **168**

E.1 Proofs of the Distributional Characterization 168

E.2 Acceptance-Rate Diagnostics 171

E.2.1 Acceptance tracks divergence 171

E.2.2 A low acceptance rate can still re-route the trajectory 171

E.3 Is the Fallback Branch Divergence-Targeting? 172

List of Figures

3.1 Illustrating the Stage I of the Annealing-KD technique. Given a pre-trained teacher network, we can derive the annealed output of the teacher at different temperature using the annealing function $\Phi(\mathcal{T})$. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$. 36

3.2 (a) Data samples and trained teacher. (b) Annealed teacher in different temperatures. (c) Student after matching to annealed teacher in $\mathcal{T} = 10$. (d) Student after matching to annealed teacher in $\mathcal{T} = 5$. (e) Student after matching to $\mathcal{T} = 1$. (f) Student trained without KD. 38

4.1 Stage I and Stage II of the Pro-KD method. Stage I: the teacher is trained on the labeled training data. The student at each step tries to mimic the behavior of the corresponding teacher checkpoint. The logits of the teacher at each time step is attenuated with the temperature parameter. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$. Stage II: training the student with the labeled data only using the cross entropy loss. n_i for $1 \leq i \leq E_T$ (with E_T the number of teacher epochs) refers to the number of training epochs of the student corresponding to the i^{th} training epoch of the teacher. n_is should add up to N, the preset total number of Stage-I student epochs. Bear in mind that since the teacher is usually early stopped, teacher is trained less longer than the student model. 59

5.1 Principle diagram illustrating different components of Continuation-KD. The figure shows the three scheduled components: the teacher logits are scaled by $\Phi(\mathcal{T}_i)$, the hinge margin is scaled by the same factor, and $\psi(i)$ interpolates between the continuation loss and the supervised task loss. The diagram specifies the training objective; it does not establish that the continuation loss is easier to optimize, or has lower curvature, than the supervised objective. For legibility the diagram writes the annealing factor and its index as $\phi(\tau_i)$; these are the $\Phi(\mathcal{T}_i)$ and $\mathcal{T}_i$ of the text. The term labeled $\mathcal{L}_{CE}$ denotes the supervised task loss, which is cross-entropy for the classification tasks and the corresponding regression loss for STS-B. 90

5.2 (a) Illustrates the behavior of the student model after training with a noisy teacher without using continuation-KD. (b) Illustrates the behavior of the student model after training with a noisy teacher by using continuation-KD for training. Blue curves are the samples from the noisy teacher. Orange curves are the samples of the trained student in each scenario. The figure is a qualitative illustration from a single run on one synthetic function; it reports no held-out error and does not by itself establish robustness across noise levels or random initializations, nor validate the Gaussian margin model of Section 5.5.2. 91

6.1 Our data distillation pipeline: starting from 23K candidate MS MARCO examples, ChatGPT (teacher) generates citation-enriched answers; after filtering and citation-format cleaning, approximately 18K examples are used to fine-tune the FIDCiter and LLaMACiter student models. 93

List of Tables

3.1 Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models 47

3.2 Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models 48

3.3 DistilRoBERTa results for Annealing KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy; MNLI mismatched is reported but not aggregated. Aggregates are computed from the displayed cells and rounded to two decimals. 48

3.4 Performance of DistilRoBERTa trained by annealing KD on the GLUE leaderboard (test set) compared with Vanilla KD and TAKD. We applied the standard tricks to all 3 methods and fine-tune RTE, MRPC and STS-B from the trained MNLI student model. Matthews correlation for CoLA, Pearson/Spearman for STS-B, F1/accuracy for MRPC and QQP, matched/mismatched accuracy for MNLI, accuracy elsewhere. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed tasks; paired metrics (MRPC, STS-B, QQP) are averaged within a task and MNLI matched/mismatched are averaged before aggregation. Aggregates are computed from the displayed cells and rounded to one decimal. 49

3.5 BERT-Small results for Annealing KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy; MNLI mismatched is reported but not aggregated. Aggregates are computed from the displayed cells and rounded to two decimals. 49

3.6 0-Shot normalized accuracy of hybrid models on the LM Evaluation Harness [46] benchmark. HypeNet–KL uses standard KL-Divergence distillation in HALO Stage 2, while HypeNet Annealing replaces it with Annealing-KD. The first row shows the original Qwen3-1.7B Transformer as a reference. . 52

4.1 The best performing checkpoints vary for different size of the models, and different tasks. It is evident that the best checkpoint of the teacher does not necessarily lead to the best performing student model. 57

4.2 Comparing the test accuracy of Pro-KD, TAKD [91], Annealing-KD, RCO [70], regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models 62

4.3 Comparing the test accuracy of Pro-KD, TAKD [91], Annealing-KD, RCO [70], regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models 63

4.4 Dev set results of training DistilRoBERTa using the $\text{RoBERTa}_{\text{Large}}$ model on the GLUE benchmark. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy for the remaining tasks; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of CoLA, RTE, MRPC, STS-B, SST-2, QNLI, QQP, and the mean of the two MNLI accuracies, computed consistently for every row from the displayed cells. . 64

4.5 Test set results of DistilRoBERTa trained on the GLUE tasks using the $\text{RoBERTa}_{\text{Large}}$ teacher model. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 64

4.6 BERT-Small results for Pro-KD on the GLUE dev set. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 65

4.7 The GLUE leaderboard test results of training BERT-Small from the $\text{BERT}_{\text{Large}}$ teacher. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 65

4.8 DistilRoBERTa results for Pro-KD on SQuAD. 66

4.9 BERT-Small results for Pro-KD on SQuAD. 66

4.10 Experimenting the impact of the adaptive temperature factor in Pro-KD using DistilRoBERTa trained with RoBERTa-large on the GLUE benchmark 67

5.1 Comparing the test accuracy of Continuation-KD, Annealing-KD, TAKD, Vanilla-KD, and finetuning on CIFAR-10 dataset with ResNet model . . . 84

5.2 Comparing the test accuracy of Continuation-KD, Annealing-KD, TAKD, Vanilla-KD, and finetuning on CIFAR-100 dataset with ResNet model . . . 84

5.3 DistilRoBERTa results for Continuation-KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 86

5.4 Performance of DistilRoBERTa trained by Continuation-KD on the GLUE leaderboard (test set) compared with Vanilla-KD, TAKD, and Annealing-KD. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, matched accuracy for MNLI, accuracy for the remaining tasks; Score is the unweighted mean of the eight task metrics. The last three task columns are corrected from Table 4 of the published EMNLP 2022 Findings paper, where they appear transposed relative to the dev-set table and to the Chapter 3 leaderboard results. 86

5.5 BERT-Small results for Continuation-KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 87

5.6 BERT-Small results for Continuation-KD on test set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. 87

5.7 Performance of distilRoBERTa on MRPC and RTE datasets for different dynamically changing components of continuation-KD loss 88

6.1 Human evaluation of model outputs on MS MARCO, MIRACL, ELI5, and ASQA datasets. For MS MARCO and MIRACL, 50 randomly chosen samples are assessed, while 100 randomly selected samples are evaluated for ELI5 and ASQA datasets. Each sample comprises five passages. Criteria evaluated include Fluency (FL), Informativeness (INF), and Citation Quality (CQ). 95

6.2 Comparative Evaluation on ELI5 and ASQA using ALCE Benchmark with 1000 samples for 3 and 5 passages respectively. Numbers for ChatGPT and Vicuna models are taken from [47]. 97

7.1 Main results: KD methods compared across two teacher–student pairs and three task families. We report GSM8K accuracy, DialogSum ROUGE-L on 500 held-out dialogues, and HumanEval Pass@1 (execution-based) when training on MBPP. All methods use the same training set, a matched optimization budget (3 epochs, same batch size), and the same evaluation decoding, but method-specific sampling and divergence hyperparameters (Table 7.6). Bold marks the best method per column; MDKD is best on every (model, task) cell. 116

7.2 Divergence gain analysis on GSM8K (512 samples). MDKD sequences maintain near-teacher accuracy while increasing the student's cross-entropy loss by 43.6%, confirming that they target the student's blind spots. 117

7.3 Fallback-policy ablation. Only the fallback rule varies; the teacher (Qwen2.5-14B-Instruct), student (base Qwen2.5-1.5B), data (1K GSM8K training samples), loss (reverse KL), and the full 1,319-problem test set are held fixed, and every configuration is trained for three epochs. The absolute accuracies are therefore not comparable with Table 7.5, whose 1K column trains for five epochs and reports 64.90 for the same pair; this ablation is a within-table comparison of fallback policies, not a restatement of the headline result. Every policy that retains the stochastic acceptance step performs comparably; disabling it does not. 118

7.4 In-domain transfer. *Left:* best Math-500 accuracy after s1K training (Qwen2.5-1.5B-Instruct student, Qwen2.5-14B-Instruct teacher). *Right:* UI-math distillation with 2K samples from the off-the-shelf 14B teacher; GSM8K measures within-domain transfer. 119

7.5 Scaling experiments: 14B teacher $\rightarrow$ 1.5B student on GSM8K. MDKD is the strongest method at both data sizes. Results shown at the best epoch for each method. 120

7.6 MDKD and baseline hyperparameters. Baseline parameters follow the original SKD repository; MDKD uses a single shared configuration across tasks. 121

7.7 Loss-controlled baselines on GSM8K (Qwen2.5-14B-Instruct teacher, base Qwen2.5-1.5B student, 7K training samples). The forward-KL column reproduces the 7K results of Table 7.5; the reverse-KL column re-runs each baseline under MDKD's objective. MDKD is defined with reverse KL, so no forward-KL entry applies. 122

A.1 Hyper-parameters of CIFAR-10 and CIFAR-100 experiments 152

A.2 Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE tasks 152

A.3 Model specific Hyper-parameters for DistilRoBERTa on GLUE tasks . . . 153

A.4 Ablation on DistilRoberta Annealing KD with temperature tuning 154

A.5 Ablation on DistilRoberta Annealing KD with different loss functions . . . 154

A.6 Model specific Hyper-parameters for BERT-Small on GLUE tasks 154

A.7 Multi-seed DistilRoBERTa results for Annealing KD on the GLUE dev set (10 seeds). For the student models (From scratch, Vanilla KD, TAKD, Annealing KD), each cell reports the mean ± standard deviation over 10 random seeds; the Teacher and TA rows are single runs. MNLI is reported as matched/mismatched accuracy. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy (MNLI mismatched is reported but not aggregated). Best student result per column in bold. 155

A.8 Multi-seed BERT-Small results for Annealing KD on the GLUE dev set (10 seeds). For the student models (From scratch, Vanilla KD, TAKD, Annealing KD), each cell reports the mean ± standard deviation over 10 random seeds; the Teacher and TA rows are single runs. MNLI is reported as matched/mismatched accuracy. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy (MNLI mismatched is reported but not aggregated). Best student result per column in bold. 156

B.1 Model specific Hyper-parameters for BERT-Small on GLUE 159

B.2 Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE 159

B.3 Model specific Hyper-parameters for DistilRoBERTa on GLUE 160

B.4 Model specific Hyper-parameters for DistilRoBERTa on SQuAD 160

B.5 Model specific Hyper-parameters for BERT-Small on SQuAD 160

D.1 ChatGPT output of the sample example. The answer to the given query is broken into 3 pieces 167

D.2 FLAN-T5 output of the sample example. The answer to the given query is broken into 1 section 167

D.3 LLaMA output of the sample example. The answer to the given query is broken into 1 section 167

Chapter 1

Introduction

1.1 Motivation

The evolution of deep learning over the past decade has been fundamentally driven by the principle of scale. From the proliferation of deep Convolutional Neural Networks (CNNs) in computer vision [76, 57] to the dominance of massive Transformer-based architectures in Natural Language Processing (NLP) [129], empirical scaling laws have consistently demonstrated that increasing model capacity, dataset size, and compute leads to predictable improvements in generalization and downstream task performance [71]. The introduction of BERT [36] marked a paradigm shift in NLP, establishing pre-trained language models as the foundation for transfer learning across diverse tasks. This trajectory has since accelerated dramatically: GPT-3 scaled to 175 billion parameters and demonstrated surprising few-shot capabilities [13]; PaLM extended the frontier to 540 billion parameters with improved reasoning abilities [26]; and open-weight families such as LLaMA [124] and Mistral [68] have democratized access to capable language models ranging from 7 to 65 billion parameters. The field has thus entered the era of *foundation models*, in which a single pre-trained model, often comprising hundreds of billions of parameters, serves as the starting point for a vast range of downstream applications.

While this massive scale unlocks unprecedented capabilities, including advanced reasoning, zero-shot generalization, and complex instruction following, it simultaneously introduces a severe *deployment bottleneck*. The memory footprints, inference latencies, and energy consumption of these models render them impractical for resource-constrained environments such as mobile devices, embedded systems, and real-time applications [115]. A single inference pass of a 70-billion-parameter model can require multiple high-end GPUs,

placing it far beyond the reach of most deployment scenarios. Even in datacenter settings, serving large models at scale incurs substantial financial and environmental costs. This tension between the need for massive scale and the demand for computational efficiency defines one of the central challenges of modern deep learning.

Several families of techniques have been proposed to address this challenge. *Pruning* methods remove redundant weights or entire neurons from trained networks, yielding sparser models with reduced computational cost [56, 77]. *Quantization* reduces the numerical precision of model weights and activations, compressing memory requirements and enabling efficient integer-arithmetic inference on specialized hardware [63, 34]. *Neural Architecture Search* (NAS) automates the discovery of efficient model topologies tailored to specific hardware constraints [156, 119]. *Low-rank factorization* and parameter-efficient fine-tuning methods such as LoRA [60] reduce the effective number of trainable parameters by constraining weight updates to low-rank subspaces. While each of these approaches is effective in specific settings, they share common limitations: pruning and quantization often require hardware-specific tuning and may degrade performance on complex tasks; NAS is computationally expensive and produces architectures tied to particular search spaces; and low-rank methods primarily reduce fine-tuning cost rather than inference cost. Critically, most of these techniques operate by modifying the model architecture or its numerical representation, constraining the design space and potentially sacrificing generality.

In contrast, *Knowledge Distillation* (KD) [59, 14] offers a fundamentally different and complementary approach. Rather than modifying the model itself, KD operates at the *training level*: it trains a compact *student* model to reproduce the behavior of a larger, more capable *teacher* model. The key insight, first articulated by Hinton et al. [59], is that the teacher's softened output probabilities, referred to as *soft targets*, encode rich inter-class relational information that one-hot ground-truth labels cannot convey. For instance, a teacher trained on image classification may assign non-trivial probability mass to visually similar classes (e.g., "cat" and "lynx"), revealing structural knowledge about the data manifold. This so-called *dark knowledge* has been shown to improve student generalization and convergence speed [59, 120], particularly when the student has limited capacity or is trained with few samples.

KD enjoys several properties that distinguish it from other compression paradigms. First, response-based KD can transfer across different architectures when their output spaces are aligned, so the student and teacher need not share structure, depth, or layer types. Second, classical offline KD is post hoc, applied after the teacher has been fully trained and without modifying it; online and progressive variants such as Pro-KD (Chapter 4) are not. Logit-level KD requires the teacher's logits or probability outputs, and intermediate-feature methods additionally require activations, but teacher weights and

gradients are not needed for ordinary response-based KD. Third, KD is *compatible* with other compression methods; it can be combined with pruning, quantization, or efficient architectures to achieve compounding efficiency gains. These properties make KD uniquely suited to the era of foundation models, where the goal is often to transfer the capabilities of a massive pre-trained model into a deployable student of fundamentally different design.

Despite its promise, classical KD relies on a passive, static matching paradigm that increasingly falls short as models grow in scale and complexity. The student is trained to minimize the divergence between its outputs and the teacher's outputs over a fixed dataset, assuming that all training examples are equally informative, that the student has sufficient capacity to approximate the teacher, and that the teacher's output distribution is well calibrated and noise-free. In practice, these assumptions break down, especially when distilling across large capacity gaps, across architectural families, or in generative settings where outputs are discrete token sequences. This thesis is motivated by the broad goal of *re-imagining the distillation process* to address the fundamental limitations of classical KD, proposing novel, adaptive, and dynamic approaches to knowledge transfer that are grounded in optimization theory, information theory, and the specific challenges of modern deep learning.

1.2 Problem Definition

We begin by formalizing the classical KD framework and then identify the four fundamental problems that this thesis addresses.

1.2.1 Classical Knowledge Distillation

Let T denote a pre-trained teacher model and S_θ a student model parameterized by θ. Given an input x with ground-truth label y, let $z_t(x)$ and $z_s(x;\theta)$ denote the pre-softmax logits of the teacher and student, respectively. The temperature-scaled softmax output of a model with logits z is defined as

$$p_i(\tau) = \frac{\exp(z_i/\tau)}{\sum_{j=1}^{K} \exp(z_j/\tau)}, \tag{1.1}$$

where K is the number of classes and $\tau > 0$ is a temperature parameter that controls the entropy of the distribution: higher temperatures produce smoother distributions that

reveal more inter-class structure. The standard KD training objective augments the student's cross-entropy loss with a divergence term that encourages the student to match the teacher's softened output distribution [59]:

$$\mathcal{L}_{\mathrm{KD}}(\theta) = \alpha\, \mathcal{L}_{\mathrm{CE}}\big(y,\, p_s(x;\theta,\tau=1)\big) + (1-\alpha)\, \tau^2\, \mathrm{KL}\big(p_t(x;\tau)\,\|\,p_s(x;\theta,\tau)\big), \tag{1.2}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss between the student's predictions and the hard labels, $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence, p_t and p_s are the temperature-scaled teacher and student distributions, and $\alpha \in [0,1]$ balances the two objectives. The factor τ^2 compensates for the gradient magnitude reduction caused by temperature scaling [59].

While this formulation has proven effective across image classification [149], machine translation [74], and natural language understanding [110], it rests on assumptions that become increasingly fragile as model complexity and task diversity grow. This thesis identifies and addresses four fundamental problems that bottleneck knowledge transfer in modern settings.

1.2.2 Problem 1: The Capacity Gap

Classical KD assumes that the student possesses sufficient representational capacity to approximate the teacher's function smoothly. However, when the capacity difference between teacher and student is large, the teacher's output distribution becomes excessively sharp and complex, effectively encoding high-frequency decision boundaries that a low-capacity student cannot represent.

This phenomenon has been studied both theoretically and empirically. Lopez-Paz et al. [85] analyzed KD within the generalized-distillation rate framework and showed that the benefit of distillation depends on a delicate balance between the teacher's and student's function-class complexities: when the teacher's capacity is too large relative to the student's, the resulting distillation upper bound can exceed the corresponding bound for training the student without a teacher. Mirzadeh et al. [91] provided complementary empirical evidence, demonstrating that if one gradually increases the capacity of the teacher, the student's performance first improves but then degrades beyond a critical point. Their proposed solution, Teacher–Assistant Knowledge Distillation (TAKD) [91], introduces intermediate-capacity "assistant" models that bridge the gap incrementally; however, TAKD requires training additional networks, incurring both computational overhead and error accumulation across the distillation chain.

The capacity gap problem is further exacerbated by the observation that neural networks exhibit a *spectral bias* [102]: they learn low-frequency (smooth) components of the

target function before high-frequency (sharp) components. The capacity-gap effect is empirically established in several settings, but neither VC-style capacity analysis nor spectral bias proves that a larger teacher necessarily produces sharper logits or a more rugged parameter-space loss for the student [80]. In this thesis, scheduled target scaling is a motivated intervention whose optimization effect is assessed empirically rather than a consequence of those theories. This motivates the development of KD methods that dynamically control the complexity of the supervisory signal, smoothing the knowledge transfer process to match the student's learning capacity.

1.2.3 Problem 2: The Checkpoint Search Problem

Traditional KD treats the teacher as a static, fully converged endpoint and transfers knowledge exclusively from this final model. However, Cho and Hariharan [25] demonstrated that fully trained teachers are not necessarily the best teachers for distillation; in many settings, an early-stopped teacher checkpoint yields better student performance than the converged model. This observation implies that practitioners must search through the teacher's training trajectory to identify the optimal checkpoint for distillation, a process we refer to as the *checkpoint search problem*.

The checkpoint search problem is particularly expensive for modern pre-trained language models, where fine-tuning a single task may produce dozens of checkpoints, each requiring separate distillation experiments to evaluate. Moreover, the optimal checkpoint varies across tasks and student configurations [106], making it impossible to identify a single universally optimal teacher. Existing solutions such as Early-Stopped KD (ESKD) [25] and Route-Constrained Optimization (RCO) [70] attempt to leverage the teacher's training trajectory but still require manual or heuristic checkpoint selection.

This problem reveals a deeper limitation of classical KD: by discarding the teacher's intermediate training states, it wastes the rich curriculum-like information encoded in the teacher's optimization trajectory. A student that could follow the teacher's learning path, from simple to complex representations, would benefit from a naturally smooth and progressive training signal.

1.2.4 Problem 3: The Noisy Teacher Problem

Pre-trained teachers are not infallible. They often produce overconfident, noisy, or erroneous predictions on ambiguous samples, distribution boundary cases, or out-of-distribution

inputs [54]. Classical KD forces the student to match the teacher's output distribution uniformly across all samples, including those where the teacher's predictions are unreliable.

For a capacity-constrained student, this uniform matching is particularly damaging: the student may waste its limited representational power memorizing the teacher's high-frequency noise rather than learning the underlying data manifold. This problem is related to the well-studied phenomenon of label noise in supervised learning [3, 151], where neural networks have been shown to first learn clean, generalizable patterns before overfitting to noisy labels. In the KD context, the teacher's soft labels serve as supervision, and noise in these labels propagates directly to the student.

Existing approaches to noise-robust KD are limited. The distillation signal has been analyzed as an implicit re-weighting of examples by the teacher's confidence [120], and label smoothing has been studied as a defense against label noise [86], but these accounts are descriptive or heuristic rather than giving formal guarantees. A principled solution requires a training objective that structurally prevents the student from overfitting to bounded noise in the teacher's predictions.

1.2.5 Problem 4: Generative Distribution Mismatch

The three problems above arise primarily in the context of discriminative models (classifiers, regressors). However, modern deep learning is increasingly dominated by *generative* models, particularly autoregressive language models that produce discrete token sequences one step at a time. This setting introduces a fundamentally new challenge: *distribution mismatch between training and inference.*

In classical KD for generative models, the student is trained to match the teacher's token-level probabilities on a fixed corpus of teacher-generated or ground-truth sequences (teacher-forcing). However, during inference, the student generates tokens autoregressively from its own distribution. As soon as the student produces a token that deviates from the training distribution, it enters a region of the sequence space on which it was never supervised. Small errors compound across the sequence, leading to progressively worse outputs [7, 105]. This phenomenon, known as *exposure bias*, is a fundamental limitation of teacher-forced training.

Recent work has begun to address this mismatch. Generalized Knowledge Distillation (GKD) [1] trains the student on its own generated sequences while querying the teacher for token-level supervision, adopting an on-policy approach. Speculative Knowledge Distillation (SKD) [140] interleaves student and teacher sampling, replacing poorly ranked student

tokens with teacher samples. Distribution-Aligned Sequence Distillation (DASD) [142] filters teacher-generated candidate responses based on sequence-level teacher–student divergence. However, none of these methods explicitly maximizes teacher–student disagreement at the token level during data construction, leaving a gap in the literature for *active*, divergence-driven distillation.

1.3 The Landscape of Knowledge Distillation

To contextualize the contributions of this thesis, we now survey the knowledge distillation landscape. The field has grown substantially since the foundational work of Hinton et al. [59], with comprehensive surveys provided by Gou et al. [50] and Mansourian et al. [87]. The standard taxonomy, established by these surveys, categorizes KD methods by the *source* of knowledge transferred from teacher to student: *logit-based* methods match output distributions; *feature-based* methods align intermediate representations; and *similarity-based* methods preserve relational structure among examples.

In this thesis, we adopt a complementary, *functional* taxonomy that organizes the field by the level of access to the teacher model and the nature of the optimization strategy. This perspective naturally delineates four paradigms: *logit distillation* and *intermediate-layer distillation* (both white-box, requiring access to teacher internals), *data distillation* (black-box, requiring only teacher-generated outputs), and *divergence-aware distillation* (active, seeking regions of maximum teacher–student disagreement). This functional view maps directly onto the contributions of this thesis and provides a clear organizational framework for the chapters that follow. We now describe each paradigm in detail.

1.3.1 Logit Distillation

Logit distillation, also called *response-based* distillation, is the classical and most widely utilized form of KD [59, 50]. It operates in a white-box setting, assuming full access to the teacher's pre-softmax logits. By applying temperature scaling (Equation 1.1) to the logits, sharp probability distributions are smoothed, revealing the inter-class similarity structure the teacher has learned. The student is trained to minimize the divergence between its own softened outputs and those of the teacher (Equation 1.2).

The idea of training a small model to mimic a larger one dates back to Bucilu˘a et al. [14], who demonstrated that a compact neural network can approximate the function learned by a large ensemble. Ba and Caruana [5] extended this by showing that shallow

networks trained on the soft outputs of deep networks can match their accuracy, challenging the prevailing assumption that depth was strictly necessary for high performance. Hinton et al. [59] formalized the modern KD framework by introducing temperature-scaled softmax and the combined hard/soft target loss of Equation 1.2, establishing the recipe used by most subsequent work.

Since then, a rich body of literature has refined and extended logit distillation along several axes.

Addressing the capacity gap. The capacity gap problem (Section 1.2.2) has motivated several lines of work. TAKD [91] introduces intermediate teacher-assistant models to smooth the knowledge transfer chain, but at the cost of additional training and error propagation. Annealing-KD [67], proposed in this thesis (Chapter 3), takes a different approach: it introduces a dynamic temperature function that progressively transitions the student from highly smoothed (low-frequency) targets to the full-complexity (high-frequency) teacher outputs. This creates an information-theoretic curriculum that provably and monotonically decreases target entropy, avoiding the need for auxiliary networks. Pro-KD [106], also proposed in this thesis (Chapter 4), extends this further by training the student concurrently with the teacher, following the teacher's own training trajectory to naturally smooth the knowledge transfer process.

Robustness to noisy supervision. The noisy teacher problem (Section 1.2.4) has received comparatively less attention. Some work explains the teacher's soft targets as an implicit re-weighting of training examples by teacher confidence [120], while others study label smoothing as a defense against noisy labels [86]. Continuation-KD, proposed in this thesis (Chapter 5), takes a structurally different approach inspired by continuation optimization [2]: it formulates the KD objective as an annealed hinge loss whose margin creates a flat region (zero gradient and zero Hessian) below which residual errors are ignored, limiting the transfer of small fluctuations in the teacher's outputs to the student.

Logit normalization and decoupling. Recent work has examined the role of logit preprocessing in KD. NormKD [23] customizes the temperature for each sample based on the characteristics of its logit distribution, arguing that a single global temperature is suboptimal. Sun et al. [116] propose logit standardization via z-score normalization as a plug-and-play preprocessing step. Zhao et al. [155] decouple the KD loss into target-class and non-target-class components, showing that the non-target component carries most of the dark knowledge and is suppressed by the standard formulation.

Self-distillation and born-again networks. An interesting special case of logit distillation is *self-distillation*, where the student has the same architecture as the teacher. Born-Again Networks (BANs) [44] demonstrated that a student trained to mimic a converged teacher of identical architecture can *surpass* the teacher's performance, suggesting that KD serves as a form of regularization beyond its compression role. Subsequent work on online self-distillation [152] and deep mutual learning [154] extends this idea to collaborative training settings.

Chapters 3, 4, and 5 of this thesis contribute to the logit distillation paradigm, addressing the capacity gap, checkpoint search, and noisy teacher problems respectively through novel dynamic and continuation-based training objectives.

1.3.2 Intermediate-Layer Distillation

While logit distillation focuses on the final output distribution, *intermediate-layer distillation* (ILD), also called feature-based or representation-based distillation, seeks to transfer the rich internal representations computed by the teacher's hidden layers [107, 50]. The motivation is that intermediate representations encode hierarchical, geometric, and relational information about the data that is lost when compressed into a single output distribution.

Foundational methods. FitNets [107] introduced the concept of "hint layers": the student is trained to match the teacher's intermediate feature maps at one or more designated layers, using a learned regressor to bridge dimensionality mismatches. This intermediate supervision acts as an optimization scaffold, helping thin and deep students that are otherwise difficult to train. Zagoruyko and Komodakis [149] extended this idea to *attention transfer*, matching normalized attention maps rather than raw feature tensors, which is cheaper to compute and focuses on *where* the model looks rather than *what* it computes.

Relational and contrastive approaches. Rather than aligning per-example features, relational KD methods match structural relationships across examples. Park et al. [98] proposed Relational Knowledge Distillation (RKD), which distills pairwise distances and angles among examples in the teacher's embedding space. Tung and Mori [125] introduced similarity-preserving KD, matching the pairwise similarity matrices of teacher and student activations. Tian et al. [122] brought contrastive learning into the KD framework with Contrastive Representation Distillation (CRD), using an InfoNCE-style objective to transfer representation geometry while preserving mutual information between teacher and student features.

Modern feature distillation. Recent methods have focused on making intermediate supervision more *student-friendly*, reducing the semantic mismatch between teacher and student representations. SemCKD [18] introduces soft cross-layer associations that allow each student layer to attend to multiple teacher layers, mitigating the rigid one-to-one mapping assumption. ReviewKD [20] employs cross-level supervision where shallow teacher features supervise deeper student layers through hierarchical fusion modules. SimKD [17] simplifies the paradigm dramatically: it reuses the teacher's classifier head and trains the student's encoder via a single feature-alignment loss, achieving strong results with minimal overhead.

A separate line of work processes student features before matching. Masked Generative Distillation (MGD) [145] masks portions of the student's features and trains a generative block to reconstruct the teacher's features, forcing the student to learn a richer internal representation. DiffKD [61] applies a diffusion-based denoiser to student features before matching them to the teacher, treating the representational gap as a form of noise to be removed.

Applications in NLP and multimodal settings. Intermediate-layer distillation has been particularly influential in compressing Transformer-based language models. TinyBERT [69] employs a two-stage distillation pipeline that matches attention matrices, hidden states, and embedding layers between BERT teacher and student. Patient KD [117] aligns the student's hidden states to multiple teacher layers, "patiently" learning from multiple levels of the teacher's hierarchy. In multimodal settings, TinyCLIP [139] distills CLIP models by mimicking cross-modal affinity structure, achieving substantial compression with minimal degradation in zero-shot performance.

The layer-mapping challenge. A recurring practical challenge in ILD is determining *which* student layers should be matched to which teacher layers. Common strategies include uniform stride mapping, learned soft associations [18], and random mapping [55]. A recent controlled study [147] found that in many Transformer settings, the exact mapping strategy often matters less than expected, with even reverse mappings performing comparably, suggesting that the act of intermediate matching itself may dominate the marginal differences among mapping heuristics.

This thesis does not contribute directly to intermediate-layer distillation. However, ILD provides complementary supervision signals that can be combined with the logit-based methods developed in Chapters 3–5, and the survey above serves to position the thesis contributions within the broader KD landscape.

1.3.3 Data Distillation

Classical KD assumes that both teacher and student are trained on the same dataset and that the student has access to the teacher's internal logits or features. However, the emergence of massive proprietary language models accessible only through restrictive APIs (e.g., GPT-4, Claude) has fundamentally altered this assumption. In many practical settings, researchers have access only to the teacher's generated *text outputs*, not its internal representations. This has catalyzed the rise of *data distillation* (also called black-box KD or synthetic instruction tuning), a paradigm in which the teacher's knowledge is distilled *into the dataset* rather than through a loss function [141].

Sequence-level KD. The conceptual precursor to modern data distillation is sequence-level knowledge distillation, introduced by Kim and Rush [74] for neural machine translation. Rather than matching token-level probability distributions, the teacher generates complete translation sequences that serve as pseudo-targets for the student. This converts the generative KD problem into standard supervised sequence prediction, sidestepping the need for access to teacher internals. While effective, sequence-level KD can reduce output diversity and reinforce teacher biases [43].

The instruction-tuning paradigm. The release of large instruction-tuned models triggered a wave of data-distillation research aimed at transferring instruction-following capabilities to smaller open-weight models. Self-Instruct [134] demonstrated that a language model can generate its own instruction–response pairs from a small seed set, bootstrapping a large supervised dataset. Alpaca [121] applied this principle at scale, using GPT-3.5 to generate 52,000 instruction-following examples and fine-tuning a 7B LLaMA model that closely approached ChatGPT-level performance on many benchmarks. Vicuna [24] extended this to multi-turn dialogue by fine-tuning on user–ChatGPT conversations.

Orca [94] and Orca 2 [92] advanced the paradigm by eliciting *reasoning traces* from the teacher, prompting GPT-4 to produce step-by-step explanations alongside its answers. Training the student on these enriched rationales yields substantially better generalization than training on answers alone, highlighting that the *format* and *depth* of the distilled data are as important as its quantity.

Data curation and quality. A parallel line of work has emphasized that *data quality* is as important as model scale. The Phi series of models [53, 82] demonstrated that a carefully curated corpus of synthetic "textbook-quality" data, generated by a strong

teacher, can train a 1.3B-parameter model that outperforms much larger models on coding and reasoning benchmarks. UltraChat [37] and WaveCoder [146] further illustrate how targeted data curation from teacher models can unlock specialized capabilities.

Positioning Chapter 6. Chapter 6 of this thesis contributes to the data distillation paradigm. The *Efficient Citer* framework addresses the challenge of producing verifiable, citation-grounded answers in question-answering systems. By prompting a strong teacher model (ChatGPT) to generate answers that incorporate inline citations to provided source passages, and training compact student models (a 3B FID-style Flan-T5 model and a 13B LLaMA model) on the resulting dataset, Efficient Citer demonstrates that complex citation behaviors can be distilled into deployable models that are orders of magnitude more efficient than the teacher, both in parameter count and prompt length.

1.3.4 Divergence-Aware Distillation

The three paradigms above are, in varying degrees, *passive*: they minimize teacher–student divergence over a fixed or teacher-generated dataset without actively seeking out regions where the student's predictions are most deficient. *Divergence-aware distillation* (also called active or adversarial distillation) inverts this logic: it explicitly searches for inputs or generative states where the teacher and student disagree most, and focuses training on these high-divergence regions.

Theoretical motivation. The rationale for divergence-aware distillation follows from a simple observation: the KD training objective (Equation 1.2) only aligns the teacher and student at the observed data points. Because the training set is finite, there exist regions of the input space where the student's function may diverge arbitrarily from the teacher's, even when the KD loss is well minimized on the training data [66]. Classical KD has no mechanism to detect or correct these blind spots. Divergence-aware methods address this gap by augmenting the training set with samples specifically chosen to expose and correct the student's weaknesses.

Backward KD: divergence maximization in continuous domains. Jafari et al. [66] proposed *Backward Knowledge Distillation* (Backward KD), an iterative framework that alternates between two steps: a *maximization step*, in which training samples are perturbed along the gradient of the teacher–student divergence to generate auxiliary data points in

regions of maximum disagreement; and a *minimization step*, in which the student is trained via standard KD on the augmented dataset. Formally, given a scalar divergence measure $\mathcal{L}_{\text{BKD}} = \|S(x;\theta) - T(x)\|_2^2$, auxiliary samples are generated by gradient ascent on the input:

$$x' = x + \eta \nabla_x \|S(x;\theta) - T(x)\|_2^2, \tag{1.3}$$

where η is a perturbation step size. This procedure follows a minimax principle [66]: the maximization step finds the worst-case inputs for the student, and the minimization step ensures the student performs well even on these adversarial examples. Backward KD was shown to improve KD performance on MNIST, CIFAR-10, and the GLUE benchmark, and to be particularly effective in few-sample regimes [66].

For NLP applications, where inputs are discrete tokens and cannot be perturbed directly, Jafari et al. [66] proposed operating in the continuous embedding space: the gradient is taken with respect to the student's embedding vectors, and a linear transformation matrix maps the perturbed student embeddings to the teacher's embedding space, enabling cross-architecture divergence maximization.

Disagreement-aware methods in the LLM era. The principle of focusing on teacher–student disagreement has also been explored, in weaker forms, by recent work on LLM distillation. DASD [142] filters teacher-generated candidate responses based on sequence-level divergence, retaining responses where the teacher assigns high probability but the student does not. However, DASD's divergence measure is computed *after* responses are generated; there is no mechanism to steer generation toward high-divergence regions.

Positioning Chapter 7. Chapter 7 of this thesis proposes *Maximum Divergence Knowledge Distillation* (MDKD), which extends the Backward KD philosophy to the discrete autoregressive setting of modern LLMs. At each decoding step, MDKD samples candidate tokens from the teacher and accepts them with probability proportional to how much the student underestimates them, constructing training sequences that concentrate in regions of maximum teacher–student disagreement. Unlike GKD [1], MDKD does not rely on the student's own policy; unlike SKD [140], it does not merely correct student proposals; unlike DASD [142], it operates at the token level and can steer generation trajectories toward high-divergence regions during data construction. MDKD thus directly addresses the generative distribution mismatch problem (Section 1.2.5) through active, token-level divergence maximization.

1.4 Thesis Contributions

This thesis makes five primary contributions, spanning three of the four KD paradigms identified above. We organize them by paradigm to highlight the breadth and coherence of the research program.

1.4.1 Advancing Logit Distillation via Adaptive Optimization

The first three contributions address the optimization and capacity-based bottlenecks of classical logit distillation, proposing training objectives that dynamically adapt the supervisory signal to the student's learning state.

- **Contribution 1 (Chapter 3): Annealing Knowledge Distillation.** To address the capacity gap problem, we propose Annealing-KD, which introduces a dynamic temperature function that progressively transitions the student from highly smoothed teacher targets to the full-complexity teacher outputs. Building on known properties of the temperature-scaled softmax, we give a self-contained analysis showing that this mechanism creates an information-theoretic curriculum: the Shannon entropy of the target distribution is a monotonically increasing function of temperature (strictly, whenever the logits are not all equal); so lowering the temperature over training monotonically decreases the target's entropy, exposing the student to progressively sharper targets. Annealing-KD outperforms TAKD and vanilla KD on both image classification (CIFAR-10, CIFAR-100) and natural language understanding (GLUE benchmark) tasks. This work was published in [67].

- **Contribution 2 (Chapter 4): Progressive Knowledge Distillation (Pro-KD).** To address the checkpoint search problem, we propose Pro-KD, a progressive distillation framework in which the student does not learn from a fully converged teacher but instead follows the teacher's training trajectory. Combined with an adaptive temperature schedule, Pro-KD eliminates the need for expensive checkpoint searches and allows the student to benefit from the teacher's intermediate learning stages. Pro-KD obtains the best aggregate score among the evaluated baselines on GLUE, CIFAR-10 and CIFAR-100, and outperforms Vanilla KD and TAKD on SQuAD 1.1/2.0. This work was published in [106].

- **Contribution 3 (Chapter 5): Continuation Knowledge Distillation.** To address the noisy teacher problem, we propose Continuation-KD, inspired by continuation optimization and graduated non-convexity [2, 11]. Continuation-KD formulates

the distillation loss as an annealed hinge function with a dynamic margin that creates a flat region in which residuals below the margin produce no gradient, so that small fluctuations in the teacher's outputs are not transferred to the student. We give two exact structural results about the annealed objective (a gradient identity and a Hessian decomposition, each isolating the teacher's contribution in a single term scaled by the annealing factor), and show that Annealing-KD is a special case of the framework. Continuation-KD outperforms the vanilla-KD, TAKD, and Annealing-KD baselines on GLUE and CIFAR. This work was published in [65].

1.4.2 Data Distillation for Citation-Grounded Question Answering

The fourth contribution transitions to the black-box data distillation paradigm, leveraging teacher-generated data to imbue specialized capabilities into deployable models.

- **Contribution 4 (Chapter 6): Efficient Citer.** We address the challenge of producing verifiable, citation-grounded answers in question-answering systems. By constructing a citation dataset from MS MARCO using ChatGPT as the teacher, and fine-tuning a 3B FID-style Flan-T5 model (FIDCiter) and a 13B LLaMA model (LLaMACiter), we demonstrate that complex citation behaviors can be distilled into efficient models. Both students exceed ChatGPT on human-rated informativeness (FIDCiter on two of the four evaluation sets), and FIDCiter matches or exceeds it on in-domain citation quality, while both use instruction templates hundreds of times shorter; ChatGPT retains higher citation quality on the out-of-domain sets. This work was published in [118].

1.4.3 Active Divergence Distillation for Generative Models

The fifth contribution extends the philosophy of divergence-aware distillation into the discrete, autoregressive setting of modern LLMs.

- **Contribution 5 (Chapter 7): Maximum Divergence Knowledge Distillation (MDKD).** Building upon the continuous-domain Backward KD framework [66], we propose MDKD for autoregressive language models. MDKD introduces a rejection-sampling mechanism that, at each decoding step, samples candidate tokens from the teacher and accepts them with probability proportional to how much the student

underestimates them. This constructs training sequences concentrated in regions of maximum teacher–student disagreement, directly addressing the generative distribution mismatch problem. MDKD is simple to implement, orthogonal to the choice of divergence in the loss, and compatible with standard training infrastructure.

1.5 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 provides the background and related work, covering the mathematical foundations of knowledge distillation, the optimization theory underlying the methods developed in this thesis, and a detailed review of related work on backward-pass KD and KD for large-scale generative models.

Chapter 3 presents Annealing Knowledge Distillation (Annealing-KD), including its information-theoretic formulation, the proof that dynamic temperature scaling creates a strict entropy curriculum, and experimental validation on image classification and natural language understanding benchmarks.

Chapter 4 introduces Progressive Knowledge Distillation (Pro-KD), which addresses both the capacity gap and checkpoint search problems by training the student concurrently with the teacher along its optimization trajectory.

Chapter 5 presents Continuation Knowledge Distillation (Continuation-KD), which combines a continuation (graduated non-convexity) schedule with an annealed hinge-loss margin to limit the transfer of teacher noise, and gives two exact structural results about the resulting objective.

Chapter 6 shifts to the data distillation paradigm, detailing the Efficient Citer framework for training compact, citation-capable question-answering models from teacher-generated data.

Chapter 7 explores active divergence in generative models, presenting the Maximum Divergence Knowledge Distillation (MDKD) algorithm for autoregressive language models.

Chapter 8 concludes the thesis with a summary of contributions, a discussion of limitations, and directions for future research.

Chapter 2

Background and Related Work

2.1 Foundations of Knowledge Distillation

The fundamental premise of Knowledge Distillation (KD) relies on the observation that the true generalization capabilities of a highly optimized neural network are not captured entirely by its accuracy on hard, ground-truth labels. Instead, this knowledge is deeply embedded in the continuous, real-valued outputs produced by the network before they are collapsed into discrete predictions.

Early work by Buciluǎ et al. [14] demonstrated that it is possible to compress the function learned by a large ensemble of models into a single, smaller neural network by training the smaller model to reproduce the ensemble's aggregate outputs. This concept was later formalized and popularized as *Knowledge Distillation* by Hinton et al. [59], who introduced a unified framework to transfer what they termed "dark knowledge" from a cumbersome, high-capacity *teacher* model to a lightweight, efficient *student* model.

2.1.1 Dark Knowledge and Temperature Scaling

In standard classification tasks, a neural network produces a vector of pre-softmax logits $z \in \mathbb{R}^K$ for K classes. These logits are converted into a probability distribution q over the classes by the softmax function. To facilitate the transfer of dark knowledge, classical KD

introduces a *temperature* hyperparameter $\mathcal{T}$ into the softmax computation:

$$q_i(\mathcal{T}) = \frac{\exp\big(z_i/\mathcal{T}\big)}{\sum_{j=1}^{K} \exp\big(z_j/\mathcal{T}\big)}, \qquad i = 1, \ldots, K. \tag{2.1}$$

When $\mathcal{T} = 1$ the function recovers the standard softmax. In a well-trained, high-capacity teacher, this standard distribution tends to be extremely peaked: the probability of the correct class approaches 1 while all other classes are pushed toward 0. Although such a sharp distribution minimizes the empirical cross-entropy with the true label, it obscures the relative similarities among incorrect classes, information that a student could exploit to learn a more nuanced decision boundary.

Raising the temperature ($\mathcal{T} > 1$) *softens* the distribution. Probabilities that were negligible under the standard softmax are elevated, explicitly revealing structural relationships the teacher has learned about the data manifold. For instance, a softened distribution may reveal that the teacher considers an image of a dog to be far more similar to a cat than to a car. This rich inter-class similarity structure constitutes the "dark knowledge" and serves as a highly informative supervisory signal, guiding the student toward a more generalizable representation than one-hot labels alone can provide.

The temperature also admits a clean information-theoretic interpretation. Denoting the Shannon entropy of the temperature-scaled distribution by

$$\mathcal{H}\big(q(\mathcal{T})\big) = -\sum_{i=1}^{K} q_i(\mathcal{T}) \, \log q_i(\mathcal{T}), \tag{2.2}$$

one can show that $\mathcal{H}$ is a monotonically increasing function of $\mathcal{T}$: as the temperature rises, the entropy of the target distribution increases, and the supervisory signal becomes coarser. Conversely, lowering the temperature sharpens the distribution and reduces entropy, exposing finer-grained class distinctions. Two limiting cases are instructive:

- $\mathcal{T} \to 0$: the distribution converges to a Kronecker delta on the highest-logit class (equivalent to a one-hot hard label), and $\mathcal{H} \to 0$.
- $\mathcal{T} \to \infty$: the distribution converges to the uniform distribution $(1/K, \ldots, 1/K)$, and $\mathcal{H}$ attains its maximum value $\log K$.

This monotonic relationship between temperature and entropy is central to the information-theoretic curriculum developed in Chapter 3.

2.1.2 The KD Objective Function

The classical KD objective trains the student to simultaneously approximate the true data distribution (hard labels) and the teacher's softened output distribution (soft labels). Given a training set $\mathcal{D} = \{(x, y)\}$, let $z_t(x)$ and $z_s(x)$ denote the logit vectors produced by the teacher and student, respectively. The total loss is a convex combination of two terms:

$$\mathcal{L}_{\mathrm{KD}} = \alpha\, \mathcal{L}_{\mathrm{CE}}\Big(y,\ \sigma\big(z_s(x)\big)\Big) + (1-\alpha)\, \mathcal{T}^2\, D_{\mathrm{KL}}\Big(\sigma\big(z_t(x)/\mathcal{T}\big)\ \Big\|\ \sigma\big(z_s(x)/\mathcal{T}\big)\Big). \tag{2.3}$$

Here $\mathcal{L}_{\mathrm{CE}}$ is the standard cross-entropy loss between the ground-truth one-hot label y and the student's output at the standard temperature ($\mathcal{T} = 1$); D_{KL} is the Kullback–Leibler (KL) divergence between the softened distributions of teacher and student; and $\alpha \in [0, 1]$ is the weight on the hard-label (cross-entropy) term, following the same convention as Eq. 1.2 (so $\alpha \to 1$ recovers pure supervised training and $\alpha \to 0$ pure distillation).

A conventional detail is the $\mathcal{T}^2$ factor multiplying the KL term. Writing $q_{\mathcal{T}} = \sigma(z_t/\mathcal{T})$ and $p_{\mathcal{T}} = \sigma(z_s/\mathcal{T})$, the gradient of the KL term with respect to the *unscaled* student logits is $\frac{1}{\mathcal{T}}(p_{\mathcal{T}} - q_{\mathcal{T}})$. At high temperature the difference $p_{\mathcal{T}} - q_{\mathcal{T}}$ is itself of order $1/\mathcal{T}$ under the usual centered-logit approximation, so the soft-label gradient is approximately of order $1/\mathcal{T}^2$. The conventional $\mathcal{T}^2$ multiplier compensates for this high-temperature attenuation [59]; it is a practical normalization rather than an exact identity for arbitrary logits and temperatures.

2.1.3 Limitations of the Classical Formulation

While Equation 2.3 has become the *de facto* standard for model compression [50], it rests on several rigid, implicit assumptions: (i) the teacher is treated as a static, infallible oracle; (ii) the student possesses sufficient architectural and parametric capacity to smoothly approximate the teacher's function; and (iii) the training data is stationary. As modern deep learning scales toward massive generative models and increasingly heterogeneous architectural families, these assumptions break down in systematic ways. The subsequent sections examine these failure modes in detail, beginning with the most critical bottleneck of classical KD: the *capacity gap*.

2.2 The Capacity Gap

The classical KD formulation implicitly assumes that the student network possesses sufficient representational capacity to approximate the teacher's learned function. In ideal

conditions, a more accurate teacher should produce higher-quality dark knowledge, which should in turn yield a stronger student. However, extensive empirical studies have uncovered a counter-intuitive paradox: as the capacity and accuracy of the teacher increase beyond a certain threshold, the performance of the distilled student often *degrades* rather than improves [25]. This phenomenon is known as the *capacity gap*.

2.2.1 Theoretical Foundations: Generalized Distillation and VC Dimension

To understand the mathematical underpinnings of this degradation, KD must be viewed through the lens of statistical learning theory. Lopez-Paz et al. [85] unified KD and Vapnik's framework of *Learning Using Privileged Information* (LUPI) [127, 128] under a single formalism they called *Generalized Distillation*.

In standard supervised learning, the algorithm receives input–label pairs (x, y). In the LUPI framework, the learner also has access to *privileged information* x^* during training, additional explanations about each example that are unavailable at test time. In the context of KD, the teacher's soft targets play the role of this privileged information.

The three-actor analysis. Consider three actors: a student function $f_s \in \mathcal{F}_s$, a teacher function $f_t \in \mathcal{F}_t$, and the true target function $f \in \mathcal{F}$. Lopez-Paz et al. decompose the student's generalization error into two telescoping terms:

$$R(f_s) - R(f) = \underbrace{\big[R(f_s) - R(f_t)\big]}_{\text{student–teacher gap}} + \underbrace{\big[R(f_t) - R(f)\big]}_{\text{teacher–truth gap}}. \tag{2.4}$$

The framework is developed under a set of *rate assumptions* rather than as a universal theorem, and we state them in that form. Write $|\mathcal{F}|_C$ for a generic capacity measure of a function class. Following Lopez-Paz et al., assume that a student learning directly from hard labels obeys the *slow* rate

$$R(f_s) - R(f) \;\le\; \mathcal{O}\big(|\mathcal{F}_s|_C/\sqrt{n}\big) + \varepsilon_s, \tag{2.5}$$

where n is the number of training samples, and that the teacher and the student–teacher matching problem obey

$$R(f_t) - R(f) \;\le\; \mathcal{O}\big(|\mathcal{F}_t|_C/n\big) + \varepsilon_t, \qquad R(f_s) - R(f_t) \;\le\; \mathcal{O}\big(|\mathcal{F}_s|_C/n^{\alpha}\big) + \varepsilon_l, \tag{2.6}$$

with rate exponent $\alpha \in [\frac{1}{2}, 1]$. Here ε_t is the approximation error of the teacher with respect to the true function and ε_l that of the student with respect to the teacher. Combining these assumptions with the decomposition (2.4) gives

$$R(f_s) - R(f) \;\leq\; \mathcal{O}\left(\frac{|\mathcal{F}_s|_C + |\mathcal{F}_t|_C}{n^\alpha}\right) + \varepsilon_l + \varepsilon_t. \tag{2.7}$$

Distillation is favored only when this complete upper bound is smaller than the direct-learning bound (2.5), which depends jointly on α, the capacity terms, and the two approximation errors: it requires (i) the teacher's own capacity $|\mathcal{F}_t|_C$ to be manageable, (ii) the teacher's approximation error ε_t to be small, and (iii) the student–teacher approximation error ε_l to be small relative to the student's direct approximation error ε_s. The fast $\mathcal{O}(n^{-1})$ regime, $\alpha = 1$, requires additional conditions such as separability of the student–teacher matching problem; the availability of privileged information does not by itself make an arbitrary neural-network learning problem separable. The comparison is between two upper bounds, so a larger right-hand side does not establish that distillation is in fact worse in any particular instance.

The capacity gap through the lens of this bound. When distilling a massive teacher into a compact student, we have $|\mathcal{F}_t|_C \gg |\mathcal{F}_s|_C$. In this regime, the teacher's soft targets represent a highly complex function that may fall outside the student's representational capacity, causing the student–teacher approximation error ε_l to dominate. A large teacher–student mismatch may therefore enlarge the capacity term or the student–teacher approximation error, so that the distillation upper bound exceeds the corresponding direct-learning bound. This is a statement about statistical upper bounds; it does not by itself imply any particular parameter-space geometry or optimization failure mode for the student. The teacher's "explanation" is, in effect, too complex for the student to absorb.

2.2.2 Teacher–Assistant Knowledge Distillation (TAKD)

To bridge this capacity mismatch, Mirzadeh et al. [91] proposed *Teacher–Assistant Knowledge Distillation* (TAKD). TAKD introduces one or more intermediate networks, termed *teacher assistants* (TAs), whose capacities fall sequentially between those of the teacher and the student. The training proceeds in stages: the teacher first distills into the TA, and the TA in turn distills into the student. By reducing the capacity difference at each step, the target distributions remain within the expressivity bounds of the receiving network.

While TAKD successfully improves student performance and provides empirical validation of the capacity-gap theory, it suffers from two practical drawbacks. First, it is computationally expensive, requiring the sequential training of multiple deep networks. Second, the approximation errors introduced at each intermediate stage can compound, ultimately degrading the information that reaches the final student. These limitations motivate the search for methods that can bridge the capacity gap without physically inserting additional networks, a goal pursued by several contributions of this thesis.

2.2.3 Beyond the Parametric Capacity Gap

The capacity gap was originally characterized primarily through parameter count and VC dimension. However, as modern architectures diversify, a related but distinct challenge arises when the teacher and student belong to fundamentally different *architectural families* [21, 144, 132].

In such cross-architecture settings, the student may lack the structural mechanisms to directly mimic certain patterns in the teacher's output, for example, the sharp, position-specific attention spikes produced by softmax attention. Applying standard KL-divergence to this heterogeneous distillation task can lead to optimization instability because the target distribution exhibits high-frequency structure that the student's architecture cannot faithfully represent.

This thesis argues that overcoming both parametric and architectural capacity gaps does not necessarily require training intermediate networks (as in TAKD), but can instead be achieved through *dynamic smoothing of the target distribution*. As demonstrated in Chapter 3 (Annealing-KD) and Chapter 5 (Continuation-KD), controlling the complexity of the supervisory signal over the course of training acts as a form of optimization intervention, enabling direct distillation across large capacity gaps where classical KD degrades.

2.3 Theoretical Preliminaries

The failure modes of classical Knowledge Distillation analyzed above, the capacity gap and susceptibility to overly complex teacher targets, are fundamentally rooted in the geometry of the optimization landscape and the information density of the target distributions. To address these bottlenecks rigorously, this thesis introduces information-theoretic and geometric interventions. This section defines the mathematical preliminaries that serve as the foundation for the theoretical analyses developed in subsequent chapters.

2.3.1 Information Theory and Target Complexity

In Chapter 3 we construct an information-theoretic curriculum that controls the complexity of the supervisory signal presented to the student. The central quantity is *Shannon entropy* [31].

For a discrete probability distribution q over K classes, the Shannon entropy

$$\mathcal{H}(q) = -\sum_{i=1}^{K} q_i \log q_i \tag{2.8}$$

measures the expected uncertainty or information content of the distribution. In the context of KD, $\mathcal{H}(q)$ quantifies the concentration of the class-probability target that the student is asked to match; it does not by itself quantify the representational or optimization difficulty of fitting that target. A highly confident teacher produces a distribution approaching a Kronecker delta, which minimizes entropy ($\mathcal{H} \to 0$ when the maximum logit is unique; with r tied maxima the limit is uniform on those r classes and $\mathcal{H} \to \log r$) but provides a sparse and potentially unstable gradient signal. A completely uniform distribution maximizes entropy ($\mathcal{H} = \log K$) but carries no discriminative information.

When temperature scaling is applied to the teacher's logits (Equation 2.1), the entropy of the resulting distribution is a deterministic, monotonically increasing function of $\mathcal{T}$. One can verify this by expressing the entropy in terms of the partition function $Z(\beta) = \sum_j \exp(\beta\, z_j)$ with inverse temperature $\beta = 1/\mathcal{T}$, and showing that $d\mathcal{H}/d\mathcal{T} \geq 0$ through the variance of the logits under the distribution q (see Chapter 3 for the full derivation). Controlling this entropy provides an operational handle on the sharpness of the supervisory signal, enabling the construction of dynamic, entropy-based curricula that gradually expose the student to increasingly fine-grained targets.

2.3.2 Optimization Topology and Loss Landscapes

Deep neural networks are trained by minimizing a highly non-convex loss function $\mathcal{L}(\theta)$ over the network parameters $\theta \in \mathbb{R}^d$. The geometric structure of this loss landscape governs both the stability of optimization and the generalization properties of the converged model [80].

The local curvature of the landscape at any point θ is captured by the *Hessian matrix* $\nabla^2\mathcal{L}(\theta) \in \mathbb{R}^{d\times d}$, whose entries are the second-order partial derivatives of the loss with respect to the weights. The "sharpness" of a local minimum is tightly related to the *spectral norm* of the Hessian, denoted $\|\nabla^2\mathcal{L}(\theta)\|_2$ [73]. For a real symmetric Hessian this

is the largest *absolute* eigenvalue, $\|\nabla^2\mathcal{L}(\theta)\|_2 = \max_i |\lambda_i(\nabla^2\mathcal{L}(\theta))|$, which coincides with the largest algebraic eigenvalue only where the Hessian is positive semidefinite, such as at a twice-differentiable local minimum. We use the spectral norm consistently and do not interchange it with $\lambda_{\max}$.

A large spectral norm indicates a sharp, rugged landscape: the loss changes rapidly under small parameter perturbations, and gradient-based optimizers may oscillate or converge to narrow minima. A small spectral norm corresponds to a flatter landscape, which has been empirically linked to wider basins of attraction and better generalization in many settings [80, 73], though Hessian-based sharpness is parameterization-dependent and a smaller raw Hessian norm does not universally guarantee better generalization.

The relationship between target smoothing and this parameter-space curvature is more subtle than it first appears. For the soft-target cross-entropy loss, the Hessian with respect to the student *logits* is $\mathrm{diag}(p) - pp^\top$, which is independent of the target distribution q; softening the teacher target therefore does not directly change the logit-space Hessian. Effects in *parameter* space can still arise, through the student Jacobian, the residual-dependent second-derivative terms, and the parameter point that training reaches. Chapter 5 analyses this for the annealed squared-error objective, where the target *does* enter the parameter Hessian, and obtains an exact decomposition of that Hessian. The claim that target smoothing lowers landscape curvature in general is treated in this thesis as a motivating hypothesis, not an established fact.

2.3.3 Spectral Bias and the Frequency Principle

A complementary perspective on the interaction between teacher complexity and student learning comes from the *spectral bias* (also called the *Frequency Principle*) of neural networks [102]. Empirical and theoretical studies have shown that neural networks trained with gradient descent learn low-frequency components of the target function first, and only gradually capture higher-frequency details as training progresses.

Spectral bias concerns the Fourier content of the learned *input–output function*, not the entropy or "sharpness" of a class-probability vector, so the following is an analogy rather than a direct application of the result. By analogy, the fully-trained teacher's output can be thought of as combining a coarse component, the broad inter-class similarity structure, and a fine component, the class-specific confidence peaks. A student in the early stages of training fits the coarse component more readily; if the teacher's targets are sharp (low temperature), the fine component dominates and may act as noise that the student cannot yet exploit.

Under this analogy, high temperature suppresses the fine peaks and exposes only the coarse structure; as training progresses and the student acquires the capacity to resolve finer details, the temperature can be gradually reduced to reveal the full target distribution. This perspective *motivates* the *decreasing* temperature schedule adopted in Annealing-KD (Chapter 3) and the continuation strategy of Chapter 5; it is an interpretive analogy, not a formal justification for entropy annealing.

2.3.4 Continuation Methods and Graduated Non-Convexity

To navigate the highly non-convex landscapes generated during distillation, Chapter 5 (Continuation-KD) draws upon a class of numerical optimization techniques known as *continuation methods* [2], and in particular the strategy of *Graduated Non-Convexity* (GNC) [11, 93].

When a target objective $\mathcal{L}(\theta)$ is too rugged to optimize directly without becoming trapped in poor local minima, continuation methods embed it within a parameterized family of surrogate functions $\mathcal{L}_\lambda(\theta)$, where $\lambda \in [0, 1]$ acts as a smoothing parameter:

- At $\lambda = 0$, the surrogate $\mathcal{L}_0(\theta)$ is a highly smoothed, easily optimizable approximation of the original loss (often convex or nearly so).
- At $\lambda = 1$, the surrogate recovers the original, non-convex target objective: $\mathcal{L}_1(\theta) = \mathcal{L}(\theta)$.

Optimization proceeds by first finding the minimizer of the easy surrogate $\mathcal{L}_0$, and then gradually tracking the solution as λ is continuously increased toward 1. Under suitable regularity conditions, an isolated solution branch, and sufficiently small continuation steps, the optimizer may remain near that branch, bypassing geometric barriers (sharp ridges, narrow valleys) that plague direct optimization. No such tracking guarantee is established for the neural-network objectives used in this thesis.

In the context of KD, the smoothing parameter λ can be realized through the temperature $\mathcal{T}$ or through explicit target scaling factors. Chapter 5 uses this connection to design Continuation-KD, and analyses the structure of the resulting annealed objective.

2.4 Distillation in the Generative Era

Classical Knowledge Distillation was developed and optimized primarily for classification tasks, where the model maps an input to a static, fixed-dimensional categorical distribu-

tion. However, the deep learning landscape has shifted fundamentally toward autoregressive Large Language Models (LLMs) [141]. In these generative models, the output is a sequential trajectory of discrete tokens, each conditioned on the previously generated tokens. This paradigm shift demands a re-evaluation of how knowledge is transferred, giving rise to distinct distillation settings and new failure modes.

2.4.1 White-Box vs. Black-Box Distillation

Depending on the level of access to the teacher, modern distillation techniques fall into two broad categories:

- **White-box distillation.** The classical setting in which the practitioner has full access to the teacher's architecture, weights, and internal states. This permits direct minimization of a divergence on the output logits (*logit distillation*) or alignment of internal hidden states and attention matrices (*intermediate-layer distillation*). While highly effective for structural alignment, white-box KD is often memory-prohibitive for foundation models exceeding 100 billion parameters.
- **Black-box distillation.** A setting necessitated by the deployment of state-of-the-art LLMs behind proprietary APIs (e.g., GPT-4). The teacher's logits, weights, and gradients are entirely inaccessible; the student can observe only the final text generated by the teacher in response to a given prompt.

2.4.2 Data Distillation and Synthetic Instruction Tuning

The restriction to black-box access has driven the rapid adoption of *data distillation*, also referred to as *synthetic instruction tuning* [141], building on the instruction-tuning paradigm introduced by FLAN [137]. Instead of matching probability distributions, data distillation treats the proprietary teacher purely as a data synthesizer. By carefully designing prompts, researchers can elicit high-quality, complex outputs, step-by-step reasoning traces, code implementations, detailed factual explanations, from the teacher. A smaller, open-weight student (e.g., LLaMA [124] or Flan-T5 [27]) is then trained via standard supervised fine-tuning on this synthetic corpus.

Pioneering work such as Alpaca [121] demonstrated that this approach can effectively transfer the instruction-following capabilities of massive proprietary models into compact students at a fraction of the training cost. Subsequent efforts, including Self-Instruct [134],

Vicuna [24], and Orca [94], refined the data synthesis pipeline by incorporating richer prompting strategies, multi-turn dialogues, and reasoning traces. This black-box data-synthesis methodology serves as the foundational mechanism for the citation-grounded research presented in Chapter 6 (Efficient Citer).

2.4.3 Exposure Bias in Autoregressive Generation

Despite the success of data distillation and generative fine-tuning, autoregressive KD suffers from a fundamental structural flaw known as *exposure bias* [7, 105].

During training, the student model is conditioned on the gold-standard, ground-truth prefix sequence, a regime known as *teacher forcing*. At each time step, the student predicts the next token given the true preceding tokens (or the teacher's soft distribution over the next token). However, at inference time the gold-standard prefix is unavailable: the student must condition each prediction on the tokens it has itself generated. Because the student is imperfect, it will inevitably sample suboptimal tokens. Once such a deviation occurs, the student enters a region of the sequence space that it was never exposed to during training. Because teacher-forced KD never taught the student to recover from its own errors, these deviations compound over the course of generation, leading to degraded coherence, repetitive loops, and hallucinations in long-form output.

The existence of exposure bias reveals a fundamental limitation of passive, dataset-based distillation. A complete remedy would expose the student to its own generative deviations during training; short of that, it motivates a shift from passive dataset matching toward active, divergence-aware frameworks that target teacher–student disagreement, described in the next section.

2.5 Active and Divergence-Based Distillation

The failure modes of classical KD, including exposure bias, overfitting to average-case behavior, and neglect of boundary cases, stem from its fundamentally *passive* nature. In the standard paradigm, the student acts as a static receiver of teacher outputs, minimizing an average divergence metric over a fixed, pre-defined dataset. To build models that generalize robustly under distributional shift and during complex autoregressive generation, distillation must transition into an *active* paradigm that explicitly targets the regions of the input space where teacher and student disagree the most.

2.5.1 The Limitation of Static Datasets

Classical logit distillation minimizes the expected KL divergence over the empirical training distribution. Neural networks, however, tend to exhibit high confidence on typical, well-represented examples while failing on boundary cases or out-of-distribution inputs. Because static datasets are naturally dominated by such "easy" examples, the distillation loss is driven primarily by samples where the student already closely matches the teacher. Consequently, the student receives very little gradient signal in precisely the regions where its decision boundaries are most flawed. Standard KD lacks a structural mechanism to dynamically discover and correct the student's specific weaknesses.

2.5.2 Continuous-Domain Precursors: Backward Knowledge Distillation

The philosophy of explicitly targeting teacher–student disagreement was formalized in continuous domains through our prior work on *Backward Knowledge Distillation* (Backward KD) [66].

The central idea is a *minimax* procedure. Given an input sample x and the corresponding outputs from the teacher ($f_t(x)$) and the student ($f_s(x)$), Backward KD defines a divergence objective that measures the discrepancy between the two models. Crucially, rather than immediately minimizing this divergence with respect to the student's weights, as in standard KD, the method first performs a *maximization* step: it computes the gradient of the divergence with respect to the *input* itself,

$$x' = x + \eta \, \nabla_x \, \mathrm{Div}\big(f_t(x),\, f_s(x)\big), \tag{2.9}$$

and uses this gradient to perturb the input in the direction that maximally increases the disagreement between teacher and student. The perturbed samples x' are then labeled by the teacher and added to the training set. In the subsequent *minimization* step, the student is trained via standard KD on the augmented dataset, which now includes data from the regions where the student's representations diverge most from the teacher's.

This adversarial augmentation forces the student to align its decision boundaries with the teacher precisely in the regions where it previously struggled, transforming KD from a passive matching exercise into an active exploration of the student's blind spots. Experiments on MNIST, CIFAR-10, and the GLUE benchmark confirmed that backward-pass supervision improves student performance, particularly for narrow and shallow student networks [66].

2.5.3 Transition to the Discrete Generative Domain

While Backward KD established that actively targeting divergence yields superior alignment, applying this exact mechanism to LLMs is problematic. The continuous-domain gradient perturbation in Equation 2.9 requires a differentiable input space (e.g., continuous image pixels or embedding vectors). LLMs, however, operate on sequences of discrete tokens, rendering direct gradient-based input perturbation intractable without complex, and often unstable, continuous relaxations.

To bring the active-divergence philosophy to autoregressive generation, the continuous gradient perturbation must be replaced by a discrete, *generation-based* mechanism. One route is on-policy: sample sequences from the student itself and supervise where its predictions diverge from the teacher (as in GKD [1]). A second route keeps generation teacher-driven but biases *which* teacher-supported tokens are selected toward those the student underweights.

The *Maximum Divergence Knowledge Distillation* (MDKD) framework of Chapter 7 takes the second route. At each step MDKD draws candidate tokens from the teacher's (nucleus-filtered) distribution and preferentially accepts those to which the student assigns much less probability than the teacher, so the resulting sequences are teacher-supported at every position but concentrated on the student's blind spots. This translates the divergence-driven logic of Backward KD into the discrete setting and targets the *distributional-coverage* aspect of the training–inference mismatch: the student is supervised on teacher-supported trajectories it currently underweights. Because the prefixes remain teacher-generated rather than student-generated, MDKD does not by itself expose the student to its own inference-time errors; combining it with student-generated prefixes for more direct exposure-bias mitigation is noted as future work in Chapter 7 and Chapter 8.

2.6 Summary

This chapter has established the foundational concepts, theoretical limitations, and mathematical preliminaries necessary to contextualize the contributions of this thesis.

We began by formalizing classical Knowledge Distillation, its objective function, the role of temperature scaling and dark knowledge, and its implicit assumptions. We then examined the *capacity gap*: through the lens of Generalized Distillation and VC-dimension theory, we showed how a large teacher–student capacity mismatch can cause the LUPI-based acceleration guarantees to break down, leading to degraded student performance.

We reviewed TAKD as the prevailing mitigation strategy and noted its computational limitations.

To address these non-convex optimization bottlenecks without relying on intermediate networks, we introduced the mathematical toolkit that underpins the first half of this thesis: Shannon entropy as a measure of target complexity, the Hessian spectral norm as a quantifier of landscape sharpness, the spectral bias of neural networks as a lens on the frequency structure of learning, and Graduated Non-Convexity as a framework for navigating rugged loss landscapes through systematic smoothing.

Finally, we mapped the transition of distillation into the generative era. We detailed how the rise of proprietary LLMs has given prominence to black-box data distillation, and we exposed the fundamental flaw of exposure bias in autoregressive generation. By reviewing the continuous-domain minimax mechanics of Backward KD, we established the conceptual necessity of shifting from passive dataset matching to active, divergence-aware frameworks. The MDKD method of Chapter 7 carries this idea into the discrete autoregressive domain, targeting the distributional-coverage aspect of the mismatch by biasing teacher-driven generation toward the student's blind spots; a fully on-policy treatment of exposure bias is left as future work.

Equipped with this theoretical and historical context, the following chapters present the core methodological contributions of this thesis, beginning in Chapter 3 with Annealing Knowledge Distillation.

Chapter 3

Annealing Knowledge Distillation

3.1 Overview

In Chapter 2 we established the formal foundations of Knowledge Distillation: its objective function (Eq. 2.3), the role of temperature scaling in revealing dark knowledge, and the information-theoretic interpretation of entropy as a measure of target complexity. We also examined a central limitation of the classical KD framework, the *capacity gap*, which arises when the representational power of the teacher far exceeds that of the student. Through the lens of VC-dimension theory and the Generalized Distillation framework [85], we showed that this gap can cause the estimation error on the left-hand side of the LUPI bound to dominate the approximation error advantage, which can degrade the student's performance rather than improving it.

As discussed in Section 2.2, the Teacher Assistant Knowledge Distillation (TAKD) method [91] was proposed to mitigate this gap by introducing intermediate networks of decreasing capacity. However, TAKD incurs substantial computational overhead, since each teacher assistant must be trained independently. Furthermore, sequential distillation through multiple intermediaries results in error accumulation: each assistant introduces its own approximation error, which propagates through the chain to the final student.

In this chapter, we tackle the capacity-gap problem from a fundamentally different angle. Rather than physically reducing the teacher's capacity through auxiliary networks, we control the complexity of the supervisory signal transmitted by the teacher, operationalized through the entropy of its output distribution. A high-capacity teacher, having been trained to convergence, produces sharp, low-entropy softmax distributions that encode

highly complex supervisory signals. When a low-capacity student must match these sharp targets from the onset, it can create an optimization bottleneck that impedes effective knowledge transfer.

We propose *Annealing Knowledge Distillation* (Annealing-KD), a technique that mitigates the effects of this gap by constructing a principled information curriculum motivated by controlling the entropy of the target distribution. By introducing a dynamic temperature term applied exclusively to the teacher's logits, Annealing-KD acts analogously to a "low-pass filter" on the teacher's targets early in training. As training progresses, the temperature is gradually decreased, smoothly transitioning from high-entropy soft labels to low-entropy soft labels. This creates a structured curriculum of target complexity matched to the student's learning dynamics.

The contributions of this chapter are summarized as follows:

1. We propose the novel Annealing-KD solution to the KD *capacity-gap problem*, based on modifying the KD loss and introducing a dynamic temperature function to make the student training gradual and smooth.

2. We provide a self-contained information-theoretic analysis of how the temperature controls the Shannon Entropy of the target distribution (entropy bounds and monotonicity in temperature, which are known properties of the temperature-scaled softmax, with references given in Section 3.3.2), and use it to motivate the curriculum interpretation of our method.

3. We apply our technique to ResNet-8 and plain CNN models on both CIFAR-10 and CIFAR-100 image classification tasks, and to the natural language inference task using different BERT-based models such as DistilRoBERTa and BERT-Small on the GLUE benchmark, where it outperforms the vanilla-KD and TAKD baselines. We further show the effectiveness of Annealing-KD in a hybrid-LLM upcycling task.

4. Our technique is simple, architecture agnostic, and can be applied on top of different variants of KD.

3.2 Related Work

3.2.1 The Capacity Gap and Existing Solutions

The capacity-gap problem in KD, whereby a stronger teacher does not necessarily yield a stronger student, has been studied from both theoretical and empirical perspectives. We

refer the reader to Section 2.2 of the Background chapter for a detailed treatment of this phenomenon, including the generalized-distillation rate analysis by Lopez-Paz et al. [85] and the empirical findings of Mirzadeh et al. [91]. In brief, the standard KD loss function (Eq. 2.3) minimizes the KL divergence between the softened output distributions of the student and teacher networks, with a temperature parameter $\mathcal{T}$ controlling the softness of these distributions. The capacity-gap manifests when increasing the teacher's capacity causes the distillation upper bound to exceed the corresponding direct-learning bound, effectively negating the advantages of a more accurate teacher.

As reviewed in Section 2.2, the prevailing solution to this problem, TAKD [91], introduces one or more intermediate teacher-assistant networks to bridge the capacity mismatch between the teacher and student. While TAKD does alleviate the gap, it comes with the considerable downsides of additional computational cost and error accumulation through the chain of intermediaries. These limitations motivate an alternative approach that achieves a smooth knowledge transfer without the need for auxiliary networks, which is the goal of the Annealing-KD technique proposed in this chapter.

3.2.2 Annealing in Knowledge Distillation

Clark et al. [28] proposed an annealing idea in their Born-Again Multi-task (BAM) paper, to train a multitask student network using distillation from some single-task teachers. They introduce a so-called teacher annealing scheme to distill from a dynamic weighted mixture of the teacher prediction and the ground-truth label. In this regard, the weight of teacher's prediction is gradually reduced compared to the weight of ground-truth labels during training. Therefore, early in training, the student model mostly learns from the teacher and later on, it learns mostly from target labels.

However, our Annealing-KD is different from [28] in different aspects. First, the introduced annealing term in BAM is conceptually different from our annealing. While in BAM, teacher annealing controls the contribution of the teacher dark knowledge compared to the ground-truth labels during training, our Annealing-KD is only applied to the teacher output in the KD loss to address the capacity-gap problem between the teacher and student networks. Second, the way we do annealing in our technique is through the temperature parameter and not by controlling the contribution of the teacher and ground-truth labels. Third, BAM falls into another category of knowledge distillation which focuses on improving the performance of the student model and not compressing it.

To address the capacity-gap problem, we modify the KD training procedure and propose annealing the temperature parameter of the teacher in order to simplify its function for

the student in the early stages of training and then toward the end of the training, by changing this parameter, the teacher provides a more complex function to the student. Also, to address the limitation concern of [28] paper, we propose a two-stage training algorithm instead of annealing the λ parameter. The details of this method is provided in the next section.

3.3 Method: Annealing Knowledge Distillation

In this section, we describe our Annealing-KD technique and show the rationale behind it. First, we start by formulating the problem and visualizing our technique using an example for a better presentation. Then, we give an information-theoretic analysis to understand how our dynamic temperature mechanism controls the entropy of the teacher's output distribution, and interpret its role as a curriculum for the student.

KD defines a two-objective loss function (i.e. the $\mathcal{L}_{KD}$ and $\mathcal{L}_{CE}$ terms in Equation 2.3) to minimize the distance between student predictions and soft labels and hard labels simultaneously. Annealing-KD adds no inference-time computation and, unlike TAKD, no auxiliary networks; its training cost is the total number of Stage I and Stage II student updates, which we report explicitly for each experiment. Annealing-KD breaks the KD training into two stages: Stage I, gradually training the student to mimic the teacher using our Annealing-KD loss $\mathcal{L}_{\text{KD}}^{\text{Annealing}}$; Stage II, fine-tuning the student with hard labels using $\mathcal{L}_{CE}$. We can define the loss function of our method as following.

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{KD}}^{\text{Annealing}}(i), & \textbf{Stage I: } 1 \leq i \leq n_1 \\ \mathcal{L}_{CE}(i - n_1), & \textbf{Stage II: } n_1 < i \leq n_1 + n_2 \end{cases} \tag{3.1}$$

In the above equation, i indicates the epoch index in the training process with the max epoch number of n for stage I, $\mathcal{T}_i$ represents the temperature value at i^{th} epoch, $\mathcal{L}_{CE}$ is the standard cross-entropy loss (see Equation 2.3), and at each epoch i, $\mathcal{L}_{\text{KD}}^{\text{Annealing}}(i)$ is a teacher temperature-scaled loss defined as $\mathcal{L}(z_s(x),\ z_t(x) \cdot \Phi(\mathcal{T}_i))$, where $\Phi(\mathcal{T})$ is the *annealing function* and $\mathcal{L}$ can be instantiated as either an MSE objective in logit space or a KL-divergence objective in probability space:

$$\begin{aligned} \mathcal{L}_{\text{MSE}}^{\text{Annealing}}(i) &= ||z_s(x) - z_t(x) \cdot \Phi(\mathcal{T}_i)||_2^2 \\ \mathcal{L}_{\text{KL}}^{\text{Annealing}}(i) &= D_{\text{KL}}(\sigma(z_t(x) \cdot \Phi(\mathcal{T}_i)) \parallel \sigma(z_s(x))) \end{aligned} \tag{3.2}$$

where $\sigma(\cdot)$ denotes the softmax function, and the annealing function is defined as:

$$\Phi(\mathcal{T}) = 1 - \frac{\mathcal{T} - 1}{\tau_{\max}}, \quad 1 \leq \mathcal{T} \leq \tau_{\max}, \quad \mathcal{T} \in \mathbb{N} \tag{3.3}$$

The MSE variant operates directly on logits and is suited for logit-matching tasks. The KL variant operates in probability space and is suited for distribution-matching tasks such as autoregressive language model distillation. The two variants also differ in one further respect. Logits are determined only up to an additive constant, since the softmax is unchanged by adding the same scalar to every coordinate. Scaling by $\Phi(\mathcal{T}_i)$ does not remove such an offset but rescales it, $\Phi(\mathcal{T}_i)(z_t + c\mathbf{1}) = \Phi(\mathcal{T}_i)z_t + \Phi(\mathcal{T}_i)c\,\mathbf{1}$, so the KL variant is invariant to it whereas the MSE variant is not: the MSE objective depends on the teacher's logit gauge. The experiments reported here use the uncentered logits produced by the stated checkpoints; evaluating centered or standardized logits is left to future work.

By squashing the teacher's target logits toward zero early in training ($\Phi(\mathcal{T})$ at its smallest value $1/\tau_{\max}$), both formulations explicitly limit the variance of the supervisory signal, mimicking a maximum-entropy uniform distribution. The annealing function $\Phi(\mathcal{T})$ can be replaced with any monotonically decreasing function $\Phi : \{1, \ldots, \tau_{\max}\} \to [1/\tau_{\max},\, 1]$. In stage I of our training, initially we set $\mathcal{T}_1 = \tau_{\max}$ (which leads to the most softened version of the teacher outputs because $\Phi(\mathcal{T}_1) = \frac{1}{\tau_{\max}}$) and decrease the temperature during training as the epoch number grows (that is $\mathcal{T} \to 1$ while $i \to n$). Training in stage I continues until $i = n, \mathcal{T} = 1$, for which $\Phi(\mathcal{T}_n) = 1$ and we get the sharpest version of z_t without any softening.

Therefore, our Annealing-KD mitigates the effects of the capacity gap between the student and teacher models by introducing the dynamic temperature term (that is the annealing function $\Phi(\mathcal{T})$) in the stage I of training. This way our Annealing-KD method is able to smoothly transfer the teacher's knowledge to the student model via a gradual transition over soft-labels generated by the teacher at different temperatures.

Algorithm 1 explains the proposed method in more detail. It takes the student S, the frozen teacher $\mathcal{F}$, training and validation data D_{train} and D_{val}, the maximum annealing index $\tau_{\max}$, and the Stage-I and Stage-II epoch counts n_1 and n_2. Stage I runs n_1 epochs in total: the annealing index q_i is held for $k = \lfloor n_1/\tau_{\max} \rfloor$ consecutive epochs before decreasing, so the schedule starts at $\Phi = 1/\tau_{\max}$ and reaches $\Phi = 1$ on the final Stage-I epochs. We write the teacher as $\mathcal{F}$ to keep it distinct from the annealing index. Stage II resumes from the best Stage-I checkpoint and fine-tunes on hard labels for n_2 epochs; both stages select checkpoints on D_{val}.

In this section, we proposed an approach to alleviate the gap between the teacher and student models as well as reducing the sharpness of the KD loss function. In our model,

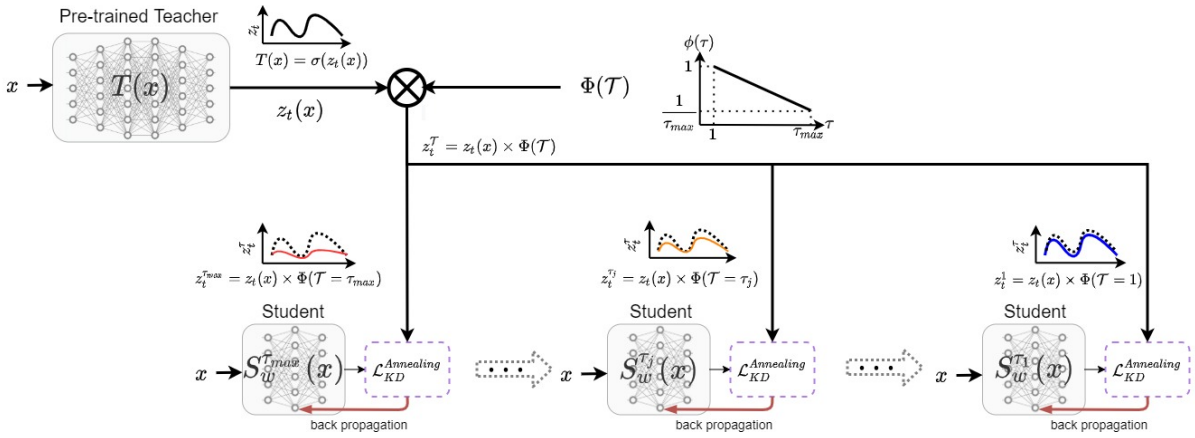


Figure 3.1: Illustrating the Stage I of the Annealing-KD technique. Given a pre-trained teacher network, we can derive the annealed output of the teacher at different temperature using the annealing function $\Phi(\mathcal{T})$. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$.

instead of pushing the student network to learn a complex teacher function from scratch, we start training the student from a softened version of the teacher and we gradually move toward the original teacher outputs through our annealing process.

3.3.1 Example

For better illustration of our proposed method, we designed a simple example to visualize different parts of our Annealing-KD algorithm. In this regard, we defined a simple regression task using a simple 2D function. This function is a linear combination of three sinusoidal functions with different frequencies $f(x) = sin(3\pi x) + sin(6\pi x) + sin(9\pi x)$. We randomly sample some points from this function to form our dataset (Figure 3.2-(a)). Next, we fit a simple fully connected neural network with only one hidden layer and the sigmoid activation function to the underlying function of the defined dataset. The teacher model is composed of 100 hidden neurons and trained with the given dataset. After training, the teacher is able to get very close to training data (see the green curve in Figure 3.2-(a)). We plot the annealed output of the teacher function in 10 different temperatures in Figure. 3.2-(b). Then, a student model with 10 hidden neurons is trained once with regular KD (Figure 3.2-(f)) and once with our Annealing-KD (Figures. 3.2-(c, d, e) depicts the student output at temperatures 10, 5, and 1 during the Annealing-KD training). As it is shown in these figures, Annealing-KD guides the student network gradually until it gets

Algorithm 1 Annealing Knowledge Distillation

function ANNEALING-KD(S, $\mathcal{F}$, D_{train}, D_{val}, $\tau_{\max}$, n_1, n_2)
  $k \leftarrow \max\{1,\ \lfloor n_1/\tau_{\max} \rfloor\}$ ▷ epochs held at each annealing level
  ▷ Stage I: annealed distillation
  for $i = 1$ `to` n_1 **do**
    $q_i \leftarrow \max\{1,\ \tau_{\max} - \lfloor (i-1)/k \rfloor\}$
    $\Phi \leftarrow 1 - \frac{q_i - 1}{\tau_{\max}}$
    for each mini-batch X from D_{train} **do**
      TRAIN-ANNEALING(S, $\mathcal{F}$, X, Φ)
    end for
    SAVE-BEST-CHECKPOINT(S, D_{val})
  end for
  $S \leftarrow$ LOAD-BEST-CHECKPOINT
  ▷ Stage II: fine-tuning on hard labels
  for $j = 1$ `to` n_2 **do**
    for each mini-batch (X, Y) from D_{train} **do**
      TRAIN-FINE-TUNE(S, X, Y)
    end for
    SAVE-BEST-CHECKPOINT(S, D_{val})
  end for
  $S \leftarrow$ LOAD-BEST-CHECKPOINT
  return S
end function

to a good approximation of the underlying function and it can match the teacher output better than regular KD.

3.3.2 Rationale Behind Annealing KD: An Information-Theoretic Curriculum

Standard Knowledge Distillation (KD) minimizes the divergence between the output distributions of a high-capacity teacher network and a lower-capacity student network. However, as demonstrated by Mirzadeh et al. (2019), when the capacity gap between the teacher and student is sufficiently large, the distillation process degrades. To understand *why* this degradation occurs and how Annealing-KD addresses it, we frame the distillation process through the lens of Information Theory and optimization. Throughout this section, we op-

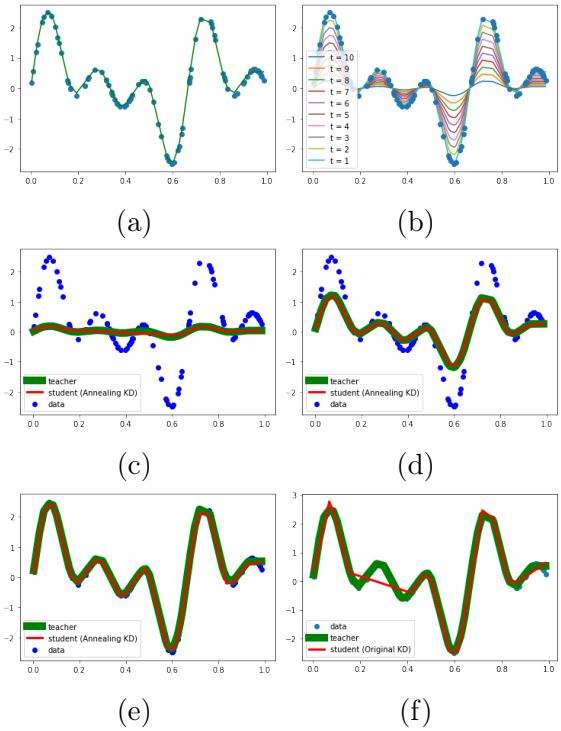


Figure 3.2: (a) Data samples and trained teacher. (b) Annealed teacher in different temperatures. (c) Student after matching to annealed teacher in $\mathcal{T} = 10$. (d) Student after matching to annealed teacher in $\mathcal{T} = 5$. (e) Student after matching to $\mathcal{T} = 1$. (f) Student trained without KD.

erationalize the notion of *target complexity* through the Shannon Entropy of the teacher's output distribution, which quantifies the uncertainty in the supervisory signal presented to the student.

A trained, high-capacity teacher network learns highly confident, sharp decision boundaries to separate difficult edge cases. These sharp, low-entropy softmax distributions impose demanding optimization targets on the student. Conceptually, in signal processing terms, attempting to fit these near-discontinuous distributions acts analogously to fitting a "high-frequency" signal. A known phenomenon in deep learning, *Spectral Bias*, dictates that neural networks naturally learn smooth, low-frequency spatial functions easily, but struggle to model high-frequency spatial variations. While Spectral Bias formally applies to the Fourier spectrum of the input-output mapping rather than the output entropy itself, the analogy holds operationally: a low-capacity student lacks the representational power

to directly model the sharp spatial gradients induced by these low-entropy targets early in training, creating a significant optimization bottleneck.

Rather than introducing intermediate Teacher Assistant networks to address the capacity gap, Annealing Knowledge Distillation (AKD) natively constructs an information bottleneck. By dynamically controlling the temperature parameter T applied exclusively to the teacher's logits, AKD acts as a dynamic low-pass filter. It creates a structured curriculum of target complexity, matching the entropy of the supervisory signal to the learning dynamics of the student.

Target Entropy and the Information Bottleneck

Notation. In the method description above, the annealing schedule is parameterized by $\mathcal{T}$ and an arbitrary monotonically decreasing scaling function $\Phi(\mathcal{T})$ applied to the teacher's logits (Eq. 3.2). For the theoretical analysis that follows, we adopt the canonical temperature parameterization T and write the teacher's softmax as $\exp(v_i/T)$, which corresponds to setting $\Phi = 1/T$. This choice simplifies the derivations without loss of generality: because the entropy of the softmax distribution depends on the logits only through the product $\Phi \cdot v_i$, the monotonicity results derived below hold for any positive, monotonically decreasing scaling function $\Phi(\mathcal{T})$ via the chain rule (see Remark 3.3).

Let $v \in \mathbb{R}^K$ denote the pre-softmax logits produced by the teacher network for a K-class classification task. The temperature-scaled probability distribution of the teacher, $q(T)$, is defined for the i-th class as:

$$q_i(T) = \frac{\exp(v_i/T)}{\sum_{j=1}^{K} \exp(v_j/T)} \tag{3.4}$$

The complexity of the information provided by the teacher can be formally quantified using Shannon Entropy, denoted as $\mathcal{H}(q_T)$:

$$\mathcal{H}(q_T) = -\sum_{i=1}^{K} q_i(T) \log q_i(T) \tag{3.5}$$

When a fixed, low temperature $T = 1$ is used (standard KD), the teacher's distribution is sharp, and $\mathcal{H}(q_T)$ is minimized. For a low-capacity student, attempting to match this low-entropy target distribution early in training can lead to overfitting and optimization instability. Conversely, raising T acts as Maximum Entropy Regularization. We establish the theoretical limits of this target entropy as follows:

Theorem 3.1 (Entropy Bounds of the Teacher Distribution)**.** *As the temperature $T \to \infty$, the target distribution maximizes entropy. As $T \to 0$, the target distribution minimizes entropy.*

The limiting behavior of the softmax function at high and low temperature is noted in [45], and the corresponding behavior of the Boltzmann (Gibbs) distribution, namely the uniform distribution of maximal entropy as $T \to \infty$ and the concentration on the ground state(s) with entropy $k_B \ln g$ (zero for a unique ground state) as $T \to 0$, is classical in statistical physics and thermodynamics [72]. We include a self-contained proof below for completeness.

Proof. As $T \to \infty$, the scaled logits $\frac{v_i}{T} \to 0$. Consequently, $\exp\left(\frac{v_i}{T}\right) \to 1$, yielding a uniform distribution $q_i(T) = \frac{1}{K}$. The entropy reaches its theoretical maximum:

$$\lim_{T\to\infty} \mathcal{H}(q_T) = -\sum_{i=1}^{K} \frac{1}{K} \log\left(\frac{1}{K}\right) = \log K$$

Conversely, as $T \to 0$, assuming a unique maximum logit $v_{\max}$, the distribution converges to a Kronecker delta (a one-hot vector). The entropy of this deterministic distribution is absolutely minimized:

$$\lim_{T\to 0} \mathcal{H}(q_T) = 0$$

If instead exactly r logits attain the maximum, the zero-temperature limit is uniform on those r classes and $\lim_{T\to 0} \mathcal{H}(q_T) = \log r$. □

The Core Curriculum Theorem: Strict Monotonicity

To guarantee that annealing the temperature creates a reliable curriculum, we must prove that dynamically lowering T from an initial state $T_{\max}$ to a final state of 1 results in a strictly monotonic decrease in target entropy. The theorem below is a statement about the continuous temperature argument. The implemented schedule holds the annealing index fixed within a block of epochs and then decrements it, so along the training run the target entropy is stepwise: constant within a block and strictly decreasing whenever the level changes.

Theorem 3.2 (Monotonicity of Target Entropy)**.** *The Shannon entropy of the teacher's temperature-scaled distribution, $\mathcal{H}(q_T)$, is a monotonically non-decreasing function of the temperature T. Furthermore, it is strictly monotonically increasing provided the pre-softmax logits are not identically equal.*

This property is known in the literature [32, 130]; we include a self-contained proof below for completeness.

Proof. To prove $\frac{\partial \mathcal{H}}{\partial T} \geq 0$, we utilize the inverse temperature $\beta = \frac{1}{T}$. The softmax probability for the i-th class is defined as $q_i(\beta) = \frac{\exp(\beta v_i)}{Z(\beta)}$, where the partition function is $Z(\beta) = \sum_{j=1}^{K} \exp(\beta v_j)$. First, we express the entropy in terms of β. Taking the natural logarithm of the softmax probability yields $\log q_i(\beta) = \beta v_i - \log Z(\beta)$. Substituting this into the Shannon entropy equation:

$$\mathcal{H}(\beta) = -\sum_{i=1}^{K} q_i(\beta)(\beta v_i - \log Z(\beta))$$

By distributing the sum and noting that $\sum q_i(\beta) = 1$, we get:

$$\mathcal{H}(\beta) = \log Z(\beta) - \beta \sum_{i=1}^{K} q_i(\beta) v_i$$

Recognizing that $\sum q_i(\beta) v_i$ is the expected value of the logits under the distribution q, denoted as $\mathbb{E}_q[v]$, we simplify the entropy to:

$$\mathcal{H}(\beta) = \log Z(\beta) - \beta \mathbb{E}_q[v]$$

Next, we take the derivative with respect to β using the product rule:

$$\frac{\partial \mathcal{H}}{\partial \beta} = \frac{\partial}{\partial \beta} \log Z(\beta) - \left(\mathbb{E}_q[v] + \beta \frac{\partial \mathbb{E}_q[v]}{\partial \beta} \right)$$

Using the chain rule on the partition function, $\frac{\partial}{\partial \beta} \log Z(\beta) = \frac{1}{Z} \sum_{i=1}^{K} v_i \exp(\beta v_i) = \mathbb{E}_q[v]$. Substituting this back, the $\mathbb{E}_q[v]$ terms cancel out exactly:

$$\frac{\partial \mathcal{H}}{\partial \beta} = \mathbb{E}_q[v] - \mathbb{E}_q[v] - \beta \frac{\partial \mathbb{E}_q[v]}{\partial \beta} = -\beta \frac{\partial \mathbb{E}_q[v]}{\partial \beta}$$

To evaluate the derivative of the expected value, $\frac{\partial \mathbb{E}_q[v]}{\partial \beta} = \sum_{i=1}^{K} v_i \frac{\partial q_i}{\partial \beta}$, we apply the quotient rule to the softmax function $q_i(\beta)$:

$$\frac{\partial q_i}{\partial \beta} = \frac{v_i \exp(\beta v_i) Z(\beta) - \exp(\beta v_i) \left(\sum_{j=1}^{K} v_j \exp(\beta v_j) \right)}{Z(\beta)^2}$$

$$\frac{\partial q_i}{\partial \beta} = v_i q_i - q_i \left(\sum_{j=1}^{K} v_j q_j \right) = q_i (v_i - \mathbb{E}_q[v])$$

Substituting this quotient rule result into the expected value derivative:

$$\frac{\partial \mathbb{E}_q[v]}{\partial \beta} = \sum_{i=1}^{K} v_i q_i (v_i - \mathbb{E}_q[v]) = \sum_{i=1}^{K} q_i v_i^2 - \mathbb{E}_q[v] \sum_{i=1}^{K} q_i v_i$$

$$\frac{\partial \mathbb{E}_q[v]}{\partial \beta} = \mathbb{E}_q[v^2] - (\mathbb{E}_q[v])^2 = \mathrm{Var}_q(v)$$

Substituting the variance back into the entropy derivative yields $\frac{\partial \mathcal{H}}{\partial \beta} = -\beta \mathrm{Var}_q(v)$. Finally, applying the chain rule to convert back to temperature T, where $\frac{\partial \beta}{\partial T} = -\frac{1}{T^2}$:

$$\frac{\partial \mathcal{H}}{\partial T} = \left(-\frac{1}{T} \mathrm{Var}_q(v) \right) \left(-\frac{1}{T^2} \right) = \frac{\mathrm{Var}_q(v)}{T^3}$$

Because the variance of a real-valued set is non-negative ($\mathrm{Var}_q(v) \geq 0$) and the temperature is strictly positive ($T > 0$), we conclude that $\frac{\partial \mathcal{H}}{\partial T} \geq 0$, proving the function is monotonically non-decreasing in the general case.

Because $q_i(T) > 0$ for every finite $T > 0$, $\mathrm{Var}_q(v) = 0$ if and only if all logits v_i are equal. Therefore $\frac{\partial \mathcal{H}}{\partial T} > 0$ exactly when the logits are not all equal; in the degenerate case $\mathrm{Var}_q(v) = 0$ and $\mathcal{H}(q_T) = \log K$ for every T. Under the non-degenerate condition, the entropy is strictly increasing in temperature, so Annealing-KD enforces a strictly monotonically increasing entropy curriculum. □

Remark 3.3 (Generalization to arbitrary scaling functions)**.** *The proof above uses the canonical parameterization $\Phi = 1/T$, but the result extends to any positive, monotonically decreasing scaling function $\Phi(\mathcal{T})$ applied to the logits. Replacing $1/T$ with a general Φ and applying the chain rule yields:*

$$\frac{\partial \mathcal{H}}{\partial \mathcal{T}} = -\Phi(\mathcal{T}) \cdot Var_q(v) \cdot \Phi'(\mathcal{T})$$

Since $\Phi(\mathcal{T}) > 0$, $Var_q(v) \geq 0$, and $\Phi'(\mathcal{T}) < 0$ (monotonically decreasing), the product is non-negative, ensuring $\frac{\partial \mathcal{H}}{\partial \mathcal{T}} \geq 0$. In particular, the specific annealing function defined in Eq. 3.3, $\Phi(\mathcal{T}) = 1 - \frac{\mathcal{T}-1}{\tau_{\max}}$, satisfies these conditions on the interval $[1, \tau_{\max}]$.

Bridging Target Entropy and Optimization Complexity

While Theorems 3.1 and 3.2 establish that dynamic temperature scaling strictly controls the Shannon Entropy of the target distribution, it is critical to distinguish between the *entropy of the output distribution* and the *geometric complexity* (e.g., VC dimension) of the underlying function. The theorems are exact statements about the teacher's output distribution (known properties of the temperature-scaled softmax, proved here for completeness); the connection between these properties and the student's optimization behavior is discussed below as a theoretically motivated interpretation, supported by known relationships between target smoothness and optimization dynamics.

A theoretical maximum-entropy target (a uniform distribution as $T \to \infty$) is trivially easy for a student network to model, but it is pedagogically useless as it contains zero mutual information regarding the class structures. Therefore, Annealing-KD does not seek to maximize entropy absolutely, but rather uses a bounded initial temperature $\tau_{\max}$ to construct a target that balances class information with optimization stability.

The pedagogical usefulness of bounding target entropy lies in its expected relationship to gradient magnitudes and landscape smoothness. In standard KD ($T = 1$), a highly confident teacher produces low-entropy, nearly one-hot target vectors. Under the KL-divergence variant of Annealing-KD where only the teacher's logits are temperature-scaled, the gradient of the KL divergence loss with respect to the student's logits z_j reduces to $(p_j - q_j(T))$, the raw residual between the student's output and the teacher's temperature-scaled distribution. When q_T is sharp (low T) and the student is still poorly trained, this residual is large, producing high-magnitude gradient updates. Such volatile updates can push the student toward sharp regions of the loss landscape [73], consistent with the Spectral Bias phenomenon [102] whereby neural networks struggle to fit high-frequency target functions.

Raising the target entropy early in training via $\tau_{\max}$ produces softer, more diffuse class-probability vectors q_T. When the student is also poorly trained and its own distribution p is diffuse, the residual $(p - q_T)$ is typically smaller in magnitude than it would be against a sharp q_T, so the KL-gradient updates are typically smaller. This is an expectation about the typical early-training regime, not a guarantee for an arbitrary p: a softer q_T does not reduce $(p - q_T)$ for every p. To the extent that gradient magnitudes are reduced in this regime, the effective supervisory signal is smoother early in training; as the temperature is annealed and target entropy decreases, this smoothing is gradually relaxed, allowing the student to fit sharper, higher-complexity targets after an initial stabilization phase. We do not claim a formal gradient bound or a proof that the empirical risk landscape becomes flatter; these connections are the interpretation delineated in Section 3.3.2.

Continuous Annealing as an Information Curriculum

Building on the entropy monotonicity established in Theorem 3.2 and the interpretive bridge to optimization discussed above, we can now articulate the principled rationale underlying AKD. Instead of training discrete Teacher Assistants to physically lower the complexity of the target space, we define a continuous, time-dependent temperature schedule $T(t)$ over normalized training time $\tau \in [0, 1]$. By selecting a decay function such that $\frac{\partial T}{\partial \tau} < 0$, we guarantee via the chain rule ($\frac{\partial \mathcal{H}}{\partial \tau} = \frac{\partial \mathcal{H}}{\partial T} \cdot \frac{\partial T}{\partial \tau}$) that the rate of change of target entropy over time is strictly negative: $\frac{\partial \mathcal{H}}{\partial \tau} < 0$. This defines AKD as a principled curriculum with three interpretable phases:

1. **Phase 1 (High T, High Entropy):** The student is exposed to a smoothed target space. The capacity gap is mitigated as the target complexity is artificially lowered to match the student's initial capacity, reducing the effect of Spectral Bias.

2. **Phase 2 (Annealing):** As $T(t)$ decreases, $\mathcal{H}(q_T)$ strictly decreases, and the teacher's outputs become progressively sharper and more class-specific, gradually revealing finer-grained relational structure among classes, the privileged "dark knowledge" of the teacher.

3. **Phase 3 (Low T, Low Entropy):** Because the student was stabilized on smooth targets during Phase 1, it is better positioned to absorb the sharper, more complex targets at $T = 1$, avoiding the suboptimal local minima that can arise in standard fixed-temperature KD.

Scope of the Theoretical Results

It is important to clearly delineate the scope of the theoretical contributions in this chapter. Our formal results operate at two distinct levels:

What is mathematically established (mechanism). Theorems 3.1 and 3.2 are exact mathematical results about the teacher's output distribution under temperature scaling. They establish that: (i) Shannon Entropy is bounded between zero and $\log K$, and (ii) entropy is a monotonically non-decreasing function of temperature, and strictly increasing under the standard non-degenerate condition that the logits are not all identical, with a closed-form derivative $\frac{\partial \mathcal{H}}{\partial T} = \frac{\mathrm{Var}_q(v)}{T^3}$. These proofs are self-contained and do not depend on assumptions about the student model, loss function, or training dynamics. Both properties

are known in the literature (see the references accompanying Theorems 3.1 and 3.2); we do not claim them as new results, and include the proofs for completeness and to fix notation.

What is interpreted (implication). The connection from entropy control to improved optimization behavior, such as smoother loss landscapes, reduced gradient magnitudes, and easier convergence for the student, is not derived as a formal theorem. Instead, it is presented as a theoretically grounded interpretation. This interpretation draws on well-established relationships between target smoothness and optimization dynamics (e.g., Spectral Bias, Lipschitz continuity of loss gradients), but the full causal chain from higher target entropy to improved generalization involves intermediate steps that are argued, not proven.

Summary. Taken together, our theoretical analysis gives exact results about the mechanism by which Annealing-KD controls the information content of the supervisory signal, and a principled interpretation of how this mechanism encourages favorable optimization dynamics. The latter claim is consistent with the LUPI framework's demand for bounded target complexity (Section 2.2) and is extensively supported by the empirical results presented in Section 3.4.

3.4 Experiments

In this section, we describe the experimental evaluation of our proposed Annealing KD method. We evaluate our technique on both image classification and natural language inference tasks. In all of our experiments, we compare the annealing KD results with TAKD, standard KD, and training student without KD results.

3.4.1 Datasets

For image classification, we assess Annealing-KD on CIFAR-10 and CIFAR-100 datasets [75] which are image datasets containing 32×32 color images with 10 and 100 classes respectively. For the natural language inference task, we employ the General Language Understanding Evaluation (GLUE) benchmark [131], which is a collection of nine different tasks for training, evaluating, and analyzing natural language understanding models. GLUE consists of Multi-Genre Natural Language Inference (MNLI) [138], Quora Question Pairs

(QQP) [100], Question Natural Language Inference (QNLI) [104], Stanford Sentiment Treebank (SST-2) [113], Corpus of Linguistic Acceptability (COLA) [135], Semantic Textual Similarity Benchmark (STS-B) [15], Microsoft Research Paraphrase Corpus (MRPC) [38], Recognizing Textual Entailment (RTE) [9], and Winograd NLI (WNLI) [78].

3.4.2 Experimental Setup for Image Classification Tasks

For image classification experiments, we used CIFAR-10 and CIFAR-100 datasets with the same experimental setup in the TAKD method [91]. In these experiments, we used ResNet and plain CNN networks as the teacher, student, and also the teacher assistant for the TAKD baseline. For the ResNet experiments, we used ResNet-110 as the teacher and ResNet-8 as the student. For plain CNN experiments, we used CNN network with 10 layers as teacher and 2 layers as the student according to TAKD. Also, for the TAKD baseline, we used ResNet-20 and CNN with 4 layers as the teacher assistant. Tables 3.1 and 3.2 compare the annealing KD performance with other baselines over CIFAR-10 and CIFAR-100 datasets respectively. For the ResNet experiments in both tables 3.1 and 3.2, the teacher ResNet-110 is trained from scratch and a ResNet-20 TA is trained by the teacher using KD. Then we would like to train a ResNet-8 student using different techniques and compare their performance against our Annealing KD method. In this regard, we evaluate the performance of training the student from scratch, training with the large ResNet-110 teacher using KD, training with TA as the teacher and using our Annealing-KD approach. The results of this experiment with ResNet show that our Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD. More details about the training hyper-parameters are added to the appendix A.1.

3.4.3 Experimental setup for GLUE tasks

For these set of experiments, we use the GLUE benchmark which consists of 9 natural language understanding tasks. In the first experiment (Table 3.3), we use RoBERTa-large (24 layers) as teacher, DistilRoBERTa (6 layers) as student, and RoBERTa-base (12 layers) as the teacher assistant for the TAKD baseline. For Annealing KD, we use a maximum temperature of 7, learning rate of 2e-5, and train for 14 epochs in Stage I, and 6 epochs in Stage II. In table 3.3 the Annealing KD and the other baselines performances on dev set of GLUE tasks are compared. Also, we compared the performances of these methods on test set based on the GLUE benchmark's leaderboard results in table 3.4.

Table 3.1: Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	Teacher(110)	from scratch	93.8
	TA(20)	KD	92.39
	Student(8)	from scratch	88.44
	Student(8)	KD	88.45
	Student(8)	TAKD	88.47
	Student(8)	**Annealing KD (ours)**	**89.44**
CNN	Teacher(10)	from scratch	90.1
	TA(4)	KD	82.39
	Student(2)	from scratch	72.75
	Student(2)	KD	72.43
	Student(2)	TAKD	72.62
	Student(2)	**Annealing KD (ours)**	**73.17**

In the second experiment (Table 3.5), we use BERT-large (24 layers) as teacher, BERT-small (4 layers) as student, and BERT-base (12 layers) as the teacher assistant of TAKD. We use a maximum temperature of 7 for MRPC, SST-2, QNLI, and WNLI, and 14 for all other tasks. The number of epochs in Stage I is twice the maximum temperature, and 6 in Stage II. We use the learning rate of 2e-5 for all tasks except RTE and MRPC which use 4e-5. Table 3.5 compares the performance of annealing KD and other baselines on dev set for small-BERT experiments. For more details regarding other hyper-parameters, refer to the appendix A. We also perform ablation on the choice of loss function in Stage I, and choice of different max temperature values, both of which can be found in the appendix A.

3.4.4 GLUE Results

We present our results in Tables 3.3, 3.4, and 3.5. We see that Annealing KD outperforms the other techniques both on dev set as well as the GLUE leaderboard. Furthermore, in table 3.5, when we reduce the size of the student to a 4 layer model (BERT-Small), we notice almost twice as big of a gap in the average score over Vanilla KD when compared with DistilRoBERTa (Table 3.3). We can also observe TAKD improving slightly over Vanilla KD, with the improvement being more significant in the case of the smaller student (BERT-Small). The results in Tables 3.3 and 3.5 are single-run results. We also conducted a multi-seed study (10 seeds) of the DistilRoBERTa and BERT-Small experiments; the

Table 3.2: Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	teacher(110)	from scratch	71.92
	TA(20)	KD	67.6
	student(8)	from scratch	61.37
	student(8)	KD	61.41
	student(8)	TAKD	61.82
	student(8)	**Annealing KD (ours)**	**63.1**
CNN	Teacher(10)	from scratch	64.89
	TA(4)	KD	60.73
	student(2)	from scratch	51.35
	student(2)	KD	51.62
	student(2)	TAKD	51.85
	student(2)	**Annealing KD (ours)**	**53.35**

Table 3.3: DistilRoBERTa results for Annealing KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy; MNLI mismatched is reported but not aggregated. Aggregates are computed from the displayed cells and rounded to two decimals.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	68.1	86.3	91.9	92.3	96.4	94.6	91.5	90.22/89.87	56.33	85.29
From scratch	59.3	67.9	88.6	88.5	92.5	90.8	90.9	84/84	52.1	79.40
Vanilla KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18/84.11	56.33	80.80
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	**91.7**	83.89/84.18	56.33	80.85
Annealing KD	**61.67**	**73.64**	**90.6**	**89.01**	**93.11**	**91.64**	91.5	**85.34/84.6**	56.33	**81.43**

corresponding results and findings are reported in Appendix A.3, where the ranking of the methods is the same as here.

Table 3.4: Performance of DistilRoBERTa trained by annealing KD on the GLUE leaderboard (test set) compared with Vanilla KD and TAKD. We applied the standard tricks to all 3 methods and fine-tune RTE, MRPC and STS-B from the trained MNLI student model. Matthews correlation for CoLA, Pearson/Spearman for STS-B, F1/accuracy for MRPC and QQP, matched/mismatched accuracy for MNLI, accuracy elsewhere. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed tasks; paired metrics (MRPC, STS-B, QQP) are averaged within a task and MNLI matched/mismatched are averaged before aggregation. Aggregates are computed from the displayed cells and rounded to one decimal.

KD Method	CoLA	MRPC	STS-B	SST-2	MNLI-m	MNLI-mm	QNLI	QQP	RTE	WNLI	Score
Vanilla KD	**54.3**	86/80.8	85.7/84.9	93.1	83.6	82.9	90.8	71.9/89.5	74.1	65.1	78.9
TAKD	53.2	86.7/82.7	85.6/84.4	93.2	**83.8**	83.2	**91**	72/89.4	**74.2**	65.1	79.0
Annealing KD	54	**88.0/83.9**	**87.0/86.6**	**93.6**	**83.8**	**83.9**	90.8	**72.6/89.7**	73.7	65.1	**79.4**

Table 3.5: BERT-Small results for Annealing KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy; MNLI mismatched is reported but not aggregated. Aggregates are computed from the displayed cells and rounded to two decimals.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	65.8	71.48	89.38	89.2	92.77	92.82	91.45	86.3/86.4	60.56	82.20
Vanilla KD	33.5	57	86	72.3	88.76	**83.15**	**87**	72.62/73.19	54.92	70.58
TAKD	34.24	59.56	85.23	71.1	89.1	82.62	**87**	72.32/72.45	54.92	70.68
Annealing KD	**35.98**	**61**	**86.2**	**74.54**	**89.44**	83.14	86.5	**73.85/74.84**	54.92	**71.73**

3.5 Scaling to the Generative Era: Upcycling Hybrid LLMs

The empirical evaluations presented in Section 3.4 validate Annealing-KD on foundational classification architectures (ResNet, CNN) and discriminative language models (BERT, RoBERTa). However, the deep learning paradigm has shifted decisively toward massive, autoregressive Large Language Models (LLMs). To demonstrate the generality of our method beyond classification settings and its specific efficacy in solving the *architectural capacity gap*, we extend our evaluation to the modern task of hybrid LLM upcycling.

3.5.1 Task Definition: Hybrid Model Upcycling

As the context window requirements for LLMs scale to hundreds of thousands of tokens, the quadratic computational complexity of standard softmax attention ($\mathcal{O}(n^2)$ in sequence length) becomes a critical inference bottleneck. A state-of-the-art solution proposed by Chen et al. [21] is to *upcycle* a fully trained Transformer model into a hybrid architecture through HALO (Hybrid Attention via Layer Optimization), a pipeline that replaces a majority of the softmax attention layers with efficient linear attention or Recurrent Neural Network (RNN) mechanisms while preserving the model's learned representations.

The HALO procedure consists of three stages: (1) *Hidden State Alignment*, where each attention layer is independently aligned with a corresponding RNN layer via mean squared error on hidden states; (2) *Knowledge Distillation*, where the full hybrid model is trained end-to-end to match the original Transformer's output distribution via KL-divergence; and (3) *Fine-tuning* with extended context length. This upcycling process introduces a severe *architectural* capacity gap: the linear attention layers in the hybrid student structurally lack the token-to-token information routing capability of softmax attention, making it fundamentally harder for the student to reproduce the sharp attention patterns of the teacher [21]. When standard KL-divergence distillation is applied in Stage 2, this high-frequency target mismatch can trap the student in suboptimal local minima, particularly for knowledge-intensive benchmarks.

3.5.2 Experimental Setup

We adopt the HALO pipeline [21] and isolate Stage 2 (knowledge distillation) as the experimental variable. We compare two distillation objectives: (i) standard (Vanilla) KL-Divergence KD, referred to as *HypeNet–KL*, and (ii) our Annealing-KD using the KL-divergence variant ($\mathcal{L}_{\text{KL}}^{\text{Annealing}}$ in Eq. 3.2), referred to as *HypeNet Annealing*, where the temperature is dynamically annealed over the distillation phase. The KL-divergence formulation is the natural choice for autoregressive language models, where the training objective is distribution matching over a large vocabulary rather than logit regression.

Model Configuration.

- **Student Model (HypeNet):** A 1.7B parameter hybrid model constructed by replacing 75% of the standard attention layers in Qwen3-1.7B [143] with linear attention (Lightning Attention) layers, following the HypeNet architectural guidelines [21].
- **Small Teacher:** The original, fully-trained Qwen3-1.7B Transformer.

- **Large Teacher:** The fully-trained Qwen3-8B Transformer.

Evaluation Framework. We evaluate all models using the Language Model Evaluation Harness (lm-eval) [46], a widely-adopted open-source framework developed by EleutherAI for standardized, reproducible evaluation of language models. The lm-eval framework provides a unified interface for zero-shot and few-shot evaluation across a broad suite of natural language understanding and reasoning benchmarks. It serves as the backend powering Hugging Face's Open LLM Leaderboard and is used extensively in the LLM research community to ensure comparable evaluation across different models and training configurations. By employing this standardized pipeline, our results can be directly contextualized within the broader LLM literature.

We report 0-shot normalized accuracy across seven diverse commonsense reasoning and knowledge benchmarks:

- **ARC-c** and **ARC-e** (AI2 Reasoning Challenge) [29]: Grade-school science questions, with the Challenge subset (ARC-c) containing only questions that retrieval and co-occurrence baselines answer incorrectly, requiring deeper multi-step reasoning.
- **HellaSwag** [150]: Evaluates grounded commonsense reasoning via sentence completion, using adversarial filtering to create examples that are trivial for humans but challenging for models.
- **MMLU** (Massive Multitask Language Understanding) [58]: A comprehensive knowledge benchmark containing multiple-choice questions across 57 subjects spanning STEM, humanities, and social sciences.
- **OpenbookQA** [90]: Tests the ability to combine provided elementary science facts with broader commonsense knowledge for multi-hop reasoning.
- **PIQA** (Physical Interaction QA) [10]: Tests physical commonsense reasoning about everyday objects and interactions.
- **Winogrande** [109]: A large-scale adversarial Winograd Schema challenge for pronoun resolution requiring commonsense reasoning.

3.5.3 Results and Discussion

The 0-shot accuracy results are presented in Table 3.6. The first row provides the original Qwen3-1.7B Transformer with full softmax attention as a reference baseline, achieving an average accuracy of 57.06% across the seven benchmarks.

Table 3.6: 0-Shot normalized accuracy of hybrid models on the LM Evaluation Harness [46] benchmark. HypeNet–KL uses standard KL-Divergence distillation in HALO Stage 2, while HypeNet Annealing replaces it with Annealing-KD. The first row shows the original Qwen3-1.7B Transformer as a reference.

Model	Teacher	ARC-c	ARC-e	HellaSwag	MMLU	OpenbookQA	PIQA	Winogrande	Average
Qwen3-1.7B (Teacher)	–	43.17	69.70	60.40	55.71	37.20	72.31	60.93	57.06
HypeNet–KL	Qwen3-1.7B	42.66	69.74	56.91	48.95	36.80	72.09	61.56	55.53
HypeNet Annealing	Qwen3-1.7B	42.41	71.76	58.18	50.74	37.60	73.29	60.54	**56.36**
HypeNet–KL	Qwen3-8B	44.62	71.63	60.97	45.14	38.60	73.67	64.88	57.07
HypeNet Annealing	Qwen3-8B	45.73	73.23	60.89	49.03	38.80	73.83	65.70	**58.17**

Table 3.6 reports the results. Converting Qwen3-1.7B into a hybrid architecture with standard KL-divergence distillation (HypeNet–KL, 1.7B teacher) lowers the seven-task average from 57.06% to 55.53% (−1.53pp), with the largest single drop on MMLU (55.71% → 48.95%). Replacing KL with Annealing-KD in Stage 2 recovers a larger portion of that lost performance than the widely used KL-divergence distillation does, reaching 56.36% (+0.83pp over HypeNet–KL) with the same teacher. With the larger Qwen3-8B teacher the ordering is the same: HypeNet–KL reaches 57.07% and HypeNet Annealing 58.17% (+1.10pp), and on MMLU Annealing-KD raises the score from 45.14% to 49.03%.

These results indicate that Annealing-KD is useful beyond the classification settings of Section 3.4. In training autoregressive generative models, such as the hybrid-model upcycling task discussed here, substituting Annealing-KD for the standard KL objective gives a consistently higher average score under both teachers. This is a setting quite different from the ones the method was originally designed for: the supervision is over next-token distributions across a long sequence rather than over a single label, and the student differs from its teacher in the form of its attention layers rather than only in depth or width. That the same annealed-target schedule still improves on the standard objective in this setting suggests the mechanism is not specific to classification. The experiment shows that the method can help in this kind of generative setting too.

3.6 Discussion

A consistent pattern emerges across all three experimental domains examined in this chapter. In image classification, the improvement gap between Annealing-KD and the baselines on CIFAR-100 is larger than on CIFAR-10. A similar trend is observed in the NLP experiments, where the performance gain of Annealing-KD over Vanilla KD is more pronounced

for the smaller BERT-Small student than for DistilRoBERTa. In both cases, the distillation problem is more difficult: CIFAR-100 requires the teacher to learn a more complex function, while BERT-Small has a smaller capacity than DistilRoBERTa, widening the capacity gap. From this observation, we can conclude that whenever the gap between the teacher and student is larger, Annealing-KD provides a greater relative improvement over the baselines.

The hybrid LLM upcycling experiments in Section 3.5 reinforce this conclusion in a qualitatively different setting. Here, the capacity gap is not merely parametric but *architectural*: the hybrid student's linear attention layers are structurally less expressive than the teacher's softmax attention. This architectural mismatch produces a 1.53 percentage point drop under standard KL-divergence distillation with the same-size teacher (57.06% $\rightarrow$ 55.53%). Annealing-KD recovers part of this loss (56.36%), and with the larger 8B teacher it reaches the highest average in the table (58.17%). On MMLU it raises accuracy from 45.14% (KL with the 8B teacher) to 49.03%, though this remains below the 1.7B teacher's 55.71%. The consistent effect across both teachers is on the seven-task average; per-task results are mixed, and this setup does not establish general parity with the teacher.

These results collectively demonstrate that Annealing-KD is effective across parametric capacity gaps (ResNet, BERT), task complexity gaps (CIFAR-10 vs. CIFAR-100), and architectural capacity gaps (Transformer to hybrid), validating its generality as a distillation technique.

3.7 Conclusion and Future Work

In this chapter, we addressed the capacity-gap problem in knowledge distillation, where the performance of a student model degrades when the teacher's capacity far exceeds its own. We proposed Annealing Knowledge Distillation (Annealing-KD), which tackles this problem by dynamically controlling the complexity of the teacher's supervisory signal through a monotonically decreasing temperature schedule. Rather than introducing auxiliary Teacher Assistant networks, Annealing-KD feeds the rich information provided by the teacher's soft targets incrementally and more efficiently.

We gave a self-contained information-theoretic analysis of the method, showing, via known properties of the temperature-scaled softmax, that dynamic temperature scaling provides monotonic control over the Shannon Entropy of the target distribution (Theorems 3.1 and 3.2). Building on these formal results, we presented a theoretically grounded

interpretation of how this entropy control encourages favorable optimization dynamics, by limiting gradient magnitudes and promoting a smoother loss landscape, thereby acting as a principled curriculum (see Section 3.3.2 for a precise delineation of what is formally proven versus what is interpreted).

Empirically, we validated Annealing-KD across three distinct experimental settings: (1) image classification on CIFAR-10 and CIFAR-100 with ResNet and CNN architectures, (2) natural language understanding on the GLUE benchmark with BERT and RoBERTa models, and (3) hybrid LLM upcycling, where we applied Annealing-KD within the HALO pipeline [21] to distill Qwen3 Transformer models into hybrid architectures with linear attention. In all settings, Annealing-KD consistently outperformed standard KD and TAKD baselines. In the hybrid upcycling setting, Annealing-KD with a Qwen3-8B teacher produced a hybrid student that surpassed even the original full-attention Qwen3-1.7B model (58.17% vs. 57.06% average accuracy), demonstrating the method's effectiveness against both parametric and architectural capacity gaps.

Importantly, the annealing mechanism is agnostic to the specific loss function: it can be instantiated with an MSE objective in logit space ($\mathcal{L}_{\text{MSE}}^{\text{Annealing}}$), as used in our classification experiments, or with a KL-divergence objective in probability space ($\mathcal{L}_{\text{KL}}^{\text{Annealing}}$), as used in the hybrid LLM experiments (Eq. 3.2). This flexibility, combined with the architecture-agnostic nature of the method, establishes Annealing-KD as a simple technique that can be applied on top of different KD variants, from traditional classification distillation to modern cross-architecture LLM conversion pipelines.

Chapter 4

Progressive Knowledge Distillation

4.1 Overview

In Chapter 3 we addressed the capacity-gap problem by introducing Annealing-KD, a technique that controls the complexity of the teacher's supervisory signal through a dynamic temperature schedule. While Annealing-KD successfully mitigates the capacity gap without auxiliary networks, it operates on a *fully trained* teacher: the practitioner must first train the teacher to convergence, then apply the annealing schedule during distillation. This raises a practical question that Annealing-KD leaves open: *which teacher checkpoint should be used for distillation?*

As demonstrated by Cho and Hariharan [25], a fully-trained teacher is not necessarily the optimal teacher for knowledge distillation, an early-stopped teacher can sometimes yield a better student. This observation implies that one must search through the teacher's training checkpoints to identify the best model for distillation, a process that can be prohibitively expensive for large pre-trained language models. We refer to this as the *checkpoint-search* problem.

In this chapter, we propose *Progressive Knowledge Distillation* (Pro-KD), a technique that simultaneously addresses both the capacity-gap problem (introduced in Section 2.2) and the checkpoint-search problem. Rather than distilling from a single, fixed teacher checkpoint, Pro-KD has the student follow the teacher's training trajectory from the beginning, learning from progressively more capable snapshots of the teacher as it trains. Combined with the dynamic temperature schedule introduced in the previous chapter, this creates a doubly smooth curriculum: the teacher itself grows in complexity over time, while the temperature annealing further softens each snapshot's outputs.

4.2 Introduction

As established in Section 2.2, the capacity-gap problem in KD refers to the phenomenon where increasing the teacher's capacity beyond a certain point degrades, rather than improves, the student's performance [85, 91]. The prevailing solution, TAKD [91], introduces intermediate teacher-assistant networks to bridge this gap, but incurs substantial computational overhead and error accumulation (see Section 2.2 for a detailed discussion).

Beyond the capacity gap, a second limitation of standard KD has received comparatively less attention: a fully-trained teacher might not be the best teacher for the student [25]. In other words, an early-stopped teacher can be a better option for KD compared to a fully-trained one. This observation implies that we need to search through the checkpoints of the teacher to find the best model for the distillation process. However, this search can be very expensive especially when we deal with pre-trained language models. We refer to this problem as the *checkpoint-search* problem. Our investigations of this checkpoint search problem show the significance of this issue especially when dealing with PLMs. Table 4.1 depicts that the best distillation checkpoint of the teacher varies for each task and for each student configuration and it is very different from the teacher's best performing checkpoint.

In this chapter, we propose our Pro-KD solution to tackle both the capacity-gap and checkpoint-search problems. In Pro-KD, the student grows gradually with the growth of the teacher. We hypothesize that the training path of the teacher can be informative for the student and we should not disregard it. Therefore, in contrast to the original KD where the student learns from the best pre-trained teacher, in Pro-KD the student starts its learning process together with the teacher. Furthermore, in contrast to the TAKD technique which reduces the capacity of the teacher by adding intermediate TAs to the distillation process, in our Pro-KD, we mitigate the capacity-gap by making the training path of the student more smooth and gradual by following the training footsteps of the teacher. Moreover, to make the training smooth further for the student, inspired by Annealing-KD, we apply an adaptive temperature factor to the output of the teacher while being trained. This temperature factor is decreased during the training. We motivate this design through the capacity-gap bound of Section 2.2 and demonstrate its effectiveness empirically, with experiments on both NLP (the GLUE [131] benchmark and SQuAD 1.1 and 2.0) and image classification tasks (CIFAR-10, CIFAR-100). The contributions of this chapter are summarized in the following:

1. We propose our Pro-KD solution to the KD *capacity-gap* and *checkpoint-search* problems. In Pro-KD, the student follows the training footsteps of the teacher. Moreover,

Table 4.1: The best performing checkpoints vary for different size of the models, and different tasks. It is evident that the best checkpoint of the teacher does not necessarily lead to the best performing student model.

Task	Model	Best #ChPt Epoch	Performance	$\Delta\uparrow$
MRPC	$\text{BERT}_{\text{LARGE}}$(♠T)	**6**	88.22	-
	$\text{BERT}_{\text{SMALL}}$(♣S)	**3**	85.29	+**0.7**
	DistilBERT(♣S)	**5**	89.16	+**0.8**
SST-2	$\text{BERT}_{\text{LARGE}}$(♠T)	**1**	92.89	-
	$\text{BERT}_{\text{SMALL}}$(♣S)	**7**	88.76	+**0.6**
	DistilBERT(♣S)	**7**	91.63	+**0.4**
QNLI	$\text{BERT}_{\text{LARGE}}$(♠T)	**6**	92.4	-
	$\text{BERT}_{\text{SMALL}}$(♣S)	**2**	87.08	+**0.3**
	DistilBERT(♣S)	**4**	90.46	+**0.2**

♠T:Teacher, ♣S: Student, #ChPt: Checkpoint Number, Performance: the standard GLUE metric of each task (F1 for MRPC, accuracy for SST-2 and QNLI), $\Delta\uparrow$: Performance improvement compared to the best checkpoint of the teacher

it introduces a dynamic temperature function to the output of the teacher when it is distilled to the student to make the student training gradual and smooth.

2. We apply our technique to a ResNet-8 model on both CIFAR-10 and CIFAR-100 image classification tasks, and to natural language understanding with BERT-based models such as DistilRoBERTa and BERT-Small on the GLUE benchmark and on SQuAD 1.1 and 2.0, where it outperforms the evaluated baselines.

3. Our technique is simple, architecture agnostic, does not require training any extra network and can be applied on top of different variants of KD.

We adopt the standard KD formulation introduced in Section 2.1 (Eq. 2.3), where the student learns from both the ground-truth *hard labels* and the temperature-softened *soft labels* produced by the teacher. The reader is referred to Chapter 2 for a detailed treatment of the KD loss and its temperature-scaling mechanism.

4.3 Related Work

In this section, we review the most related works to this chapter in the literature.

4.3.1 Capacity-Gap Problem in KD

As discussed in detail in Section 2.2, the capacity-gap problem arises when the teacher model is too powerful relative to the student, causing the distillation upper bound to exceed the direct-learning bound and the distillation to become less effective [91, 85]. The prevailing solution, TAKD [91], introduces intermediate teacher-assistant (TA) networks to bridge this gap. However, TAKD has two notable downsides: training a separate TA network is costly, and the sequential training of multiple networks can lead to error accumulation and error propagation to the student. Moreover, TAKD was only demonstrated on image classification tasks.

As an alternative, Annealing-KD (presented in Chapter 3) proposed a TA-free solution by applying an annealed dynamic temperature factor to the teacher's output, making the training process gradual for the student. The temperature starts from a large value to maximally smooth the teacher's output at the beginning of training, and gradually tends toward 1 where no smoothing is applied. Our Pro-KD is inspired by Annealing-KD in using a dynamic temperature factor to address the capacity-gap problem, but introduces two key improvements: first, Annealing-KD still suffers from the checkpoint-search problem while Pro-KD does not; second, Pro-KD takes advantage of the teacher's intrinsic training trajectory, which is not considered in Annealing-KD.

4.3.2 Checkpoint Search Problem in KD

[25] highlighted their findings over computer vision (CV) models that bigger models are not necessarily better teachers and also early stopped teachers can train students better. To overcome these issues they propose an ad-hoc approach called Early Stopped KD (ESKD); however, ESKD still needs to search among the pool of teacher's early checkpoints and the selection mechanism among this early checkpoints is not clear. Moreover, in PLMs, generally teachers are not fine-tuned for long and yet searching among the checkpoints of large PLMs is expensive. [70] proposed Route Constrained Optimization (RCO) distillation which is a curriculum learning technique to follow easy-to-hard training scheme on top of the teacher's training trajectory. We can deem this technique as the most related work to us with the following differences: 1- our Pro-KD has a temperature factor to make the training process more gradual which can handle the capacity-gap better; 2- RCO still requires search to define its trajectory; 3- we evaluated Pro-KD on both CV and NLP tasks but RCO is only evaluated on CV tasks. We will show in the Experiment section that our Pro-KD outperforms RCO consistently.

In the next section, we introduce our Pro-KD solution to the capacity-gap and checkpoint-search problems. Pro-KD exploits a smooth training process for the student to gradually learn from the teacher while being trained from scratch, instead of learning from a fully-trained Teacher.

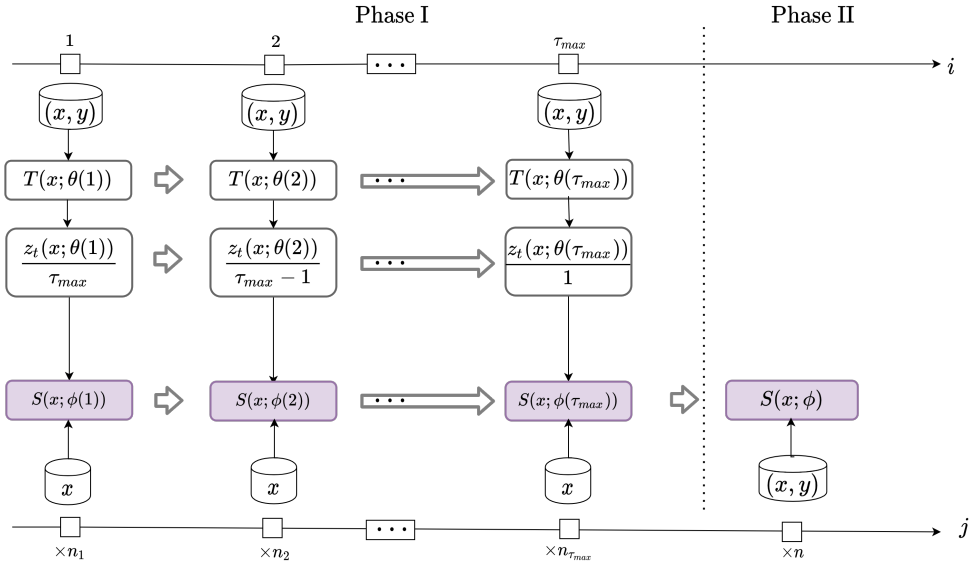


Figure 4.1: Stage I and Stage II of the Pro-KD method. Stage I: the teacher is trained on the labeled training data. The student at each step tries to mimic the behavior of the corresponding teacher checkpoint. The logits of the teacher at each time step is attenuated with the temperature parameter. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$. Stage II: training the student with the labeled data only using the cross entropy loss. n_i for $1 \leq i \leq E_T$ (with E_T the number of teacher epochs) refers to the number of training epochs of the student corresponding to the i^{th} training epoch of the teacher. n_is should add up to N, the preset total number of Stage-I student epochs. Bear in mind that since the teacher is usually early stopped, teacher is trained less longer than the student model.

4.4 Methodology: Pro-KD

The methodology of this chapter concerns addressing the capacity-gap and checkpoint-search problems in KD. We propose our Pro-KD technique which is a progressive training

procedure in which the student is trained together with the teacher. Therefore in Pro-KD, the student is not exposed to the output of a fully trained teacher from beginning, and instead, it learns from the teacher at the same time while the teacher is being trained. This training process for the student is more gradual and also the student can learn from the training path of the teacher. Inspired by Annealing-KD, we define the training in two stages: first, the student is only supervised by an adaptive smoothed version of the teacher; second, the student will be only trained on ground-truth labels. The detail of each stage is explained in the following.

Stage I) General Step-by-Step Training with a Teacher In this stage, the teacher is being trained using the cross entropy loss on the training data. Without loss of generality, let's assume that $\theta_T(i)$, the optimized parameters of the teacher at the beginning of epoch i, will be updated during the i^{th} epoch of training to obtain $\theta_T(i+1)$:

$$\begin{aligned} &\mathcal{L}_T(\theta_T(i)) = H_{CE}\big(y, T(x;\theta_T(i))\big) \\ &\theta_T(i+1) \leftarrow \theta_T(i) - \eta_T \, \nabla_{\theta_T} \mathcal{L}_T(\theta_T(i)) \end{aligned} \tag{4.1}$$

Then, a smoothed version of the teacher output using a temperature factor $\mathcal{T}_i$ at epoch i will be used to train the student at epoch j:

$$\begin{aligned} &\mathcal{L}_S^I(\theta_S(j)) = \|z_s(x;\theta_S(j)) - \frac{z_t(x;\theta_T(i))}{\mathcal{T}_i}\|_2^2 \\ &\theta_S(j+1) \leftarrow \theta_S(j) - \eta_S \, \nabla_{\theta_S} \mathcal{L}_S^I(\theta_S(j)) \end{aligned} \tag{4.2}$$

where $\mathcal{L}_S^I$ refers to the student loss function in Stage I, η_T, η_S are the teacher and student learning rates, z_s and z_t are the logits of the student and teacher respectively, $\theta_T(i)$ and $\theta_S(j)$ represent the parameters of the teacher at epoch i and of the student at epoch j respectively (the subscripts are used throughout this chapter because, unlike Chapters 5 and 7, both networks are being trained and both sets of parameters appear in the same equations), and $\mathcal{T}_i$ is the adaptive temperature factor. We write E_T for the number of teacher epochs (distinct from the maximum temperature τ_{max}) and u for the temperature-update interval; the temperature decreases stepwise from τ_{max} toward 1 as

$$\mathcal{T}_i = \max\Big\{1,\ \tau_{max} - \big\lfloor (i-1)/u \big\rfloor\Big\}, \qquad i = 1, \ldots, E_T. \tag{4.3}$$

The rationale behind applying the temperature to the teacher output is to make the output of the teacher smoother for the student to learn, especially at early stages of the training process.

It is evident that early-stopped teachers can be better teachers for KD [25]. In that case, for an early-stopped teacher, the number of training epochs of the student can go longer than that of the teacher. Therefore, for each given epoch i of the teacher, the student can be trained for $n_i \geq 1$ epochs, subject to $\sum_{i=1}^{E_T} n_i = N$ where N is the total number of Stage-I student training epochs. n_i is set to a constant integer number in most of our experiments but can be customized as well.

Bear in mind that we apply an adaptive temperature to the teacher logit in the student distillation loss, and this temperature starts from the highest τ_{max} value and decrease linearly with the epoch number of the teacher throughout training. τ_{max} is a hyper-parameter in our training. Moreover, we keep the temperature fixed at each epoch i. After training with the teacher logits, in the next stage the student model will be trained on the ground-truth labels.

Stage II) Training with the Ground-Truth Labels After training the student based on the soft targets of the teacher, the student is trained on the ground-truth labels using the cross entropy loss for a few epochs:

$$\begin{aligned} &\mathcal{L}_S^{II}(\theta_S(j)) = H_{CE}\big(y, S(x;\theta_S(j))\big) \\ &\theta_S(j+1) \leftarrow \theta_S(j) - \eta_S \nabla_{\theta_S} \mathcal{L}_S^{II}(\theta_S(j)). \end{aligned} \tag{4.4}$$

In summary, our technique enables the student to learn from the teacher more smoothly. Pro-KD is different from regular KD technique in the following aspects:

1. rather than distilling from a fully trained teacher; we distill from the teacher during its training;
2. in contrast to regular KD which applies a fixed temperature parameter to both networks, we apply an adaptive temperature factor only to the logits of the teacher;
3. we apply the KD loss and the cross entropy loss in two separate stages.

Moreover, it is worth mentioning that the overall number of training steps of our model follows regular KD techniques. For example, if in regular KD baselines, the model is trained for 30 number of epochs, we train the total training time of Stage I and Stage II of our model to 30 epochs as well.

Table 4.2: Comparing the test accuracy of Pro-KD, TAKD [91], Annealing-KD, RCO [70], regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	Teacher(110)	from scratch	93.8
	TA(20)	KD	92.39
	Student(8)	from scratch	88.44
	Student(8)	KD	88.45
	Student(8)	TAKD	88.47
	student(8)	RCO	88.90
	Student(8)	Annealing-KD	89.44
	Student(8)	**Pro-KD (ours)**	**90.01**

4.4.1 Why Does Pro-KD work?

Recall from Section 2.2 the generalized-distillation rate analysis of Lopez-Paz et al. [85], which gives the bound (Eq. 2.7) under which KD is beneficial. As discussed there, for large-capacity teachers the capacity term and the student–teacher approximation error can grow enough that the distillation bound exceeds the direct-learning bound, nullifying the benefits of distillation. The rate exponent $\alpha \in [\frac{1}{2}, 1]$ reflects the difficulty of the learning problem: $\alpha = 1$ corresponds to the fast (separable) regime and $\alpha = \frac{1}{2}$ to the slow regime. We use this bound only as *motivation*: α is determined by the separability of the student–teacher approximation problem, not by the optimizer or the training schedule, and Pro-KD does not directly control it.

We read the capacity-gap problem through the lens of this inequality. When the teacher's capacity is large (equivalently, the capacity gap is large), the bound in Eq. 2.7 is loose unless either the teacher's effective capacity is reduced (as TAKD [91] does by inserting a smaller TA network) or the student–teacher approximation error ε_l is made small. Pro-KD is aimed at the second lever: by exposing the student to a sequence of progressively sharper teacher checkpoints rather than a single fully converged one, we hypothesize that each successive matching problem is easier, so the accumulated approximation error is smaller than when matching the final teacher directly. We do not claim this changes the rate exponent α. It has been shown in [80, 16] that a smoother loss surface is easier to optimize than a sharp one, which is consistent with our use of the teacher's training trajectory and an adaptive temperature to smooth the student's training. A supporting analysis for the doubly smooth curriculum is given in Appendix B; it is likewise motivational rather than a formal guarantee.

Table 4.3: Comparing the test accuracy of Pro-KD, TAKD [91], Annealing-KD, RCO [70], regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	teacher(110)	from scratch	71.92
	TA(20)	KD	67.6
	student(8)	from scratch	61.37
	student(8)	KD	61.41
	student(8)	RCO	61.62
	student(8)	TAKD	61.82
	student(8)	Annealing KD	63.1
	student(8)	**Pro-KD (ours)**	**63.43**

4.5 Experiments and Results

In this section, we evaluate our Pro-KD on 3 different sets of experiments: on image classification, natural language understanding and question answering tasks. In all these three experiments, we compare Pro-KD with the state-of-the-art KD techniques which can address the capacity-gap problem such as TAKD [91], and Annealing-KD, or RCO [70] technique which does not require searching teacher checkpoints for KD. RCO was proposed and evaluated only on image classification and face recognition; the RCO rows in our natural-language-understanding tables are therefore our own reimplementation of the method for those tasks, trained under the same configuration as the other baselines. We also include baselines such as the original KD method [59] and also the regular training without KD.

4.5.1 Experimental Setup for Image Classification Tasks

Data We used CIFAR-10 and CIFAR-100 [75] datasets for our experiments on the image classification tasks. Both datasets have 60,000 of 32×32 color images distributed into 50,000 training and 10,000 test samples, with 10 classes for CIFAR-10 and 100 classes for CIFAR-100

Setup For these experiments, we used ResNet-8 as the student and resNet-110 as the teacher models. The experimental setups are similar to TAKD method [91]. Also for the TAKD baseline, we used ResNet-20 as the TA model. The results of these experiments

can be found in table 4.2 and 4.3 for comparison. For the baselines, first, the ResNet-110 teacher is trained from scratch on the given datasets and then it is used for training baselines with KD. For TAKD baseline, the original KD method is applied to train the TA network with the teacher network and it is applied to train the student network with TA network. For training the student with the proposed Pro-KD method, we trained the teacher for 160 epochs. For training the student, we used maximum temperature 10 and learning rate 0.1. For every 16 epochs, we decrease the temperature by 1 and $n_i = 1$.

Table 4.4: Dev set results of training DistilRoBERTa using the $\text{RoBERTa}_{\text{Large}}$ model on the GLUE benchmark. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy for the remaining tasks; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of CoLA, RTE, MRPC, STS-B, SST-2, QNLI, QQP, and the mean of the two MNLI accuracies, computed consistently for every row from the displayed cells.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
$\text{RoBERTa}_{\text{Large}}$ (Teacher)	67	85	91.63	92.53	96.21	94.53	91.45	89.94/89.97	88.54
DistilRoBERTa(NoKD)	61.9	69.31	89.85	88.46	91.86	91.31	90.04	84.03/ 83.69	83.32
Vanilla KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18/84.11	83.85
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	91.7	83.89/84.18	83.93
RCO	60.66	72.2	90.56	88.41	91.97	91.09	90.04	84.04/84.18	83.63
Annealing KD	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	84.52
Pro-KD (Ours)	62.14	73.64	91.9	88.8	92.66	91.47	91.53	84.83/84.84	**84.62**

Table 4.5: Test set results of DistilRoBERTa trained on the GLUE tasks using the $\text{RoBERTa}_{\text{Large}}$ teacher model. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
Vanilla KD	54.3	74.1	83.4	85.3	93.1	90.8	80.7	83.6/82.9	80.62
TAKD	53.2	74.2	84.7	85	93.2	91.0	80.7	83.8/83.2	80.69
RCO	55.1	73.0	87.85	84.4	93.5	88.9	78.66	83.2/82.4	80.53
Annealing KD	54	73.7	85.95	86.8	93.6	90.8	81.15	83.8/83.9	81.23
Pro-KD (Ours)	55.8	73.6	86.45	86.95	93.4	91.0	81.05	84.6/83.8	**81.56**

Table 4.6: BERT-Small results for Pro-KD on the GLUE dev set. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
$\text{BERT}_{\text{Large}}$	61.89	68.96	88.22	89.58	92.89	92.4	90.23	86.1/86.25	83.79
BERT-Small (NoKD)	44.05	64.98	83.75	87.41	88.3	86.49	88.43	78.42/78.57	77.74
Vanilla KD	43.28	64.98	84.96	85.95	88.65	86.75	88.24	78.62 /78.55	77.67
TAKD	43.79	65.7	83.98	86.44	88.88	86.78	88.4	78.78/ 78.64	77.84
RCO	44.32	65.7	84.91	85.48	88.99	86.32	87.52	78.35 /78.85	77.73
Annealing KD	45	63.9	87.09	87.04	89.56	86.99	88.58	78.66/78.23	78.33
Pro-KD	42.37	66.79	87.78	87.09	89.91	87.88	88.79	79.18/ 79.17	**78.72**

Table 4.7: The GLUE leaderboard test results of training BERT-Small from the $\text{BERT}_{\text{Large}}$ teacher. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, accuracy elsewhere; MNLI reports matched/mismatched accuracy. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
BERT-Small (NoKD)	41.3	62.6	79.7	80.05	89	86.3	78.05	78.3/ 77.6	74.37
Vanilla KD	37.3	63.4	80.55	78.15	90.2	86.5	78.25	78/76.5	73.95
TAKD	38.5	62.3	80.5	79.25	89.7	86.7	78	78.2/76.9	74.06
RCO	40.3	61.7	79.75	78.95	90.6	86.4	78.35	78.3/ 77.3	74.23
Annealing KD	38.6	63.1	81.85	80.6	91.2	87.3	78.35	77.8/77.4	74.83
Pro-KD	39	62.7	82.9	80.45	91.2	87.5	79.15	78.6/78.2	**75.16**

Results As it is shown in Tables 4.2 and 4.3, Pro-KD outperforms other baselines for both CIFAR-10 and CIFAR-100 experiments. Also, it is worth mentioning that the Annealing-KD technique is the second best result. RCO and TAKD both perform on-par with the regular KD and training the student from scratch.

4.5.2 Experimental Setup for Natural Language Understanding Tasks on the GLUE Benchmark

Data We use the General Language Understanding Evaluation (GLUE) benchmark [131], which consists of 9 natural language understanding tasks. The tasks cover textual entail-

Table 4.8: DistilRoBERTa results for Pro-KD on SQuAD.

KD Method	Squad 1.1	Squad 2.0
Teacher	93.7	87
Vanilla KD	85	73.65
TAKD	85.4	73.8
Pro-KD	**86**	**76**

Table 4.9: BERT-Small results for Pro-KD on SQuAD.

KD Method	Squad 1.1	Squad 2.0
Teacher	90.2	81
Vanilla KD	78.54	61.66
TAKD	78.5	61.66
Pro-KD	**79.7**	**62.76**

ment (RTE and MNLI), question-answer entailment (QNLI), paraphrase (MRPC), question paraphrase (QQP), sentiment (SST-2), textual similarity (STS-B), linguistic acceptability (CoLA), and Winograd Schema (WNLI).

Setup We perform experiments with multiple students of varying capacities. In the first experiment (Table 4.4 and 4.5), we use RoBERTa-large (24-layers) as teacher, DistilRoBERTa (6-layers) [110] as student, and RoBERTa-base (12-layers) [84] as the teacher-assistant for the TAKD baseline. For Pro-KD, we train the teacher for 5 epochs, and for training the student we use a maximum temperature of 5, $n_i = 2$, and the learning rate of 2e-5. For the second experiment (Table 4.6, 4.7), we use BERT-large (24-layers) as teacher, BERT-Small (4-layers) [126] as student, and BERT-base (12-layers) [36] as the teacher-assistant for TAKD. We train the teacher for 7 epochs, and for the student we use a maximum temperature of 7 for all tasks, and $n_i = 1$. For the learning rate, we use 5e-5 for RTE and MRPC, and 2e-5 for all other tasks. Additional details about other hyper-parameters can be found in the appendix. Every row of Tables 4.6 and 4.7, including the Annealing-KD baseline, was rerun for this chapter under the configuration stated above, so all methods there share the same student, teacher, schedule, and training budget. Because that configuration uses a single maximum temperature of 7 across tasks and a different learning rate for RTE and MRPC, the Annealing-KD entries differ from the corresponding BERT-Small results of Chapter 3, which used a per-task maximum temperature of 7 or 14. In the DistilRoBERTa tables (Tables 4.4 and 4.5), by contrast, the Vanilla-KD, TAKD

Table 4.10: Experimenting the impact of the adaptive temperature factor in Pro-KD using DistilRoBERTa trained with RoBERTa-large on the GLUE benchmark

KD Method	CoLA (8.5k)	RTE (2.5k)	MRPC (3.5k)	STS-B (5.7k)	SST-2 (67k)	QNLI (108k)	QQP (363k)	MNLI (392K) (392k)	WNLI	Avg
Pro-KD	**62.14**	**73.64**	**91.9**	**88.8**	**92.66**	91.47	91.53	84.83/84.84	**57.74**	**81.63**
Pro-KD w/o $\mathcal{T}$	58.66	70.03	91.74	88.0	92.43	**91.76**	**91.55**	**84.95/85.41**	56.33	80.61

and Annealing-KD rows are reproduced from Chapter 3 under that chapter's configuration, while the no-distillation, RCO and Pro-KD rows are our own runs.

Results We present our results in Tables 4.4, 4.5 and 4.6. Tables 4.4 and 4.6 compare the performance of Pro-KD with Vanilla KD and TAKD on the GLUE dev set, while Table 4.5 presents the results of DistilRoBERTa on the test set based on GLUE benchmark's leaderboard. We see that Pro-KD outperforms Vanilla KD and TAKD for both DistilRoBERTa and BERT-Small models. Even though TAKD is able to improve over Vanilla KD, the performance gain is much smaller compared to Pro-KD, demonstrating Pro-KD's high effectiveness in dealing with the large capacity-gap problem.

4.5.3 Experimental Setup for Question Answering

Data We use the Stanford Question Answering Datasets (SQuAD v1.1 [104] and SQuAD v2.0 [103]) which are a collection of 100k crowd-sourced question/answer pairs. Given a question and a passage from Wikipedia containing the answer, the task is to predict the answer text span in the passage.The SQuAD 2.0 task extends the SQuAD 1.1 problem definition by allowing for the possibility that no short answer exists in the provided paragraph.

Setup Similar to the GLUE experiments, we perform experiments on 2 different students, omitting the Annealing-KD and RCO baselines, which we did not extend to SQuAD. In the first experiment (Table 4.8), we use RoBERTa-large (24-layers) as teacher, DistilRoBERTa (6-layers) as student, and RoBERTa-base (12-layers) as the teacher-assistant for the TAKD baseline. We train the teacher for 3 epochs, and for the student we use a maximum temperature of 3, $n_i = 1$, and a learning rate of 3e-5 with a batch size of 12 for both Squad v1.1 and Squad 2.0. For the second experiment (Table 4.9), we use BERT-large (24-layers) as teacher, BERT-Small (4-layers) as student, and BERT-base (12-layers) as

the teacher-assistant for TAKD. We train the teacher for 3 epochs, and for the student we use a maximum temperature of 3, and $n_i = 1$. and a learning rate of 3e-5 with a batch size of 12. Additional details about other hyper-parameters can be found in the appendix.

Results We present our results in Tables 4.8 and 4.9. We again see that Pro-KD consistently outperforms Vanilla KD and TAKD for both Squad v1.1 and Squad 2.0 tasks.

4.5.4 Ablation Studies and Further Analysis

The Impact of the Temperature Factor Here we show the impact of having the adaptive temperature factor in our technique. The adaptive temperature helps the student model to be exposed to a softened version of the teacher in earlier stages. In this regard we repeated our experiment on the GLUE benchmark when the student is DistilRoBERTa and the teacher is RoBERTa-large. Table 4.10 shows the result of Pro-KD with and without the temperature. The results indicate that dropping the adaptive temperature will hamper the performance of our model by about 1% on the average GLUE score.

4.6 Conclusion

In this chapter, we addressed two practical limitations of knowledge distillation: the capacity-gap problem (Section 2.2) and the checkpoint-search problem. Building on the entropy-based curriculum perspective introduced in Chapter 3, we proposed Pro-KD, a progressive training procedure in which the student follows the teacher's training trajectory from the beginning rather than distilling from a single fully-trained checkpoint.

Using the generalized-distillation rate analysis of Lopez-Paz et al. [85] as motivation, we argued that following the teacher's trajectory makes each successive student–teacher matching problem easier and thus reduces the accumulated student–teacher approximation error; we do not claim this changes the rate exponent α, which is a property of the learning problem rather than the training schedule. Combined with an adaptive temperature schedule inspired by Annealing-KD, Pro-KD creates a doubly smooth curriculum: the teacher itself progresses from a simple to a complex model, while the temperature annealing further softens each snapshot's outputs.

We evaluated Pro-KD on image classification tasks (CIFAR-10 and CIFAR-100), the GLUE benchmark for natural language understanding, and SQuAD 1.1 and 2.0 for ques-

tion answering, where it outperformed Vanilla KD, TAKD, Annealing-KD, and RCO. Importantly, Pro-KD is simple, architecture-agnostic, does not require training any extra network, and avoids the need for a costly checkpoint search, though it does require access to the teacher's training trajectory and therefore does not apply when only a fixed pre-trained teacher is available.

Chapter 5

Continuation Knowledge Distillation

5.1 Overview

The previous two chapters addressed the capacity-gap problem (Section 2.2) from complementary angles: Chapter 3 introduced Annealing-KD, which controls the entropy of the teacher's supervisory signal through dynamic temperature scaling, and Chapter 4 proposed Pro-KD, which has the student follow the teacher's training trajectory to avoid the checkpoint-search problem. While both methods successfully mitigate the capacity gap, they share two limitations. First, Annealing-KD uses a rigid two-stage training structure that requires careful design of when to switch from the distillation phase to the fine-tuning phase. Second, neither method explicitly addresses the robustness of the student against noise in the teacher's output predictions, which can cause a capacity-limited student to overfit to high-frequency fluctuations rather than learning the teacher's underlying function.

In this chapter, we propose *Continuation Knowledge Distillation* (Continuation-KD), a unified training framework inspired by the continuation method from optimization theory. Continuation-KD generalizes Annealing-KD by (i) replacing the rigid two-stage structure with a single scheduled objective, and (ii) introducing an annealed hinge-loss margin that provides structural robustness against teacher noise. As we will show, Annealing-KD is a special case of Continuation-KD obtained by setting the margin to zero and using a step function for the objective transition.

5.2 Introduction

As established in the previous chapters, the capacity-gap problem [85, 91] and the challenge of effective knowledge transfer from large teachers to compact students remain central issues in KD (see Section 2.2 for a detailed treatment). The existing solutions have several complications: TAKD [91] requires training extra intermediate networks, and Annealing-KD [67] (Chapter 3) has a rigid two-stage structure that requires careful design for successful training. Moreover, none of these techniques is robust against noisy data and noisy teacher's outputs, which can distract the student with limited capacity from learning more useful features and make it overfit to the noise.

We propose a new solution, Continuation-KD, inspired by the continuation method from optimization and nonlinear equations. During training, we gradually move from a smoothed objective function toward the original one. Its central mechanism is a hinge-loss margin: once the student's residual to the scaled teacher falls below the margin, the distillation term is locally constant and contributes zero gradient and zero Hessian for that example, so residuals within the teacher's noise floor produce no KD update; the supervised term is unaffected. We also give two exact structural identities for the annealed objective, one for the gradient, one for the loss Hessian, that isolate the teacher's contribution in a single term scaled linearly by the annealing factor, and make precise what the hinge switches off.

We conduct extensive experiments on the GLUE benchmark for DistilRoBERTA (6-layer) [110] and BERT-small (4-layers) [126] core models and report the highest aggregate score among the evaluated baselines. Besides that, we demonstrate that in the computer vision setting continuation KD also outperforms its competitors. Overall, our contributions are the following:

1. We proposed a novel KD technique based on the Continuation method, which gradually increases the complexity of the loss function and empirically improves knowledge distillation in both computer vision and NLP tasks.

2. We implement a loss function like hinge loss function which makes a student trained with our technique robust against the teacher's output noise.

3. We show that the annealed hinge margin creates an exact local switch-off: below the annealed margin the continuation term's gradient and Hessian are identically zero, so residuals within the teacher's noise floor produce no distillation update, while the supervised term continues to contribute $\psi(i)\nabla\mathcal{L}_{CE}$. We make this precise with two

exact structural identities for the annealed objective, holding for any differentiable student, a gradient identity and the Hessian decomposition $H_S - \Phi\, C$, each isolating the teacher's contribution in a single term scaled linearly by the annealing factor and identifying what the hinge removes. These characterize the objective's structure; we are explicit that the link from this structure to the optimization and generalization gains observed on GLUE and CIFAR is not established by our experiments and is left as future work.

4. Our proposed method is simple and, unlike its competitors, does not have several stages. This feature makes our method stable and efficient.

5.3 Related Works

5.3.1 Knowledge Distillation and the Capacity Gap

The capacity-gap problem in KD and its existing solutions, including TAKD [91] and Annealing-KD [67], are discussed in detail in Section 2.2 and Chapters 3–4. While Annealing-KD mitigates the capacity gap effectively, it has two remaining limitations that motivate this chapter: (i) its rigid two-stage training structure requires *a priori* decisions on when to switch from the distillation phase to the fine-tuning phase, and (ii) it does not explicitly address robustness against noisy data and noisy teacher outputs, which can cause the student to overfit to high-frequency fluctuations.

5.3.2 Continuation Optimization

Continuation method was first proposed as a numerical method for solving nonlinear equations [33] and then it was adopted as a heuristic for nonconvex optimization [136]. The main intuition of the continuation method is to include the problem we are trying to solve in a continuous family of problems, such that one of the members of this family is easy to solve and its solution could be pulled over to give an approximate solution of the original problem. Machine Learning community applied the continuation idea for training neural networks. [93] proposed the first theoretical analysis of the bound on the approximate solution given by the continuation optimization. [52] suggested optimizing highly non-convex neural networks by starting with a smoothed objective function and making it more complex over the training. We adapted this general method for Knowledge Distillation.

5.4 Background: From Annealing-KD to Continuation-KD

We adopt the standard KD formulation from Section 2.1 (Eq. 2.3). The capacity-gap problem and TAKD are covered in Section 2.2. Here we briefly recall the Annealing-KD loss (Chapter 3), as Continuation-KD directly generalizes it.

Annealing-KD Recap. Annealing-KD [67] defines a monotonically increasing function $\Phi\colon \mathbb{N} \to [0, 1]$, going from zero at the beginning of training to one at the end. In the first stage, the student matches the teacher's logits scaled by Φ:

$$\mathcal{L}(i) = ||z_s(x) - \Phi(i) z_t(x)||_2^2, \tag{5.1}$$

where i is the training epoch, $z_s(x)$ and $z_t(x)$ are the logit outputs of the student and teacher respectively. In the second stage, the best checkpoint from stage one is fine-tuned on hard labels using cross-entropy for m epochs. The hyperparameters k (stage-one epochs) and m must be chosen before training, creating the rigid two-stage structure that Continuation-KD eliminates.

5.5 Methodology

5.5.1 The Structure of the Annealed Objective and the Hinge Mechanism

Continuation-KD inherits its schedule from the continuation method for non-convex optimization, and specifically from *graduated non-convexity* [52, 93]: optimization begins on a smoothed surrogate objective, and the smoothing is gradually removed until the original objective is recovered. Here the surrogate is the loss against the $\Phi(\mathcal{T})$-scaled teacher logits, with Φ increasing from near zero to one over training; Chapter 3 showed that this same scaling also acts as an entropy curriculum on the target. Whether the smoothing *eases* the student's optimization, makes the landscape measurably less rugged, is an empirical question we do not attempt to settle. What we state exactly is two things: how the annealing factor enters the objective's gradient and curvature (the two decompositions below), and how the hinge margin creates a region in which the teacher's contribution to the update is switched off entirely (Corollary 5.5). The hinge is the mechanism Continuation-KD is

built around; the decompositions are exact structural accounting that makes precise what the hinge switches off. The empirical case for the method rests on the task results of Section 5.6.

Notation

Let $z_s(x;\theta) \in \mathbb{R}^K$ denote the pre-softmax logit vector of the student network parameterized by weights $\theta \in \mathbb{R}^d$, and let $z_t(x) \in \mathbb{R}^K$ denote the static logits of the pre-trained teacher. The Jacobian of the student logits with respect to the weights is $J_\theta \in \mathbb{R}^{K\times d}$, whose k-th row is $(\nabla_\theta z_{s,k})^T$. The annealed MSE loss is

$$\mathcal{L}_\mathcal{T}(\theta) = \frac{1}{2}\sum_{k=1}^{K}\big(z_{s,k}(x;\theta) - \Phi(\mathcal{T})\, z_{t,k}(x)\big)^2, \tag{5.2}$$

where $0 \le \Phi(\mathcal{T}) \le 1$ is the monotonically increasing annealing function.

Gradient decomposition

The first result relates the gradient of the annealed loss to that of the unscaled loss, for *any* differentiable student.

Theorem 5.1 (Gradient decomposition of the annealed loss)**.** *For any differentiable student network, the gradient of the annealed loss and the gradient of the unscaled loss ($\Phi = 1$) are related by the exact identity*

$$\nabla_\theta \mathcal{L}_\mathcal{T}(\theta) = \nabla_\theta \mathcal{L}_1(\theta) + (1 - \Phi(\mathcal{T}))\; J_\theta^T\, z_t. \tag{5.3}$$

The annealing factor therefore enters the gradient through a single term. Writing the gradient directly as $\nabla_\theta \mathcal{L}_\mathcal{T} = J_\theta^T z_s - \Phi(\mathcal{T}) J_\theta^T z_t$, the explicitly teacher-dependent term is $-\Phi(\mathcal{T})\, J_\theta^T z_t$; the term $(1 - \Phi(\mathcal{T}))\, J_\theta^T z_t$ appearing in (5.3) is the correction relative to the unscaled $\Phi = 1$ objective, which vanishes as $\Phi(\mathcal{T}) \to 1$ and grows as the target is smoothed. At a parameter value where $z_s(x;\theta) = \mathbf{0}$ the identity reduces to the multiplicative form $\nabla_\theta \mathcal{L}_\mathcal{T} = \Phi(\mathcal{T})\, \nabla_\theta \mathcal{L}_1$, but ordinary random initialization does not satisfy $z_s = \mathbf{0}$ (biases and nonzero weights produce nonzero logits), so that special case does not apply in the settings we train. Nothing here implies that $\|\nabla_\theta \mathcal{L}_\mathcal{T}\|$ is monotone in $\Phi(\mathcal{T})$. We therefore do not rely on the multiplicative form; the exact additive identity (5.3) is what we use.

Proof. The gradient of the annealed MSE loss is $\nabla_\theta \mathcal{L}_\mathcal{T} = J_\theta^T(z_s - \Phi\, z_t)$. Adding and subtracting $J_\theta^T z_t$,

$$\nabla_\theta \mathcal{L}_\mathcal{T} = J_\theta^T(z_s - z_t) + (1-\Phi)\, J_\theta^T z_t = \nabla_\theta \mathcal{L}_1 + (1-\Phi)\, J_\theta^T z_t.$$

When $z_s = 0$, $\nabla_\theta \mathcal{L}_\mathcal{T} = -\Phi\, J_\theta^T z_t = \Phi \cdot \nabla_\theta \mathcal{L}_1$. □

Equation (5.3) is an algebraic rearrangement, and the rearrangement is precisely its use: the annealing factor multiplies exactly one interpretable quantity, the teacher logits pulled back through the student Jacobian, and no other part of the gradient. At a fixed parameter point, Φ therefore controls the explicitly teacher-dependent term $-\Phi J_\theta^T z_t$ linearly, and the correction $(1-\Phi)J_\theta^T z_t$ in Eq. (5.3) is the change relative to unscaled teacher-logit matching. Taking $\Phi = 1$ recovers unscaled teacher-logit matching, *not* hard-label training: hard-label-only training is recovered in Eq. (5.10) when $\psi(i) = 1$. The implemented $\mathcal{T}_i$ schedule is integer-valued, so Φ advances in steps rather than continuously.

The second result is the analog for the curvature.

Hessian decomposition

We characterize the curvature of the annealed objective through the spectral norm of its Hessian [73, 80], and derive an *exact* decomposition that separates the student-only and teacher-induced contributions.

Definition 5.2 (Student-only Hessian and teacher–student coupling)**.**

$$H_S(\theta) = J_\theta^T J_\theta + \sum_{k=1}^{K} z_{s,k}(x;\theta)\, \nabla^2 z_{s,k}(x;\theta), \tag{5.4}$$

$$C(\theta) = \sum_{k=1}^{K} z_{t,k}(x)\, \nabla^2 z_{s,k}(x;\theta). \tag{5.5}$$

$H_S(\theta)$ is the Hessian of $\frac{1}{2}\|z_s(x;\theta)\|_2^2$ with respect to θ. It depends *only* on the student and is independent of $\Phi(\mathcal{T})$. $C(\theta)$ captures the teacher–student coupling and is also independent of $\Phi(\mathcal{T})$.

Theorem 5.3 (Exact Decomposition of Landscape Curvature)**.** *Assume $z_s(x;\theta)$ is twice continuously differentiable in θ on a neighbourhood of the parameter point being evaluated,*

and hold x, $z_t(x)$, and $\Phi(\mathcal{T})$ fixed under differentiation. The Hessian of the annealed MSE loss (5.2) admits the exact decomposition

$$\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) = H_S(\theta) - \Phi(\mathcal{T}) \cdot C(\theta). \tag{5.6}$$

In particular:

1. *The Hessian is an **affine function** of $\Phi(\mathcal{T})$: the teacher's contribution to landscape curvature enters **exactly linearly**.*
2. *At $\Phi(\mathcal{T}) = 0$ (maximally smoothed targets), $\nabla^2 \mathcal{L}_0 = H_S(\theta)$: the landscape curvature is determined entirely by the student.*
3. *At $\Phi(\mathcal{T}) = 1$ (standard KD), $\nabla^2 \mathcal{L}_1 = H_S(\theta) - C(\theta)$.*
4. *The rate of curvature change with respect to the annealing parameter is constant: $\frac{\partial}{\partial \Phi} \nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) = -C(\theta)$.*

For the spectral norm:

$$\left\| \nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) \right\|_2 \leq \left\| H_S(\theta) \right\|_2 + \Phi(\mathcal{T}) \left\| C(\theta) \right\|_2. \tag{5.7}$$

The matrix $\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta)$ is affine in $\Phi(\mathcal{T})$ at a fixed θ. Because $C(\theta)$ is in general neither positive nor negative semidefinite, this does not *imply that the spectral norm $\|\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta)\|_2$ is monotone or linear in $\Phi(\mathcal{T})$ at a fixed point; decreasing $\Phi(\mathcal{T})$ can increase, decrease, or leave the spectral norm unchanged.*

Proof. Starting from the exact Hessian of the annealed MSE (derived in Appendix C):

$$\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) = J_\theta^T J_\theta + \sum_{k=1}^{K} \big(z_{s,k} - \Phi(\mathcal{T})\, z_{t,k} \big) \nabla^2 z_{s,k}. \tag{5.8}$$

Expanding the residual:

$$\begin{aligned} \nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) &= J_\theta^T J_\theta + \sum_{k=1}^{K} z_{s,k} \nabla^2 z_{s,k} - \Phi(\mathcal{T}) \sum_{k=1}^{K} z_{t,k} \nabla^2 z_{s,k} \\ &= \underbrace{\Big(J_\theta^T J_\theta + \sum_{k=1}^{K} z_{s,k} \nabla^2 z_{s,k} \Big)}_{H_S(\theta)} - \Phi(\mathcal{T}) \underbrace{\sum_{k=1}^{K} z_{t,k} \nabla^2 z_{s,k}}_{C(\theta)}. \end{aligned} \tag{5.9}$$

No inequality has been applied; the decomposition is exact. The spectral-norm bound (5.7) follows from a single application of the triangle inequality at the matrix level. □

Remark 5.4 (What the decomposition does and does not give)**.** *The decomposition* (5.6) *isolates the teacher's entire influence on the curvature in the single matrix $C(\theta)$, scaled linearly by $\Phi(\mathcal{T})$: at $\Phi = 0$ the curvature is the student's alone, and the teacher's contribution enters linearly as the target is sharpened toward $\Phi = 1$. This is an exact matrix identity. It constrains the Hessian* matrix *but not its* spectral norm*: because $C(\theta)$ need not be positive or negative semidefinite, and its eigenspaces need not align with those of $H_S(\theta)$, the identity implies neither monotonicity nor linearity of $\|\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta)\|_2$ in $\Phi(\mathcal{T})$; decreasing $\Phi(\mathcal{T})$ can increase, decrease, or leave the spectral norm unchanged. The decomposition is a structural description of the objective, and that is all we claim from it.*

The hinge margin: an exact local switch-off

The mechanism Continuation-KD is designed around is local and exact. The loss adds an annealed hinge margin, so that once the student's residual to the scaled teacher falls below the margin, the continuation term is exactly flat there, zero gradient and zero Hessian, and the teacher's contribution to the update is switched off; the complete objective remains non-flat because the supervised term still contributes whenever $\psi(i) > 0$. This is immediate from the definition of the hinge.

Corollary 5.5 (Local flatness of the Continuation-KD hinge objective)**.** *Let $\mathcal{L}_{CNT}(\theta) = \max\{0, \mathcal{L}_{\mathcal{T}}(\theta) - \Phi(\mathcal{T})m\}$ be the Continuation-KD objective, where $m > 0$ is a fixed margin, $\Phi(\mathcal{T})$ is the monotonic annealing function, and $\mathcal{L}_{\mathcal{T}}(\theta)$ is the annealed MSE loss. Then:*

1. *In the* active region *where $\mathcal{L}_{\mathcal{T}}(\theta) > \Phi(\mathcal{T})m$, the gradient and Hessian of $\mathcal{L}_{CNT}$ coincide with those of $\mathcal{L}_{\mathcal{T}}$, so the exact decomposition $\nabla^2 \mathcal{L}_{CNT} = H_S(\theta) - \Phi(\mathcal{T})\, C(\theta)$ applies directly.*

2. *In the* inactive region *where $\mathcal{L}_{\mathcal{T}}(\theta) < \Phi(\mathcal{T})m$, both the gradient and Hessian of $\mathcal{L}_{CNT}$ are exactly zero, creating a locally flat plateau.*

3. *At the boundary $\mathcal{L}_{\mathcal{T}}(\theta) = \Phi(\mathcal{T})m$ the objective is in general non-differentiable; its Clarke subdifferential is $\partial \mathcal{L}_{CNT}(\theta) = \{\lambda \nabla \mathcal{L}_{\mathcal{T}}(\theta) : \lambda \in [0, 1]\}$ and a classical Hessian need not exist.*

The statement concerns the hinge component $\mathcal{L}_{CNT}$ in isolation. In the full objective (5.10) *the inactive region gives $\nabla \mathcal{L} = \psi(i)\, \nabla \mathcal{L}_{CE}$ and $\nabla^2 \mathcal{L} = \psi(i)\, \nabla^2 \mathcal{L}_{CE}$, which are generally nonzero whenever $\psi(i) > 0$.*

Proof. **Case 1 (Active region):** When $\mathcal{L}_{\mathcal{T}}(\theta) > \Phi(\mathcal{T})m$, the hinge evaluates to $\mathcal{L}_{CNT}(\theta) = \mathcal{L}_{\mathcal{T}}(\theta) - \Phi(\mathcal{T})m$. Since $\Phi(\mathcal{T})m$ is constant with respect to θ: $\nabla\mathcal{L}_{CNT}(\theta) = \nabla\mathcal{L}_{\mathcal{T}}(\theta)$ and $\nabla^2\mathcal{L}_{CNT}(\theta) = \nabla^2\mathcal{L}_{\mathcal{T}}(\theta) = H_S(\theta) - \Phi(\mathcal{T})C(\theta)$.

Case 2 (Inactive region): When $\mathcal{L}_{\mathcal{T}}(\theta) < \Phi(\mathcal{T})m$, continuity of $\mathcal{L}_{\mathcal{T}}$ gives a neighbourhood of θ on which the same strict inequality holds, so the loss is identically zero there: $\mathcal{L}_{CNT} \equiv 0$. Therefore $\nabla\mathcal{L}_{CNT}(\theta) = \mathbf{0}$ and $\nabla^2\mathcal{L}_{CNT}(\theta) = \mathbf{0}$.

Case 3 (Boundary): $\mathcal{L}_{CNT} = \max\{0, g(\theta)\}$ with $g(\theta) = \mathcal{L}_{\mathcal{T}}(\theta) - \Phi(\mathcal{T})m$ smooth. At a point where $g(\theta) = 0$, the max-rule for Clarke subdifferentials gives $\partial\mathcal{L}_{CNT}(\theta) = \mathrm{conv}\{\mathbf{0}, \nabla g(\theta)\} = \{\lambda\nabla\mathcal{L}_{\mathcal{T}}(\theta) : \lambda \in [0,1]\}$, and $\mathcal{L}_{CNT}$ is differentiable there only if $\nabla\mathcal{L}_{\mathcal{T}}(\theta) = \mathbf{0}$. □

Remark 5.6 (Noise suppression mechanism)**.** *The inactive region provides a concrete mechanism for noise suppression: when the student's residual error relative to the scaled teacher falls below the annealed margin $\Phi(\mathcal{T})m$, the hinge component contributes no gradient for that example; when $\psi(i) > 0$ the supervised term still updates the student. Small, noisy fluctuations in the teacher's logits, provided their induced MSE deviation stays below the margin, therefore transmit no KD gradient to the weights. The mechanism is amplitude-based rather than frequency-selective: it switches off residuals that are small, whatever their origin, so whether it in fact prevents fitting teacher noise depends on the margin, the loss reduction, the optimization path, and the model, and is assessed empirically rather than guaranteed by the corollary.*

This flat region is the concrete way the annealing factor is put to use, and the two decompositions above say exactly what falls inside it: within the inactive region the entire KD gradient $J_\theta^T(z_s - \Phi(\mathcal{T})z_t)$ for that example is switched off, including the teacher term $\Phi(\mathcal{T})\,C(\theta)$ in the curvature. Continuation-KD's design question is therefore not whether annealing flattens the landscape, but how large the switched-off region should be, which the margin m controls; we return to this in Section 5.5.2.

5.5.2 Loss Formulation and Training Procedure

In this section, we describe the Continuation-KD loss, which addresses both the capacity-gap problem and the lack of robustness against noise in the teacher's output. Continuation-KD uses a loss function with two objectives (Eq. 5.10). The first objective $\mathcal{L}_{CE}$ is a cross-entropy loss term that trains the student based on the given hard labels. The second objective $\mathcal{L}_{KD}^{CNT}$ is our proposed annealed hinge loss function that gradually trains the student to mimic the behavior of the teacher. Inspired by the continuation method [52],

Continuation-KD starts with an easy objective to train at the beginning of training. As the training proceeds, the whole objective function becomes more and more complex. Formally, the loss function of the Continuation-KD is defined as follows:

$$\mathcal{L} = (\psi(i))\mathcal{L}_{CE} + (1 - \psi(i))\mathcal{L}_{KD}^{CNT} \tag{5.10}$$

where $1 \leq i \leq n$ indicates the epoch index with the maximum number of epochs n. The $0 \leq \psi(i) \leq 1$ is a nondecreasing function that starts at a small value at the first epoch and rises to one by the end of training. The schedules actually used in our experiments are given in Eqs. (5.16) and (5.17); both begin at a small positive value rather than exactly zero, so the supervised term carries a small weight from the first epoch onward. L_{KD}^{CNT} defines as:

$$\mathcal{L}_{KD}^{CNT} = \max\{0, \|z_s - \Phi(\mathcal{T}_i)z_t\|_2^2 - m\ \Phi(\mathcal{T}_i)\} \tag{5.11}$$

where z_s and z_t are the output logits of the student and teacher networks *for a single training example*, and the norm is taken over that example's K logits; m is the margin factor; $1 \leq \mathcal{T}_i \leq \tau_{\max}$ is the annealing index, $\tau_{\max}$ is its maximum value, and $\Phi(\mathcal{T}_i)$ takes values in $[1/\tau_{\max}, 1]$. The map is decreasing in $\mathcal{T}_i$; because $\mathcal{T}_i$ decreases over training, Φ is nondecreasing with the epoch index, reaching one only when $\mathcal{T}_i = 1$. Its smallest value is $1/\tau_{\max}$ rather than zero, so the first target is close to zero only when $\tau_{\max}$ is large. Since scaling the logits by Φ corresponds to an effective softmax temperature $1/\Phi$, we refer to $\mathcal{T}_i$ as an annealing index rather than a literal temperature. We define this function as:

$$\Phi(\mathcal{T}_i) = 1 - \frac{\mathcal{T}_i - 1}{\tau_{\max}}, 1 \leq \mathcal{T}_i \leq \tau_{\max}, \mathcal{T}_i \in \mathbb{N}. \tag{5.12}$$

Note that in Eq. 5.11, $\|z_s - \Phi(\mathcal{T}_i)z_t\|_2^2$ is a summed squared-error loss over the K logits between the student's logits and the annealed version of the teacher's logits (a mean-squared loss would divide this term by K; the two differ only by the constant factor K, which is absorbed into the loss weight and the margin m).

Loss normalization. Two scalings of the same squared error appear in this chapter, and it is worth stating their relationship once. The analysis of Section 5.5.1 uses the half-sum $\mathcal{L}_{\mathcal{T}} = \frac{1}{2}\sum_k (z_{s,k} - \Phi z_{t,k})^2$ of Eq. (5.2), which removes a factor of two from the gradient and Hessian identities, whereas the implemented objective in Eq. (5.11) and Algorithm 2 uses the unhalved sum. The two differ by the constant factor two, so the implemented

gradient and Hessian are exactly twice those of Theorems 5.1 and 5.3, and every structural conclusion (the affine dependence on Φ, the isolation of the teacher term, and the vanishing of both derivatives inside the hinge's inactive region) is unaffected. The one quantity that does depend on the convention is the margin: the calibration derived below gives $m = K\sigma^2$ for the unhalved sum used in Eq. (5.11), and $m = K\sigma^2/2$ under the half-sum convention. Throughout, m is quoted in the units of Eq. (5.11).

Logit gauge. A second convention is implicit rather than chosen. Logits are determined only up to an additive constant, and scaling by $\Phi(\mathcal{T}_i)$ rescales such an offset rather than removing it, since $\Phi(z_t + c\mathbf{1}) = \Phi z_t + \Phi c\,\mathbf{1}$. The residual $\|z_s - \Phi z_t\|_2^2$ is therefore not invariant to the teacher's logit gauge, and neither is the set of examples whose residual falls inside the annealed margin Φm: two teachers that produce identical probabilities but logits differing by a constant can leave the hinge inactive in one case and active in the other. The experiments reported here use the uncentered logits of the stated checkpoints; evaluating centered or standardized logits is left to future work.

Batch reduction. Equations (5.10) and (5.11) define the loss for one training example. The implemented mini-batch objective applies the hinge *per example* and then averages over the batch,

$$\mathcal{L}_{KD}^{CNT}(X) \;=\; \frac{1}{|X|}\sum_{x\in X}\max\Big\{0,\;\big\|z_s(x) - \Phi(\mathcal{T}_i)\,z_t(x)\big\|_2^2 - \Phi(\mathcal{T}_i)\,m\Big\}, \tag{5.13}$$

and likewise averages $\mathcal{L}_{CE}$ over the batch. The distinction matters: applying a single hinge to the summed batch residual would threshold a quantity of order $|X|K\sigma^2$ against a margin calibrated for one example, and an example inside the noise floor would still receive gradient whenever any other example in the batch lay outside it. The per-example reduction is what makes the local switch-off of Corollary 5.5 hold for each example separately. Also, $\max\{0, \|z_s - \Phi(\mathcal{T}_i)z_t\|_2^2 - \Phi(\mathcal{T}_i)m\}$ is a hinge loss with an annealed margin $\Phi(\mathcal{T}_i)m$. As shown in Corollary 5.5, this hinge formulation clips gradient updates when the residual error falls below the annealed margin, creating locally flat regions in the loss landscape. This loss function avoids penalizing negligible differences (the ones less than $m\ \Phi(\mathcal{T}_i)$) between the outputs of the student and teacher. This feature helps the student to learn a meaningful behavior of the teacher rather than focusing on higher frequency fluctuations.

Choosing the margin. The margin m sets the squared-error scale below which residuals are ignored; the inactive region is the Euclidean ball around Φz_t of radius $\sqrt{\Phi m}$ in the units

of Eq. (5.11), so m is an energy rather than a radius. A useful scale comes from a simple noise model: if the teacher logits are $z_t = z_t^* + \epsilon$ with $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ i.i.d., then a student that matches the scaled clean target has residual $\|z_s - \Phi\, z_t\|_2^2 = \Phi^2\|\epsilon\|_2^2$ with expectation $\Phi^2 K\sigma^2$. Requiring the annealed margin $\Phi\, m$ to cover this expected noise residual for all Φ gives the energy scale

$$m = K\sigma^2. \tag{5.14}$$

We use (5.14) as an order-of-magnitude starting point; the actual margin, and the choice between a constant margin m and an annealed margin $\Phi(\mathcal{T}_i)\, m$, are set by cross-validation in each experiment. Two caveats apply. First, $K\sigma^2$ matches the *expected* noise energy and is not a coverage guarantee: a high-probability rule would use a chi-squared quantile, $m = \sigma^2\chi^2_{K,1-\delta}$, which gives $\Pr[\mathcal{L}_\mathcal{T} \le \Phi m] \ge 1 - \delta$ for every $\Phi \in (0, 1]$. Second, when the clean mismatch $z_s - \Phi z_t^*$ is nonzero the hinge responds to the total residual, so its activation cannot be attributed to teacher noise alone.

At the beginning of training, we set $\mathcal{T}_1 = \tau_{\max}$ which leads to the most softened version of the teacher's output ($\Phi(\mathcal{T}_1) = \frac{1}{\tau_{\max}}$). Since $\psi(1)$ is small, the student learns almost entirely from the teacher's smoothest version at this point, which is an easy target to learn. Then, during training, we decrease the temperature. At this phase, functions $\psi(i)$ and $\Phi(\mathcal{T}_i)$ are both increasing which in turn leads to increasing the sharpness of the teacher and shifting from the hinge loss to the cross-entropy loss. Both of these operations increase the complexity of the whole loss function. The transition is scheduled rather than continuous throughout: Φ advances in integer steps of the annealing index, and the NLU schedule of Eq. (5.17) rises to 0.5 at epoch 20 and then changes discontinuously to 1 at epoch 21. Under both printed schedules the KD term is switched off before Φ reaches 1, so the objective does not pass through a phase of unscaled teacher matching with the KD term still active. Figure 5.1 visualizes different components of Continuation-KD.

Also note that, if we set $m = 0$ and $\psi(i)$ to the step function in Eq 5.15, then Continuation-KD recovers Annealing-KD's two phase-specific loss types, where k is the number of epochs in the first stage and $n - k$ is the number of epochs in the second stage. The equivalence is at the level of the objective; matching the full procedure additionally requires the same target-scale schedule, loss normalization, epoch counts, validation criterion, and the transition through the best Stage-I checkpoint that Annealing-KD performs. This fact shows that the Annealing-KD is actually a special case of our Continuation-KD.

$$\psi(i) = \begin{cases} 0, & 1 \le i \le k \\ 1, & k < i \le n \end{cases} \tag{5.15}$$

Algorithm 2 demonstrates the details of Continuation-KD. It requires a student S, a

frozen teacher $\mathcal{F}$, training data D_{train}, validation data D_{val}, a maximum annealing index $\tau_{\max}$, a number of epochs n, a nondecreasing schedule ψ, and a margin m as inputs, and returns the trained student at the output. We write the teacher as $\mathcal{F}$ here to keep it distinct from the annealing index, which is denoted q in the pseudocode. At the beginning, it sets $q = \tau_{\max}$ to get the maximum smoothness of the teacher. Also, variable k indicates the number of epochs before updating q during training. The update interval is $k = \max\{1, \lfloor n/\tau_{\max} \rfloor\}$ and the index is clamped at $q \geq 1$, so the schedule is well defined for every $n \geq 1$ and Φ never leaves $[1/\tau_{\max}, 1]$; in all reported experiments $n \geq \tau_{\max}$ and the clamp is inactive except on the final epochs, which run at $q = 1$ (full teacher). Φ and Ψ are the output values of $\Phi(q)$ and $\psi(i)$. The inner loop iterates over the mini-batches of one full epoch of D_{train}; each mini-batch (X, Y) feeds the two loss terms, whose linear combination gives the continuation loss $\mathcal{L}$. Finally, $\mathcal{L}$ is fed into Optimization-Back-Propagation(.) function to optimize the student network based on the back propagation of the gradient of this loss function and update the weights of the student. This part of the training is identical to regular training of the neural networks. Save-Best-Checkpoint(.) function evaluates the current student on D_{val} at the end of each epoch. If it is better than the previous checkpoints, it saves the checkpoint. In the end, we load the best checkpoint and return it. In the next section, we report the experimental results of Continuation-KD method.

5.6 Experiments

This section demonstrates our evaluation results comparing Continuation-KD with other baselines for natural language processing and computer vision tasks. We compare our method with state-of-the-art techniques such as annealing-KD [67], TAKD [91] and other baselines like Vanilla-KD [59] and training the student only with hard labels. In the following sub-sections, we will discuss each in more detail.

5.6.1 Hardware Details

We trained all our baselines using a single NVIDIA V100 GPU. All experiments were run using the PyTorch framework and for NLP experiments we used HuggingFace API.

Algorithm 2 Continuation Knowledge Distillation

function CONTINUATION-KD(S, $\mathcal{F}$, D_{train}, D_{val}, $\tau_{\max}$, n, $\psi(\cdot)$, m)
  $q \leftarrow \tau_{\max}$ ▷ annealing index, not a softmax temperature
  $k = \max\{1,\ \lfloor \frac{n}{\tau_{\max}} \rfloor\}$ ▷ guard against $n < \tau_{\max}$
  for $i = 1$ `to` n **do**
    $\Phi \leftarrow 1 - \frac{q-1}{\tau_{\max}}$, $\Psi \leftarrow \psi(i)$
    for each mini-batch (X, Y) from D_{train} **do**
      $\mathcal{L}_{CE} \leftarrow \frac{1}{|X|} \sum_{(x,y)\in(X,Y)} H_{CE}(\sigma(S(x)), y)$
      $\mathcal{L}_{KD}^{CNT} \leftarrow \frac{1}{|X|} \sum_{x\in X} \max\{0, \|S(x) - \Phi\mathcal{F}(x)\|_2^2 - m\ \Phi\}$ ▷ hinge per example, then average
      $\mathcal{L} = \Psi\mathcal{L}_{CE} + (1 - \Psi)\mathcal{L}_{KD}^{CNT}$
      OPTIMIZATION-BACK-PROPAGATION($\mathcal{L}$)
    end for
    SAVE-BEST-CHECKPOINT(S, D_{val})
    if i mod $k = 0$ **then**
      $q \leftarrow \max\{1,\ q - 1\}$ ▷ decrement at epoch end; clamp so $q \geq 1$, hence $\Phi \in [1/\tau_{\max}, 1]$
    end if
  end for
  $S \leftarrow$ LOAD-BEST-CHECKPOINT( )
  return S
end function

5.6.2 Image Classification

For the image classification tasks, we used CIFAR-10 and CIFAR-100 datasets with 10 and 100 classes respectively. Both of these datasets have 60,000 data samples, and each of them is a 32×32 pixel color image. Also, both of these datasets have 50,000 train and 10,000 test samples. In all of our computer vision (CV) experiments, ResNet-8 and ResNet-110 models are used as the student and the teacher respectively. For the TAKD baseline, the ResNet-20 model is used as the TA model. Continuation-KD follows the same CIFAR training configuration as the Annealing-KD experiments of Chapter 3; the batch size, learning rate, optimizer, weight decay, and other settings are those listed in Appendix A. Continuation-KD is a single-stage schedule and runs for 160 epochs, the same total budget as the baselines. We use a maximum annealing index of $\tau_{\max} = 20$. The baseline rows of Tables 5.1 and 5.2 are reproduced from that chapter rather than rerun for this one, so the comparison is made under a shared training configuration. Also, the

following $\psi(\cdot)$ function used in these experiments:

$$\psi(i) = \begin{cases} \frac{i}{150}, & 1 \leq i \leq 150 \\ 1, & i \geq 150 \end{cases} \tag{5.16}$$

The CIFAR-10 and CIFAR-100 experiments results are reported in Tables 5.1 and 5.2. We can see that Continuation-KD clearly outperforms other baselines. Also, Annealing-KD achieved second-best results in these experiments. However, other baselines showed almost similar performances.

Table 5.1: Comparing the test accuracy of Continuation-KD, Annealing-KD, TAKD, Vanilla-KD, and finetuning on CIFAR-10 dataset with ResNet model

Type	Training method	Accuracy
Teacher(110)	from scratch	93.8
TA(20)	KD	92.39
Student(8)	from scratch	88.44
Student(8)	KD	88.45
Student(8)	TAKD	88.47
Student(8)	Annealing-KD	89.44
student(8)	**Continuation-KD**	**90.21**

Table 5.2: Comparing the test accuracy of Continuation-KD, Annealing-KD, TAKD, Vanilla-KD, and finetuning on CIFAR-100 dataset with ResNet model

Type	Training method	Accuracy
teacher(110)	from scratch	71.92
TA(20)	KD	67.6
student(8)	from scratch	61.37
student(8)	KD	61.41
student(8)	TAKD	61.82
student(8)	Annealing-KD	63.1
student(8)	**Continuation-KD**	**64.2**

5.6.3 Natural Language Understanding

For the NLU experiments, we use the General Language Understanding Evaluation (GLUE) benchmark [131]. The benchmark contains several tasks, including textual entailment (RTE and MNLI), question-answer entailment (QNLI), paraphrase (MRPC), question paraphrase (QQP), sentiment (SST-2), textual similarity (STS-B), linguistic acceptability (CoLA), and Winograd Schema (WNLI). Two types of students with different capacities are used in these experiments. In our first experiment, we used distilRoBERTa with 6 layers as our student and RoBERTa-Large (24 layers) as our teacher. Also, for the TAKD baseline, we used RoBERTa-base (12 layers) as the teacher assistant model. For this experiment, the maximum temperature was 10, learning rate was 2e-5, and batch size was 64. Also, we used the pre-trained distilRoBERTa as our student and we fine-tuned it for 30 epochs. The ψ function for this experiment is defined as following:

$$\psi(i) = \begin{cases} \frac{i}{20}, & 1 \leq i \leq 20 \\ 1, & 20 < i \leq 30 \end{cases} \tag{5.17}$$

Tables 5.3 and 5.4 show the results of this experiment on the dev set and test set. As shown in these tables, Continuation-KD achieved superior results in most of the tasks and it improves the previous baselines with a good overall average score. The Finetune, Vanilla-KD, TAKD, and Annealing-KD rows of Tables 5.3 and 5.4 are reproduced from the corresponding DistilRoBERTa experiments of Chapter 3 rather than rerun for this chapter, and were obtained under that chapter's configuration (Appendix A), which uses a different maximum annealing index and a different epoch budget from the Continuation-KD run reported here. These two tables therefore compare the reported configurations of each method and are not budget-matched comparisons; the BERT-Small experiment below provides the matched comparison. For the second experiment, we utilize BERT-large (24-layers) as the teacher, BERT-Small (4-layers) as the student and BERT-base (12-layers) as the teacher-assistant for TAKD. We train the teacher for 30 epochs. For the student, this experiment uses the same configuration as the BERT-Small experiment of Chapter 4 rather than that of the DistilRoBERTa experiment above. We report the results in Tables 5.5 and 5.6. Here, Continuation-KD also outperforms other baselines and gets better overall performance than its competitors. Unlike the DistilRoBERTa tables, every row of Tables 5.5 and 5.6 was rerun for this chapter under the configuration stated above, so all methods there share the same student, teacher, schedule, and training budget. The BERT-Small numbers consequently differ from the corresponding entries in Chapter 3, which used a different maximum annealing index and a two-stage epoch schedule. They differ from the corresponding entries in Chapter 4 only in the small residual sense: that chapter's

BERT-Small experiment shares this configuration, and the remaining discrepancies reflect run-to-run variation across differing software and hardware environments rather than any difference in setup. All presented results in our experiments show that continuation-KD performs better than TAKD and Annealing-KD, which indicates the effectiveness of this method.

Table 5.3: DistilRoBERTa results for Continuation-KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	score
Teacher	68.1	86.3	91.9	92.3	96.4	94.6	91.5	90.22	88.91
Finetune	59.3	67.9	88.6	88.5	92.5	90.8	90.9	84	82.81
Vanilla-KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18	83.86
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	**91.7**	83.89	83.91
Annealing KD	61.67	73.64	90.6	89.01	93.11	91.64	91.5	**85.34**	84.56
Continuation-KD	**63.75**	**77.62**	**91.93**	**89.71**	**93.58**	**91.76**	91.64	85.16	**85.64**

Table 5.4: Performance of DistilRoBERTa trained by Continuation-KD on the GLUE leaderboard (test set) compared with Vanilla-KD, TAKD, and Annealing-KD. Matthews correlation for CoLA, Pearson correlation for STS-B, F1 for MRPC, matched accuracy for MNLI, accuracy for the remaining tasks; Score is the unweighted mean of the eight task metrics. The last three task columns are corrected from Table 4 of the published EMNLP 2022 Findings paper, where they appear transposed relative to the dev-set table and to the Chapter 3 leaderboard results.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	score
Finetune	52.97	**74.56**	87.93	84.9	93.1	90.57	88.77	83.4	82.02
Vanilla-KD	54.3	74.1	86	85.7	93.1	90.8	89.5	83.6	82.14
TAKD	53.2	74.2	86.7	85.6	93.2	91	89.4	83.8	82.14
Annealing KD	54	73.7	88	**87.0**	93.6	90.8	89.7	83.8	82.58
Continuation-KD	**54.5**	74.2	**90**	**87.0**	**93.8**	**91.6**	**90.1**	**84.7**	**83.24**

Table 5.5: BERT-Small results for Continuation-KD on dev set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	score
Teacher	65.8	71.48	89.38	89.2	92.77	92.82	91.45	86.3	84.90
Finetune	41.7	64.98	83.75	87.41	88.3	86.49	88.43	78.42	77.44
Vanilla-KD	**41.89**	64.98	86	85.95	88.76	86.75	88.24	78.62	77.65
TAKD	40.2	**65.7**	85.23	86.44	88.88	86.78	88.4	78.78	77.55
Annealing-KD	41.36	64.9	87.93	87.04	89.56	86.99	88.58	78.66	78.13
Continuation-KD	41.51	65.26	**89.44**	**87.81**	**90.21**	**87.52**	**90.61**	**79.27**	**78.95**

Table 5.6: BERT-Small results for Continuation-KD on test set. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the displayed task metrics and is not comparable across tables with different task sets.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	score
Finetune	38.1	61.8	83.4	78.8	89.7	86.4	78.1	77.6	74.24
Vanilla-kd	37.3	63.4	80.6	78.2	90.2	86.5	78.3	78.3	74.10
TAKD	**38.5**	62.3	80.5	79.3	89.7	86.7	78	78.2	74.15
Annealing-KD	38.3	63.3	81.9	80.6	89.8	86.8	78.4	79.3	74.80
Continuation-KD	**38.5**	**64.8**	**84.6**	**82.5**	**90.6**	**87.2**	**80.1**	**79.5**	**75.98**

5.7 Analysis

To investigate how Continuation-KD works, we did two ablation studies explaining different aspects of this technique: effects of the dynamically changing factors and noise mitigation.

Effects of Dynamic Factors Φ and ψ. In the first ablation, we scrutinize the effect of each dynamically changing component of the Continuation-KD loss function. It basically considers the effect of functions: $\psi(i)$ from Eq. 5.10, $\Phi(\mathcal{T}_i)$ when it anneals the outputs of the teacher in Eq. 5.11, and $\Phi(\mathcal{T}_i)$ when it anneals the margin of the hinge loss in Eq. 5.11. To investigate the effects of these components, we evaluate three one-factor-active configurations of the NLU experiments with distilRoBERTa model described in section 5.6.3 on MRPC and RTE datasets. In each configuration, we fixed two of the

components, and we only let one of them dynamically change during training. Table 5.7 reports the performance of each of these experiments. The first three columns of this table show the performance of distilRoBERTa on each of the datasets when only one of the three dynamical components changes and two others are fixed. The last column shows the performance of the model on these datasets in a regular training when all components dynamically change. In the first column, the $\Phi = 1$ is fixed for both teacher and margin. In the second column, the margin's coefficient is fixed to 1 and $\psi = 0.5$. Also, in the third column, the teacher's coefficient is fixed to 1 and $\psi = 0.5$. As shown in Table 5.7, with only one component dynamic the results fall in a narrow band on both datasets (MRPC ranges from 90.90 to 91.21 and RTE from 71.48 to 73.64), while the jointly scheduled configuration reaches 91.93 and 77.62. In these reported runs the joint schedule is therefore the best of the four configurations on both tasks. This design varies one factor at a time and does not identify the individual contribution of each component or establish that any one of them is necessary; separating those effects would require an all-fixed control and freeze-one-at-a-time conditions, preferably across several seeds.

Table 5.7: Performance of distilRoBERTa on MRPC and RTE datasets for different dynamically changing components of continuation-KD loss

dataset	ψ	Φ (teacher)	Φ (margin)	all
MRPC	91.09	91.21	90.90	91.93
RTE	72.56	73.64	71.48	77.62

Noise Robustness In the second ablation study, we visualize the effect of Continuation-KD with a noisy teacher. For this purpose, we consider a sinusoidal function with low frequency, and then we add to it a sinusoidal noise with high frequency (blue curves in Figure 5.2). Then, we take a fully connected network with one-dimensional input and output and two hidden layers with 128 neurons in each layer as a student model. We sample 3,000 points from the graph of the noisy sinusoidal function and train the student model once with Vanilla-KD and once with Continuation-KD (orange curves in Figure 5.2). As Figure 5.2 shows, in this run the model trained with Continuation-KD has a visibly smoother curve than the model trained with vanilla-KD, and tracks the underlying low-frequency function rather than the added noise. This is a single qualitative run on one synthetic function and is not a quantitative robustness measurement. This empirical result is consistent with the local-flatness mechanism described in Corollary 5.5: the margin-based plateau suppresses gradient updates from noise whose induced residual falls below

the annealed margin, allowing the student to learn the teacher's underlying low-frequency behavior.

5.8 Conclusion

In this chapter, we presented Continuation-KD, a unified KD framework that generalizes the Annealing-KD approach introduced in Chapter 3. Inspired by the continuation method from optimization theory, Continuation-KD starts with a smoothed version of the objective and gradually recovers the original one, eliminating the rigid two-stage structure of Annealing-KD.

The chapter's mechanism is the annealed hinge margin: Corollary 5.5 shows that below the annealed margin $\Phi\, m$ the continuation term's gradient and Hessian are identically zero, so residuals within the teacher's noise floor produce no distillation update, and a simple Gaussian noise model gives the energy scale $m = K\sigma^2$ as a starting point for the margin. Two exact identities make the mechanism precise: Theorem 5.1 gives, for any differentiable student, $\nabla_\theta \mathcal{L}_\mathcal{T} = \nabla_\theta \mathcal{L}_1 + (1-\Phi) J_\theta^T z_t$, and Theorem 5.3 the Hessian decomposition $\nabla^2 \mathcal{L}_\mathcal{T} = H_S - \Phi\, C$, each isolating the teacher's contribution in a single term scaled linearly by the annealing factor and identifying exactly what the hinge switches off. These are structural facts about the objective; we are explicit that the step from this structure to the optimization and generalization gains observed on GLUE and CIFAR is not established by controlled experiments and is left as future work.

Our technique does not require multiple training stages, and obtains the highest aggregate score among the evaluated baselines (Vanilla-KD, TAKD, and Annealing-KD) across both computer vision (CIFAR-10, CIFAR-100) and NLP tasks (GLUE benchmark) with different student architectures. Wall-clock efficiency was not measured. For this investigation, we implemented the continuation method by smoothing the objective with the hinge loss; however, it can potentially be realized differently. Investigating other realizations of continuation optimization for improving small models' performance is an interesting direction for future work.

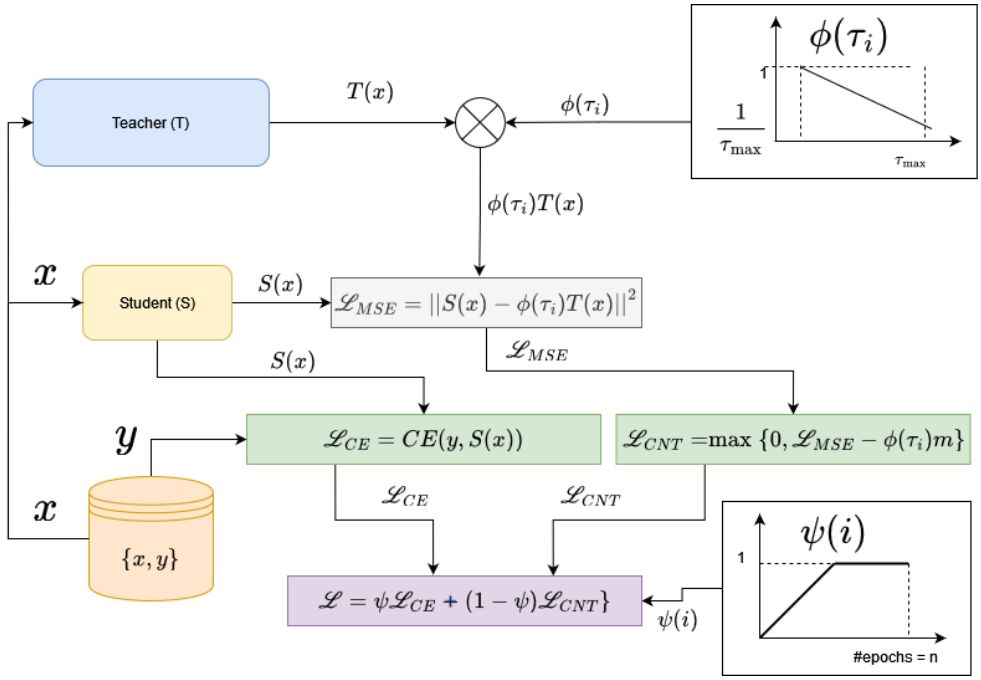
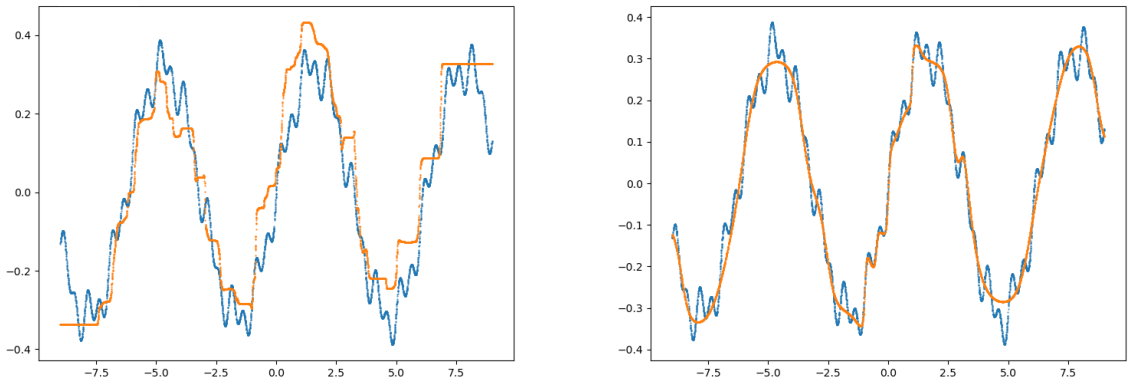


Figure 5.1: Principle diagram illustrating different components of Continuation-KD. The figure shows the three scheduled components: the teacher logits are scaled by $\Phi(\mathcal{T}_i)$, the hinge margin is scaled by the same factor, and $\psi(i)$ interpolates between the continuation loss and the supervised task loss. The diagram specifies the training objective; it does not establish that the continuation loss is easier to optimize, or has lower curvature, than the supervised objective. For legibility the diagram writes the annealing factor and its index as $\phi(\tau_i)$; these are the $\Phi(\mathcal{T}_i)$ and $\mathcal{T}_i$ of the text. The term labeled $\mathcal{L}_{CE}$ denotes the supervised task loss, which is cross-entropy for the classification tasks and the corresponding regression loss for STS-B.



(a) (b)

Figure 5.2: (a) Illustrates the behavior of the student model after training with a noisy teacher without using continuation-KD. (b) Illustrates the behavior of the student model after training with a noisy teacher by using continuation-KD for training. Blue curves are the samples from the noisy teacher. Orange curves are the samples of the trained student in each scenario. The figure is a qualitative illustration from a single run on one synthetic function; it reports no held-out error and does not by itself establish robustness across noise levels or random initializations, nor validate the Gaussian margin model of Section 5.5.2.

Chapter 6

Data Distillation: Efficient Citer

6.1 Overview

The preceding chapters (Chapters 3–5) addressed the capacity-gap problem through *logit-level* knowledge distillation, where the student directly matches the teacher's output distributions during training. This paradigm assumes white-box access to the teacher's internals, its logits, activations, and gradients. However, as discussed in Section 2.4.1, the most capable language models today are deployed behind proprietary APIs (e.g., GPT-4, ChatGPT), making white-box distillation infeasible.

This limitation has given rise to *data distillation* (Section 2.4.2), a black-box paradigm in which the proprietary teacher is used purely as a data synthesizer: carefully designed prompts elicit high-quality outputs from the teacher, and a smaller, open-weight student is then fine-tuned on this synthetic corpus via standard supervised learning. Pioneering work such as Alpaca [121] and Vicuna [24] demonstrated the effectiveness of this approach for general instruction-following capabilities.

In this chapter, we apply the data distillation paradigm to a specific and practically important task: *citation-grounded question answering*. We use ChatGPT as the teacher to synthesize a training corpus of answers enriched with inline citations, and fine-tune two compact student models, a 3B-parameter FID-style Flan-T5 (FIDCiter) and a 13B LLaMA model (LLaMACiter), on this distilled dataset. In our evaluation the resulting models exceed the ChatGPT teacher on human-rated informativeness, and match or exceed it on citation quality in domain, while using much shorter instruction templates; ChatGPT retains an edge on citation quality on the out-of-domain datasets. This demonstrates that

data distillation can transfer a substantial part of a specialized capability from a proprietary teacher to efficient, deployable students.

6.2 Introduction

The growing popularity of LLMs in information-seeking tasks is undeniable, thanks to their ability to generate fluent, realistic responses. However, there is a growing concern regarding the information accuracy of these responses, and the ability to verify them. Moreover, information is a temporal, ever-changing concept and therefore a model's internal knowledge can quickly become outdated. One possible way to address these concerns, which has gathered a heightened interest, is retrieval-based LLMs, which incorporate external knowledge during both the training and inference stages. However, factual verification of model responses still remains a challenge.

An effective approach to facilitate factual verification involves equipping LLMs with the ability to cite external information. Several commercial systems, including Bing Chat[1], you.com[2], and perplexity[3], have already implemented this approach by leveraging web-based queries to find relevant information, and utilize it to answer specific questions with the relevant citations. However, details of these models are not publicly available.

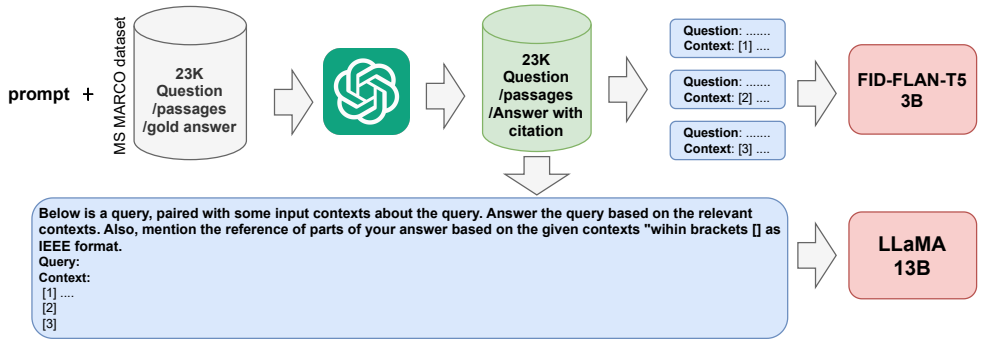


Figure 6.1: Our data distillation pipeline: starting from 23K candidate MS MARCO examples, ChatGPT (teacher) generates citation-enriched answers; after filtering and citation-format cleaning, approximately 18K examples are used to fine-tune the FIDCiter and LLaMACiter student models.

[1]https://www.bing.com/new?scdexwlcs=1
[2]https://you.com/
[3]https://perplexity.ai/

Some recent works [95, 88] have attempted to enable LLMs to cite the provided contexts in their response. However, these works fine-tune models which consist of hundreds of billions of parameters (175 billion and 280 billion respectively) and support a range of functions. In contrast, our objective is to develop an efficient answer composition module that can provide informative answers with correct citations, independent of the passage retrieval module.

Recently, [47] employed in-context learning (ICL), using instructions and demonstrations to facilitate the models' ability to cite relevant context. They applied this approach to various LLMs, including LLaMA [124], Vicuna (instruction-tuned models) [24], and ChatGPT (a closed-source model) [96]. While ChatGPT is able to provide relatively high-quality answers with relevant citations, for models like LLaMA 13B and Vicuna providing demonstrations alone proved insufficient. Moreover, the use of long demonstrations in ICL increases the prompt length by thousands of tokens, thereby making inference extremely inefficient.

The success of data distillation for general instruction-following [121, 35, 99] suggests an alternative route: rather than relying on expensive in-context demonstrations at inference time, one can distill the teacher's capability into a compact student through supervised fine-tuning on a small, high-quality synthetic dataset. Building upon this idea, we apply the data distillation paradigm to citation-grounded QA. We use ChatGPT as the teacher to synthesize a citation-enriched training corpus, and fine-tune two efficient student models on this distilled dataset. Specifically, our work makes the following contributions:

- We construct a citation dataset for supervised learning building upon MS MARCO [6], an open-book QA dataset, by prompting ChatGPT, serving as the teacher in our data distillation pipeline, to incorporate citations within the gold responses.
- We fine-tune two compact student models on this distilled dataset: a 3B-parameter FID-style Flan-T5 [27] model with Fusion-in-Decoder (FID) [62], named FIDCiter, and a 13B LLaMA model, named LLaMACiter. Both student models demonstrate strong performance against the ChatGPT teacher and other baselines on both in-domain and out-domain datasets using both human and automatic evaluation.

6.3 Related work

WebGPT [95], is trained on humans demonstrations on how to use the web environment by issuing commands and answering questions. These demonstrations are used to fine-tune a

pretrained GPT-3 [13] model. Subsequently, a reward model is employed for reinforcement learning with human feedback (RLHF), which further improves the model's performance.

In contrast, GopherCite [88] focuses on reading comprehension by utilizing a large context that includes thousands of tokens from multiple pages. GopherCite also utilizes RLHF to fine-tune the model based on human preferences. Both approaches highlight the significance of larger model sizes, with WebGPT using a 175 billion parameter model and GopherCite employing a 280 billion parameter model.

Attributed question answering [12] involves answering a question while simultaneously providing a reference to a brief segment of text that supports the answer. Unlike our proposed methods where the link to the reference accompanies the answer directly, in this approach, the model generates the answer along with the reference link as an attribute.

Table 6.1: Human evaluation of model outputs on MS MARCO, MIRACL, ELI5, and ASQA datasets. For MS MARCO and MIRACL, 50 randomly chosen samples are assessed, while 100 randomly selected samples are evaluated for ELI5 and ASQA datasets. Each sample comprises five passages. Criteria evaluated include Fluency (FL), Informativeness (INF), and Citation Quality (CQ).

Model	**Prompt Length**	**MS MARCO**			**MIRACL**			**ELI5**			**ASQA**		
	(words)	**FL**	**INF**	**CQ**	**FL**	**INF**	**CQ**	**FL**	**INF**	**CQ**	**FL**	**INF**	**CQ**
ChatGPT	1776	**1**	0.76	0.86	**1**	0.56	**0.87**	**0.99**	0.99	**0.88**	**0.97**	0.99	**0.91**
FIDCiter (3B)	10 (2/encoder)	**1**	**0.97**	**0.89**	0.99	0.86	0.76	**0.99**	0.95	0.81	0.86	0.94	0.78
LLaMACiter (13B)	61	0.98	0.95	0.72	0.97	**0.90**	0.79	**0.99**	**1**	0.81	0.89	**1**	0.80

6.4 Training

6.4.1 Data Distillation Pipeline

The core of our approach follows the data distillation paradigm described in Section 2.4.2: we treat ChatGPT as a black-box teacher and use it to synthesize a training corpus of citation-enriched answers, which we then use to fine-tune compact student models via standard supervised learning.

Source Data. The Microsoft Machine Reading Comprehension (MS MARCO) dataset is a comprehensive collection of data designed for machine reading comprehension, question answering, and passage ranking [6]. The dataset consists of about 1M instances of web

search queries accompanied by human-generated correct answers, as well as sets of positive and negative passages, sourced from the Bing search engine. The provided answers do not have references to the associated passages, and a significant number of samples contain only one positive passage.

Teacher-Generated Synthetic Targets. We select 23K candidate samples from the MS MARCO dataset that have at least two relevant contexts. Each sample contains a query, relevant and irrelevant contexts, as well as the gold answer without citation. We leverage ChatGPT as the teacher model to generate answers that cite the factual information from relevant contexts. To achieve this, for each sample, we provide the teacher with a query, a list of relevant contexts (between 2-5), and a gold answer. We then prompt the model to use the gold answer and the provided relevant context to generate a new answer that includes citations in IEEE format. Finally, we add irrelevant contexts to the positive contexts to make the number of contexts equal to 5. This way each query in training data has five contexts from which 2-5 are relevant and the rest are irrelevant. This enforces the student model to learn to ignore irrelevant contexts (due to imperfections in IR systems) and only use the positive ones for inference. We then cleaned the teacher's outputs by removing any hallucinated references and standardizing the citation format (see Appendix D.3). After this filtering and citation-format cleaning, approximately 18K of the 23K candidates remain, and this is the final distilled dataset used for training.

It is important to be precise about the teacher's role here. ChatGPT is used as a *gold-conditioned citation rewriter*: it receives the query, the retrieved passages, *and* the human gold answer, and produces a citation-enriched rewriting of that answer. It is therefore not answering the training questions independently, and the supervision the students receive combines teacher output with human answer supervision. The setting is best described as data distillation from a gold-conditioned rewriter rather than as distillation from an independently answering teacher. This also affects the comparison reported in Section 6.6: the ChatGPT baseline evaluated there is prompted *without* the gold answer, so the teacher and the students are not evaluated under identical inputs, and the resulting numbers should not be read as a controlled measurement of student-versus-teacher question-answering capability.

6.4.2 Student Models

We fine-tune two student architectures on the distilled dataset described above.

For our 3B student, we adopt an FID-style architecture [62] that enables independent passage processing within the encoder while ensuring collective aggregation throughout the decoder. For a fixed passage length, encoding k passages independently makes the encoder cost grow linearly in k, rather than applying self-attention to one concatenated k-passage sequence. Self-attention remains quadratic in the length of each individual passage, and decoder cross-attention grows with the total number of encoded tokens. As the backbone encoder-decoder model we use Flan-T5 [27] with strong instruction-following abilities. We refer to this student as FIDCiter.

Furthermore, in line with the success and popularity of LLaMA for instruction following, we also fine-tune a 13B LLaMA student referred to as LLaMACiter. See Appendix D.1 for detailed hyperparameters used during training.

Table 6.2: Comparative Evaluation on ELI5 and ASQA using ALCE Benchmark with 1000 samples for 3 and 5 passages respectively. Numbers for ChatGPT and Vicuna models are taken from [47].

		ELI5				**ASQA**			
Passages	**Model**	**Prompt Length**	**Correctness**	**Citation Quality**		**Prompt Length**	**Correctness**	**Citation Quality**	
		(words)	(Claim)	Precision	Recall	(words)	(EM)	Precision	Recall
5	ChatGPT	2668	12.0	**51.1**	**50.0**	2550	40.4	**72.5**	**73.6**
	FIDCiter (3B)	10 (2/encoder)	14.3	44.1	44.8	10 (2/encoder)	39.4	62.7	64.2
	LLaMACiter (13B)	61	**16.1**	45.7	34.6	61	**41.8**	56.1	55.2
3	Vicuna (13B)	2668	10.0	15.6	19.6	2550	31.9	51.1	50.1
	FIDCiter (3B)	6 (2/encoder)	15.1	**53.0**	**46.0**	**6** (2/encoder)	40.0	**61.6**	**60.3**
	LLaMACiter (13B)	61	**16.8**	44.6	34.8	61	**41.7**	60.1	60.2

6.5 Evaluation Criteria

6.5.1 Human evaluation

Due to the lack of well-studied benchmarks for this task, our main metric in this paper is human evaluation. For the MS Marco dataset [6], we select 50 held-out samples to serve as our test set. Additionally, to ensure diversity and enable out-of-scope evaluation, we also perform human evaluation on a subset of randomely selected 50 samples from MIRACL dataset [153] and 100 samples from ELI5 [41] and ASQA [114] datasets. During the evaluation process, the LM receives queries and passages along with a prompt and generates answers while citing relevant passages.

After anonymization, the resulting queries, passages, and generated answers are provided to specialist annotators for evaluation. It is important to note that datasets like

MIRACL do not provide gold answers. As a result to ensure reproducibility and fair comparison, we do not provide gold answers to the annotators for any of the samples. Three specialist annotators with over one year of experience in data annotation were hired for the evaluation process. They received task-specific training to ensure consistency and reduce bias, and were blind to model identity (outputs were anonymized before annotation). The annotators were compensated at an hourly rate of 25 CAD. During the evaluation, each answer was divided into segments separated by citations, and the metrics were calculated individually for each segment and averaged for each sample. Inter-annotator agreement was not computed for this study, so the human-evaluation scores should be read as descriptive point estimates rather than measurements with a quantified reliability. The following evaluation metrics were used to assess our test sets:

- **Informativeness**: The metric evaluates the extent to which a generated answer answers the question. For each segment, if it responds at least partially to the query, we consider it informative and assign a score of 1; otherwise, it is given a score of 0.
- **Fluency**: The metric assesses the naturalness and linguistic correctness of the generated segments. If the segment contains no typographical, morpho-syntactic, or lexical errors, it receives a score of 1, otherwise 0.
- **Citation Quality (CQ)**: This metric measures whether factual claims in the generated segment can be supported by corresponding quotes. If a cited quote supports the information appearing in the segment it is considered correct. The final score, between 0 and 1, is calculated by dividing the number of correctly cited passages by the total number of passages cited in the segment. If the segment does not cite any passage, it is scored as 0. This is the criterion labeled CQ in Table 6.1.

6.6 Results

Table 6.1 presents the results of our human evaluation on 50 held-out samples from MS MARCO. As noted in Section 6.4.1, the ChatGPT row is a baseline prompted without the gold answer, whereas the training targets were produced by a gold-conditioned ChatGPT; the comparison therefore contrasts the students against a differently-conditioned teacher rather than against the model that supervised them. In the case of ChatGPT, the prompt was designed by the ALCE benchmark [47] with four demonstrations. We removed extra information like the title of the URL from the prompt to make it more efficient. The results demonstrate that all models achieve almost ideal fluency. However, when it comes to informativeness, both FIDCiter and LLaMACiter outperform ChatGPT.

Regarding supportedness, our FIDCiter model provides the best citation quality, followed by ChatGPT. To assess the generalization of our model to new datasets, we further evaluate its performance on MIRACL, ELI5, and ASQA datasets, utilizing a distinct pipeline for extracting relevant passages. The picture here is mixed. LLaMACiter has a higher informativeness score than ChatGPT on all three out-of-domain sets, whereas FIDCiter is higher on MIRACL (0.86 against 0.56) but lower on ELI5 (0.95 against 0.99) and ASQA (0.94 against 0.99). Where our models do lead, this is at least partly because our informativeness metric penalizes answers that include unnecessary irrelevant contexts. ChatGPT in particular has a tendency to provide such chatty answers (See Appendix D.5). As for citation quality, ChatGPT surpasses both FIDCiter and LLaMACiter. Overall, we observe that the drop in both informativeness and citation quality scores when generalizing to other datasets is less pronounced for LLaMACiter compared to FIDCiter. This pattern is consistent with stronger transfer in the 13B configuration, but model size is confounded with architecture, tokenizer, initialization, and training procedure, so it does not isolate scale as the cause. More importantly, our models use very short prompts compared to ChatGPT, making them far more efficient candidates for this task. To further evaluate the model's performance on out-of-domain samples, we utilize correctness, citation precision, and citation recall metrics on 1000 randomly selected samples from the ELI5 and ASQA datasets provided by ALCE benchmark. The results are presented in Table 6.2. Following the approach of [47], we compare the performance of FIDCiter with ChatGPT, LLaMA 13B, and Vicuna 13B models. The prompt used for these baselines is the same as the one used for ChatGPT, which includes four demonstrations from the ELI5 dataset. However, due to the extensive prompt length, Vicuna is limited to only three passages. With three passages, the Vicuna model shows very low correctness and citation quality, whereas our FIDCiter model performs better. FIDCiter's instruction template is about 444× shorter than the four-demonstration in-context prompt used for the baselines; because Table 6.2 counts only the instruction template (not the passage text) and does not report latency, memory, or FLOPs, this figure quantifies template-length reduction rather than end-to-end computational cost. Independently of the template, FIDCiter's Fusion-in-Decoder encoder processes the passages separately, so its encoder cost grows linearly rather than quadratically in the number of passages. When comparing the performance of our models with ChatGPT using five passages, our LLaMACiter model achieves the highest correctness score among all models.

6.7 Conclusion

In this chapter, we demonstrated the effectiveness of data distillation (Section 2.4.2) for transferring a specialized capability, citation-grounded question answering, from a proprietary teacher (ChatGPT) to compact, deployable student models. Unlike the logit-level distillation methods studied in Chapters 3–5, where the student directly matches the teacher's output distributions, the data distillation paradigm employed here treats the teacher as a black-box data synthesizer: we prompted ChatGPT to generate citation-enriched answers and fine-tuned two student models, FIDCiter (3B) and LLaMACiter (13B), on this distilled corpus via standard supervised learning.

The evaluation, conducted through human assessment and the ALCE benchmark, shows that the relative ranking is metric-dependent. LLaMACiter achieves higher human-rated informativeness than ChatGPT on all four evaluated datasets; FIDCiter does so on MS MARCO and MIRACL but not on ELI5 or ASQA. FIDCiter has the highest citation-quality score on the in-domain MS MARCO set, while ChatGPT has the highest citation-quality score on the three out-of-domain human-evaluation sets (MIRACL, ELI5, ASQA). On the ALCE five-passage setting, LLaMACiter obtains the highest correctness score, whereas ChatGPT retains higher citation precision and recall. The students use much shorter instruction templates throughout. Particularly, the 3B FIDCiter, despite its smaller size, performs competitively with our 13B LLaMACiter, indicating that data distillation can transfer complex capabilities even to very compact students.

While data distillation provides a practical and effective mechanism for black-box knowledge transfer, it relies on a fixed, teacher-generated training corpus. The student is trained only on examples that the teacher happened to produce, with no mechanism to target regions where the student's behavior diverges most from the teacher's. In the next chapter, we address this limitation by proposing Maximum Divergence Knowledge Distillation (MDKD), which explicitly seeks out high teacher–student divergence during data construction to improve the efficiency and coverage of the distillation process.

Chapter 7

Maximum Divergence Knowledge Distillation

7.1 Overview

The previous chapter demonstrated the effectiveness of data distillation (Chapter 6), where a proprietary teacher (ChatGPT) is used as a black-box data synthesizer to generate a training corpus for compact student models. While data distillation successfully transfers specialized capabilities, it suffers from a fundamental limitation: the student is trained on a *fixed* corpus of teacher-generated outputs, with no mechanism to target regions where the student's behavior diverges most from the teacher's.

This limitation connects to a broader issue in knowledge distillation for autoregressive language models: *exposure bias* (Section 2.4.3). A student trained on teacher-generated sequences may behave very differently when generating its own sequences at inference time, because it never encounters its own errors during training. As discussed in Section 2.5, addressing this problem requires an *active* distillation paradigm that explicitly targets teacher–student disagreement, rather than passively training on a fixed dataset.

In continuous domains such as computer vision, our prior work on Backward KD (Section 2.5.2) showed that actively maximizing teacher–student divergence through gradient-based input perturbation yields superior student models [66]. However, this approach relies on differentiable inputs and cannot be directly applied to discrete token sequences. In this chapter, we propose *Maximum Divergence Knowledge Distillation* (MDKD), which translates the active-divergence philosophy of Backward KD into the discrete, autoregressive

setting. At each generation step, MDKD uses a rejection-sampling mechanism to preferentially select tokens where the student underestimates the teacher, constructing training sequences concentrated in regions where the student underweights teacher-supported tokens.

Because MDKD builds its training sequences from teacher proposals rather than from the student's own rollouts, it targets the *distributional-coverage* aspect of this problem, supervising the student on teacher-supported trajectories it currently underweights, rather than the *on-policy* aspect of exposing the student to its own inference-time errors. Combining MDKD with student-generated prefixes for more direct exposure-bias mitigation is a natural extension, discussed in Chapter 8.

Unlike the proof-of-concept formulation, MDKD now rests on an exact characterization of its stochastic acceptance branch (Theorem 7.1): that step samples precisely from the teacher's *uncovered mass* $[T-S]_+$, with acceptance probability equal to the teacher–student total variation distance, and Corollary 7.3 places it inside a mixture decomposition of the full sampler. This puts MDKD's sampler on the same footing as the entropy-control result of Chapter 3 and the structural analysis of the annealed objective in Chapter 5, and reveals MDKD as the structural complement of speculative decoding along a shared overlap–residual partition of teacher sampling.

7.2 Introduction

As established in Section 2.1, classical KD aligns teacher and student distributions on a fixed set of training examples. For autoregressive language models, this means the student is trained on a fixed corpus of teacher-generated (or human-written) sequences and receives no incentive to match the teacher in unexplored regions of the output space. The student may therefore assign very different probability mass to tokens it would generate at inference time, the exposure bias problem formalized in Section 2.4.3.

Several recent methods have attempted to address this distribution mismatch. Generalized Knowledge Distillation (GKD) trains the student on its own self-generated sequences while using the teacher for token-level supervision [1]. Speculative KD (SKD) interleaves student proposals with teacher corrections: the student proposes tokens and the teacher replaces poorly ranked ones, thereby constructing high-quality sequences for distillation [140]. While these methods address distribution mismatch or sample difficulty, none explicitly seeks training examples where the student and teacher diverge most.

The distribution-aligned sequence distillation (DASD) framework recently argued that

supervised fine-tuning on teacher outputs fails to represent the teacher's sequence-level distribution and introduces exposure bias [142]. DASD remedies this by temperature-scheduled sampling and a sequence-level divergence-aware filter: it generates multiple candidate responses from the teacher and retains those in which the teacher is confident and the student is not [142]. Although this provides the important insight that teacher-high/student-low regions are more informative, it operates at the level of entire responses, not individual tokens, and does not steer generation trajectories toward divergence.

We propose *Maximum Divergence Knowledge Distillation* (MDKD), a simple yet powerful framework for generative KD that explicitly seeks high teacher–student divergence at the *token level*. At each autoregressive step, MDKD draws m candidate tokens from the teacher's nucleus-filtered distribution and accepts each with probability $[\,1 - S(v)/T(v)\,]_+$; if no candidate is stochastically accepted, a deterministic fallback selects the candidate with the largest relative shortfall among the sampled candidates. In effect, MDKD constructs training sequences in regions where the teacher and student disagree most. Unlike GKD, we do not rely on the student's own policy to explore; unlike SKD, we do not merely correct student proposals; unlike DASD, we operate at the token level and can direct generation into high-divergence trajectories. Our method is simple to implement, requires no auxiliary models or reinforcement learning, and is orthogonal to the choice of divergence in the loss.

Contributions. This chapter makes three contributions.

1. A **unified active-distillation principle** that connects gradient-based input perturbation in continuous domains (Backward KD, Section 2.5.2) with token-level rejection sampling in discrete domains, giving a single account of how to construct KD training data actively across modalities.

2. **MDKD**, an auxiliary-model-free rejection-sampling rule with an exact characterization of its stochastic acceptance branch and a mixture decomposition of the full finite-candidate sampler. Theorem 7.1 shows that the stochastic step samples exactly from $[T - S]_+$ with acceptance probability equal to the teacher–student total variation distance; Corollary 7.3 expresses the full selected-token distribution as a mixture of this exact-residual branch and a deterministic fallback, with the weight between them a direct readout of the per-step teacher–student divergence; a fallback-policy ablation in Section 7.5.5 shows that the stochastic branch is what produces the method's gain. This reveals MDKD as the structural complement of speculative decoding along the partition $T = \min(T, S) + [T - S]_+$.

3. **Comprehensive experiments** across arithmetic reasoning (GSM8K), dialogue summarization (DialogSum), and code generation (MBPP→HumanEval), on two architecturally distinct teacher–student pairs. With a Qwen2.5-14B-Instruct teacher and only 1,000 KD samples, MDKD lifts a base Qwen2.5-1.5B student from 8.26% to 64.90% on GSM8K, +10.3 percentage points over the next-best baseline and recovering 65% of the teacher–student gap.

7.3 Related Work

7.3.1 Divergence-aware distillation in continuous domains

The philosophy of actively targeting teacher–student disagreement was first explored in continuous domains. As discussed in Section 2.5.2, Backward KD [66] proposes an iterative backward-pass technique: given a batch of training samples, the gradient of the divergence between teacher and student is computed with respect to the inputs, and the inputs are perturbed in that direction (Eq. 2.9). These high-divergence samples highlight regions where the student's predictions diverge most strongly from the teacher's and are used to augment the training set. This approach improves distillation in computer vision and NLP classification tasks but relies on differentiability of the input, making it inapplicable to discrete token sequences.

7.3.2 Generalized knowledge distillation

Generative language models produce sequences rather than single labels, and thus the distribution mismatch between training and inference is a major obstacle for KD. Generalized Knowledge Distillation (GKD) frames distillation for autoregressive models as an imitation-learning problem [108]: instead of training on a fixed set of teacher outputs, GKD trains the student on its own self-generated sequences while using the teacher to provide token-level probabilities as supervision [1]. This "on-policy" approach mitigates the mismatch between training data and inference distributions, and it allows alternative divergence measures, such as reverse KL or Jensen–Shannon divergence, when the student is less expressive than the teacher. MiniLLM [51] takes a related approach, optimizing reverse KL on student-generated sequences to mitigate mode-averaging. These methods do not, however, explicitly target high-divergence regions: the student explores its own distribution, which may overlook tokens where the teacher–student gap is largest.

7.3.3 Speculative knowledge distillation

Building on the insight that both fixed-dataset KD and on-policy KD suffer from complementary weaknesses, Speculative Knowledge Distillation (SKD) proposes an *interleaved sampling* mechanism. In SKD, the student model proposes a token based on its own distribution, and the teacher examines whether the token lies within its top-K candidates; if not, the token is replaced by a sample from the teacher's distribution [140]. This procedure yields high-quality training sequences that align with the student's inference-time distribution while avoiding the low-quality samples that can plague on-policy training. SKD adapts speculative decoding [79], which targets the teacher–student overlap $\min(T, S)$ to bypass teacher computation; SKD inherits this overlap-based acceptance rule and uses it to verify student-proposed sequences. As we show in Section 7.4.4, MDKD instead targets the complementary residual $[T - S]_+$, making it the learning-side structural dual of speculative decoding.

7.3.4 Divergence-aware sequence distillation

Recognizing that standard sequence-level distillation fails to represent the teacher's full distribution, Yan *et al.* introduce Distribution-Aligned Sequence Distillation (DASD) [142]. DASD argues that supervised fine-tuning on teacher outputs misrepresents the teacher distribution, misaligns with the student's capacity, and introduces exposure bias (Section 2.4.3). To address these issues, DASD uses temperature-scheduled sampling from the teacher and a sequence-level divergence-aware filtering step: for each prompt, multiple candidate responses are generated at different temperatures, and the distribution of teacher and student probabilities across these responses is analyzed to identify patterns. Responses where the teacher is confident and the student is not are retained, while low-divergence responses are discarded. DASD also incorporates a mixed-policy stage where partial student generations are completed by the teacher to reduce exposure bias. Although DASD highlights the importance of teacher-high/student-low regions, its divergence criterion operates at the granularity of entire sequences rather than individual tokens, and the divergence is computed only after generation; there is no mechanism to steer generation towards high-divergence tokens during sampling.

7.3.5 Active, hard-example, and adversarial distillation

A broader line of work seeks training data that exposes student weaknesses. Hard-example mining and curriculum learning [8] re-weight existing examples by difficulty; data-free and

synthetic-data KD [89, 42] generate inputs that maximize teacher–student disagreement, typically via a learned generator or adversarial optimization [133]. These methods share MDKD's premise, that the student should be trained on its weakest regions, but rely on auxiliary models, generator networks, or differentiable inputs. MDKD achieves the same goal through a closed-form, auxiliary-model-free rejection-sampling rule applied directly to the teacher's own distribution; the teacher and student models are of course still required.

7.3.6 Discussion

These methods highlight a common theme: effective distillation for generative models requires attention to the distribution mismatch between training and inference and to regions where the student's predictions diverge from the teacher's. Backward KD generates adversarial inputs to expose divergence, GKD trains on student-generated sequences to match the inference distribution, SKD interleaves student and teacher sampling to filter out implausible student tokens, and DASD selects teacher responses where the student's probability is low. However, none of these techniques simultaneously addresses discrete input domains and explicitly maximizes teacher–student divergence at the token level during data construction. In the next section, we introduce MDKD, which fills this gap by using a simple rejection-sampling rule to prioritize teacher tokens that the student underestimates at each generation step.

7.4 Methodology

We formalize Maximum Divergence Knowledge Distillation (MDKD) for autoregressive language models. MDKD seeks to focus student training on the regions of the output distribution where the teacher and student disagree most.

7.4.1 Preliminaries

Let $\mathcal{V}$ denote the vocabulary and let $\mathcal{V}^*$ denote the set of all finite token sequences. For a prefix $x_{\le i} = (x_1, \dots, x_i)$, let $T(\cdot \mid x_{\le i})$ denote the teacher's conditional distribution over the next token and let $S(\cdot \mid x_{\le i}; \theta)$ denote the student's conditional distribution parameterized by θ. In classical KD, one minimizes

$$\mathcal{L}_{\mathrm{KD}}(\theta) \;=\; \mathbb{E}_{x \sim \mathcal{D}}\big[\mathrm{KL}\big(T(\cdot \mid x) \,\|\, S(\cdot \mid x; \theta)\big)\big], \tag{7.1}$$

where $\mathcal{D}$ is a fixed dataset. This objective only aligns teacher and student on the observed data. MDKD instead constructs a distribution μ over sequences that concentrates on areas where T and S diverge and trains the student on samples from μ.

7.4.2 The divergence-maximization principle

The core insight behind MDKD is that training data should be concentrated where the student most underestimates the teacher. In continuous domains, this principle is realized by gradient-based input perturbation: given a sample x, one computes $\nabla_x D(T, S)$ and perturbs x in that direction to produce a harder training example (Backward KD, Section 2.5.2). In discrete autoregressive settings, inputs are non-differentiable token sequences, so gradient-based perturbation does not apply. MDKD instead realizes the same principle through a rejection-sampling rule applied *during* sequence generation, described next. The two instantiations, continuous and discrete, are thus the same active-distillation principle expressed in two domains.

7.4.3 Divergence-aware token sampling

Given a prompt $x_{1:t}$, MDKD generates a continuation one token at a time by sampling candidate tokens from the teacher's distribution and retaining those that the student underestimates. At position i ($i \geq t+1$), the teacher logits are scaled by a temperature τ and restricted to a nucleus of cumulative probability p_{nuc}, yielding a filtered distribution $T_\tau(\cdot \mid x_{<i})$ supported on the nucleus; m candidate tokens are then drawn i.i.d. from this filtered distribution. For each candidate $\hat{x}_j$, the acceptance score is computed as

$$\alpha_j = \max\Big(0,\ 1 - \frac{S_\tau(\hat{x}_j \mid x_{<i})}{T_\tau(\hat{x}_j \mid x_{<i})}\Big), \tag{7.2}$$

where T_τ is the temperature-scaled, nucleus-filtered teacher and S_τ is the student distribution *renormalized over the nucleus support of* T_τ, so that T_τ and S_τ are both probability distributions on the same set. Since $\alpha_j = [\,T_\tau(\hat{x}_j) - S_\tau(\hat{x}_j)\,]_+ / T_\tau(\hat{x}_j)$, it measures the teacher's *relative* undercoverage at that token rather than the absolute residual mass; maximizing it therefore selects the largest relative shortfall among the sampled candidates, not necessarily the largest contribution to the total variation distance. The candidates are considered in turn: for each candidate j an independent $r_j \sim \text{Uniform}(0, 1)$ is drawn, and the first candidate with $r_j < \alpha_j$ is appended to the sequence. If no candidate is accepted via this stochastic step, the candidate with the *highest* α_j is selected as a fallback, ensuring

that even in the worst case MDKD chooses the candidate the student most underweights relative to the teacher. Algorithm 3 details the procedure.

The rejection step accepts tokens where the student assigns much less probability than the teacher (α_j is large), biasing the generation towards high-divergence regions. When the two distributions agree ($S_\tau \approx T_\tau$), $\alpha_j \approx 0$ and the candidate is almost surely rejected. The arg max fallback ensures that even when no candidate passes the stochastic threshold, the selected token is still the one where the student underestimates the teacher the most. By construction every accepted token lies in the teacher's nucleus, so the resulting sequences are teacher-supported at every position. Temperature τ and nucleus threshold p_{nuc} control the sharpness and breadth of the candidate pool; their effect on downstream performance is analyzed in Section 7.5.8.

7.4.4 Theoretical analysis

The stochastic acceptance step of the previous subsection admits an exact distributional characterization, and the full finite-candidate procedure a mixture decomposition. Throughout this subsection, T and S refer to T_τ and S_τ as defined in Eq. (7.2), treated as two probability distributions over their common nucleus support $\mathcal{V}_0 \subseteq \mathcal{V}$ with $T(v) > 0$ for all $v \in \mathcal{V}_0$. We first characterize the stochastic acceptance step in isolation (Theorem 7.1), then the full selected-token distribution under the finite-candidate algorithm with deterministic fallback (Corollary 7.3).

Theorem 7.1 (The stochastic MDKD step samples the teacher's uncovered mass)**.** *Fix a prefix $x_{<i}$ and let T and S be teacher and student next-token distributions over a common support $\mathcal{V}_0$ with $T(v) > 0$ for all $v \in \mathcal{V}_0$. Consider a single proposal $v \sim T$ accepted with probability $\alpha(v) = [\,1 - S(v)/T(v)\,]_+$. Then:*

1. *The probability of stochastic acceptance equals the teacher–student total variation distance:*
$$\Pr[\text{accept}] = \sum_{v \in \mathcal{V}_0} [\,T(v) - S(v)\,]_+ = \text{TV}(T, S). \tag{7.3}$$

2. *Whenever $\text{TV}(T, S) > 0$, conditioned on acceptance the selected token is distributed as the normalized positive residual:*
$$\Pr[v \mid \text{accept}] = \frac{[\,T(v) - S(v)\,]_+}{\text{TV}(T, S)}. \tag{7.4}$$

3. *Whenever* $\mathrm{TV}(T,S) < 1$*, conditioned on rejection the proposal is distributed as the normalized teacher–student overlap:*

$$\Pr[v \mid \text{reject}] = \frac{\min\{T(v), S(v)\}}{1 - \mathrm{TV}(T,S)}. \tag{7.5}$$

The stochastic MDKD step thus exactly decomposes teacher sampling into an agreement component $\min\{T,S\}$ *(handled by rejection) and a disagreement component* $[T-S]_+$ *(handled by stochastic acceptance). This characterizes the stochastic acceptance branch in isolation; the finite-candidate procedure with deterministic fallback is described by the mixture in Corollary 7.3, and under the reported hyperparameters the fallback branch is taken on the large majority of decoding steps (Section 7.5.4).*

Proof. A token v is proposed with probability $T(v)$ and accepted with probability $\alpha(v) = [\,1 - S(v)/T(v)\,]_+$. For any $v \in \mathcal{V}_0$, the joint probability of proposal-and-acceptance is

$$T(v)\,\alpha(v) = T(v)\Big[\,1 - \tfrac{S(v)}{T(v)}\,\Big]_+ = [\,T(v) - S(v)\,]_+,$$

where the right-hand side covers both cases ($T(v) > S(v)$ and $T(v) \le S(v)$) by direct verification. Summing over v and using the identity $\sum_v [\,T(v) - S(v)\,]_+ = \frac{1}{2}\sum_v |T(v) - S(v)| = \mathrm{TV}(T,S)$, which follows because $\sum_v (T(v)-S(v)) = 0$ implies $\sum_v [\,T(v)-S(v)\,]_+ = \sum_v [\,S(v)-T(v)\,]_+$, yields claim 1. Claim 2 follows immediately by normalizing $T(v)\alpha(v) = [\,T(v) - S(v)\,]_+$ by $\Pr[\text{accept}] = \mathrm{TV}(T,S)$, which is well-defined whenever $\mathrm{TV}(T,S) > 0$. For claim 3, observe that $T(v)(1-\alpha(v)) = \min\{T(v), S(v)\}$ by direct case analysis: if $T(v) > S(v)$, the left-hand side equals $T(v) \cdot S(v)/T(v) = S(v) = \min\{T(v), S(v)\}$; if $T(v) \le S(v)$, it equals $T(v) \cdot 1 = T(v) = \min\{T(v), S(v)\}$. Summing gives $\Pr[\text{reject}] = \sum_v \min\{T(v), S(v)\} = 1 - \mathrm{TV}(T,S)$, and normalizing by this quantity yields the claim. The full derivation is given in Appendix E.1. □

Remark 7.2 (Coupling interpretation)**.** *Theorem 7.1 gives MDKD a clean coupling interpretation. Teacher sampling decomposes pointwise into an overlap component* $\min\{T,S\}$*, corresponding to tokens already covered by the student, and a residual component* $[T-S]_+$*, corresponding to teacher-supported mass missing from the student. The MDKD stochastic acceptance rule samples exactly from this residual component. Standard teacher sampling, by contrast, allocates most of its probability to the overlap, where a training update carries little discrepancy signal because the student already closely matches the teacher; such updates are not necessarily useless, but they are less informative per teacher query.*

Corollary 7.3 (Selected-token distribution under finite candidates)**.** *Suppose the m candidates are drawn i.i.d. with replacement from T, and let $\delta = \mathrm{TV}(T, S)$. Then with probability $1-(1-\delta)^m$ at least one candidate is stochastically accepted, and conditioned on this event (whenever $\delta > 0$) the accepted token is distributed according to the residual distribution $R(v) = [\,T(v) - S(v)\,]_+/\delta$. With the complementary probability $(1-\delta)^m$, no candidate is stochastically accepted and the deterministic fallback selects $\arg\max_j \alpha_j$ from the m candidates, inducing a fallback distribution $F_m(v)$ over $\mathcal{V}_0$. The full selected-token distribution is therefore the mixture*

$$P_{\mathrm{MDKD}}(v) = \left(1-(1-\delta)^m\right) R(v) + (1-\delta)^m\, F_m(v). \tag{7.6}$$

Proof. By Theorem 7.1, each of the m candidates is independently accepted with marginal probability δ, so the probability that none is accepted is $(1-\delta)^m$. Conditioned on the complementary event, every accepted candidate has the same conditional law R, and the law of total probability yields Eq. (7.6). See Appendix E.1. □

Remark 7.4 (The deterministic fallback)**.** *The fallback distribution F_m is* not *uniform teacher sampling. By construction it selects, within each drawn candidate tuple, the token the student most underweights. It is not, however, the unconditional law of* $\arg\max_{j\in[m]} \alpha_j$ *over m i.i.d. draws from T: the fallback is reached only when* every *candidate has been rejected, and conditioning on that event re-weights the tuples toward those whose candidates all carried small acceptance scores. Appendix E.3 states the exact law and gives a counterexample showing that the resulting marginal need not favor the residual. This turns out not to matter in practice: the fallback-policy ablation in Section 7.5.5 shows that replacing the* $\arg\max$ *with a random or teacher-sampled choice leaves accuracy essentially unchanged, while disabling the stochastic branch costs about five points. The quantity $1-(1-\delta)^m$ is the probability that at least one of the m proposals is stochastically accepted. It is a strictly increasing transformation of the one-proposal total variation δ, not itself a total variation distance between the teacher, the student, or the selected-token distributions: it grows when local teacher–student disagreement is large and shrinks when the student already matches the teacher.*

One limiting case is worth recording. If $S = T$ on the nucleus support then $\alpha \equiv 0$, so no candidate is ever accepted, and the $\arg\max$ *in the fallback is a tie among all m candidates, resolved at the first index. Since the candidates are i.i.d. draws from T_τ, MDKD then reduces* exactly *to ordinary nucleus sampling from the teacher: the sampler is a no-op precisely when there is no divergence to exploit.*

Connection to speculative decoding

Theorem 7.1 sharpens the connection to speculative decoding [79]. Speculative decoding accepts a draft proposal v with probability $\min(1, T(v)/S(v))$, which corresponds to sampling from the overlap component $\min\{T, S\}$ with the residual mass handled by a target-side correction, allowing teacher computation to be skipped on tokens where draft and target agree. MDKD's stochastic acceptance rule $[\,1 - S(v)/T(v)\,]_+$ is the exact complement: it samples from the residual component $[T - S]_+$ to extract maximum learning signal on tokens where the student does not agree with the teacher. The two methods thus form dual decompositions of teacher sampling along the same partition

$$T(v) = \min\{T(v), S(v)\} + [\,T(v) - S(v)\,]_+, \tag{7.7}$$

with opposite acceptance polarities and opposite goals: speculative decoding accepts on agreement to skip teacher work; MDKD accepts on disagreement to extract maximum learning signal per teacher query.

Scope of the theoretical results

As in Chapters 3 and 5, we delineate what is proven from what is interpreted. Theorem 7.1 and Corollary 7.3 are exact statements about the *per-step selected-token distribution* for fixed T and S; they make no assumption about the student architecture or training dynamics. They do *not* establish a sequence-level guarantee, since each accepted token shifts subsequent teacher conditionals and biased per-step choices can in principle compound across a trajectory; a formal sequence-level treatment is left to future work (Chapter 8). The stochastic branch that Theorem 7.1 characterizes exactly fires on a minority of decoding steps. Section 7.5.5 shows empirically that it is nonetheless the branch responsible for the method's gain.

7.4.5 Training objective

Given a divergence-aware sequence $x \sim \mu_{\theta^-}$ produced by Algorithm 3 run against a frozen student snapshot S_{θ^-}, the student parameters are updated by minimizing the sum of divergences at each decoding step:

$$\mathcal{L}_{\mathrm{MDKD}}(\theta) \;=\; \mathbb{E}_{x\sim\mu_{\theta^-}}\left[\sum_{i=t+1}^{|x|} D\big(T(\cdot \mid x_{\le i-1}) \,\big\|\, S(\cdot \mid x_{\le i-1}; \theta)\big)\right], \tag{7.8}$$

where D is a chosen divergence measure. Crucially, gradients are *not* propagated through the sampling process: μ_{θ^-} acts as a data-generating distribution, while the loss is computed on the resulting sequences using the live student S_θ. MDKD does not depend on a specific divergence; in our experiments we evaluate forward KL, reverse KL, and Jensen–Shannon divergence, and reverse KL gave the best results in the configurations tested (Section 7.5.8).

On the choice of divergence. Reverse KL, $\sum_v S(v) \log\big(S(v)/T(v)\big)$, penalizes student probability placed where the teacher assigns low probability, and is in that sense mode-seeking. It does *not* directly target teacher-supported regions that the student underweights (forward KL does that more directly), so the sampler does not mathematically imply reverse KL, and the two are separate design choices. Reverse KL was selected as part of the reported empirical configuration because it performed best in the runs of Section 7.5.8; establishing a general advantage would require the matched divergence ablation described there, in which the sampler, temperature, and truncation are held fixed while only the divergence varies.

7.4.6 Practical considerations

Candidate pool size. The number of teacher candidates m controls how widely each decoding step explores the teacher's nucleus. It costs little: the teacher and student distributions are each evaluated once per decoding step, and all m candidates are then sampled and scored in parallel from those two distributions, so increasing m adds no further model forward passes. A small constant (e.g., $m = 5$) suffices in practice; larger values increase the chance of finding tokens the student underweights, and very large m trades teacher accuracy for only marginal gains in difficulty (Section 7.5.8).

Fallback strategy. When no candidate passes the stochastic acceptance test, the algorithm selects the candidate with the highest relative teacher-undercoverage score α_j. Within the drawn candidate tuple this is by construction the token the student most underestimates. Remark 7.4 notes that the resulting marginal F_m need not favor $[T - S]_+$ for every (T, S), and Appendix E.3 gives a counterexample; we make no claim about which way it leans in the deployed regime. The ablation of Section 7.5.5 shows that the fallback rule is in any case not what drives the method.

Temperature and nucleus filtering. The teacher temperature τ controls how peaked the candidate distribution is: lower values concentrate candidates around the teacher's top

predictions, while higher values encourage diversity. The nucleus threshold p_{nuc} restricts candidates to a core set of plausible tokens, preventing the sampling of very low-probability tokens that would produce incoherent text. As shown in Section 7.5.8, the optimal combination is task-dependent.

Integration with other KD methods. MDKD is orthogonal to the choice of divergence in the training loss. Any f-divergence can be used in Algorithm 4, and the generation procedure (Algorithm 3) remains unchanged.

7.4.7 Training algorithm

The overall MDKD training procedure combines the divergence-aware sampling of Algorithm 3 with standard KD optimization. Algorithm 4 summarizes the training loop. For each mini-batch of prompts, it snapshots the live student, generates divergence-aware responses against the snapshot, computes the divergence between teacher and the live student over the generated sequences, and updates the student parameters; the snapshot is refreshed once per mini-batch.

In this training loop, MDKD_GENERATE refers to Algorithm 3. The generation step constructs sequences that target teacher–student disagreement; the training step computes a divergence loss on these sequences and updates the student. Note that the divergence D in the loss is a design choice independent of the generation procedure: we experiment with forward KL, reverse KL, and Jensen–Shannon divergence in Section 7.5.8.

7.5 Experiments

We evaluate MDKD across three text-generation task families, arithmetic reasoning, dialogue summarization, and code generation, on two architecturally distinct teacher–student pairs. All methods share the same training data, teacher–student pairs, number of epochs, effective batch size, and evaluation decoding. Their training-time configurations differ in more than the sequence sampler: MDKD uses reverse KL with a teacher sampler at $\tau = 0.7$ and nucleus $p_{\text{nuc}} = 0.95$, whereas the baselines use forward KL with the method-specific temperatures and top-p values of Table 7.6. The main comparison therefore measures the fully configured methods rather than the isolated causal effect of the MDKD sampler. Table 7.7 isolates the divergence by re-running every baseline under MDKD's reverse-KL

objective; holding temperature and truncation fixed across methods as well remains future work.

7.5.1 Setup

Teacher–student pairs. We use two pairs that span the size and pre-training conventions common in modern open-weight releases: (i) QWEN2.5-14B-INSTRUCT → QWEN2.5-1.5B-INSTRUCT [101], a $\sim 9\times$ parameter compression with both checkpoints instruction-tuned, and (ii) GEMMA-3-12B-IT → GEMMA-3-1B-IT [48], a $\sim 12\times$ compression on a different base architecture and pre-training corpus. The teachers are used off-the-shelf without further fine-tuning. Each student is first lightly task-SFT'd on the same 1K-sample subset that subsequently serves as the KD training data, providing a shared initialization for every KD method so that the comparison isolates the KD procedure itself.

Training. All methods share the same training infrastructure: 8× AMD MI300X GPUs, per-GPU batch size 8 (effective batch 64 via DDP through HuggingFace Accelerate), bfloat16 precision, learning rate 1×10^{-5} with cosine schedule and no warmup, gradient clipping at 1.0, and seed 42. Each KD method runs for 3 epochs over the KD training set (1,000 examples for GSM8K and summarization; all 374 MBPP problems for code generation) and checkpoints are saved during training. For each run, the checkpoint used for the final evaluation is the saved checkpoint with the lowest validation loss. The validation loss is the distillation divergence between the student and the teacher, computed every 25 training steps on a held-out validation set of prompts with their reference answers (a teacher-forced loss, with no generation involved). Test scores were never used to select checkpoints, so the reported test results are not affected by selection on the test data. At evaluation time the selected checkpoint is decoded with greedy decoding (vLLM, max length 2,048, seed 42). Following common practice for experiments at this scale, each configuration was run with a single training seed (42); the infrastructure is not bit-for-bit deterministic (nondeterministic CUDA kernels, DDP all-reduce), so small differences between methods should be read as results of the reported run rather than as estimates with a quantified run-to-run variance.

Baselines. We compare MDKD against three established sampling-based KD baselines, all using forward KL on the teacher's next-token distribution:

- **Supervised KD:** forward KL between teacher and student on teacher-labeled sequences.

- **On-Policy KD (GKD)** [1]: the student generates sequences from its own policy; the teacher provides token-level soft labels.
- **Speculative KD (SKD)** [140]: the student proposes tokens; the teacher replaces those outside its top-K candidates and resamples from its own distribution.

MDKD uses reverse KL with a teacher sampler at $\tau = 0.7$, nucleus $p_{\text{nuc}} = 0.95$, and $m = 5$ candidates per token. Baseline hyperparameters follow the original SKD repository.

Evaluation. All models use greedy decoding at evaluation. Task-specific metrics are described below with each task.

7.5.2 Tasks

Arithmetic Reasoning (GSM8K). We use the GSM8K dataset [30], consisting of grade-school math word problems that require multi-step reasoning. Following [140], we randomly sample 1,000 problems from the training set as our KD training data and evaluate on the full GSM8K test set (1,319 problems) using a lenient (format-agnostic) symbolic grader that checks numerical equivalence so that solutions in either teacher's preferred chain-of-thought style are scored uniformly.

Dialogue Summarization (DialogSum). We use the DialogSum dataset [22], which pairs multi-turn dialogues with human-written summaries, augmented with the SAMSum corpus [49] to broaden the conversation distribution. We randomly sample 1,000 dialogue–summary pairs for training and evaluate on 500 held-out dialogues using ROUGE-L [83].

Code Generation (MBPP→HumanEval). We adopt a cross-distribution evaluation: we train on the MBPP dataset [4] (374 program-synthesis problems) and evaluate Pass@1 on HumanEval [19] (164 problems) with execution-based grading. Pass@1 is the fraction of the 164 problems solved by a single greedy completion, so every reported value is an exact multiple of $1/164 \approx 0.61\%$; the figures in Table 7.1 correspond to 76, 72, 73, and 82 problems solved for the Qwen pair and 52, 53, 53, and 59 for the Gemma pair. MBPP and HumanEval differ in problem format, function-naming conventions, and test-case style, so this setup probes whether distillation transfers programming competence beyond the training distribution rather than memorizing surface patterns of MBPP.

Table 7.1: Main results: KD methods compared across two teacher–student pairs and three task families. We report GSM8K accuracy, DialogSum ROUGE-L on 500 held-out dialogues, and HumanEval Pass@1 (execution-based) when training on MBPP. All methods use the same training set, a matched optimization budget (3 epochs, same batch size), and the same evaluation decoding, but method-specific sampling and divergence hyperparameters (Table 7.6). Bold marks the best method per column; MDKD is best on every (model, task) cell.

Method	**Qwen2.5-14B-it → 1.5B-it**			**Gemma-3-12B-it → 1B-it**		
	GSM8K Acc. (%)	Summ. ROUGE-L	HumanEval Pass@1 (%)	GSM8K Acc. (%)	Summ. ROUGE-L	HumanEval Pass@1 (%)
Supervised KD	33.2	36.61	46.34	33.7	36.85	31.71
SKD	33.0	36.03	43.90	36.3	35.91	32.32
On-Policy KD (GKD)	41.1	36.88	44.51	33.8	35.31	32.32
MDKD (ours)	**44.4**	**37.39**	**50.00**	**41.8**	**37.24**	**35.98**

7.5.3 Main Results

Table 7.1 reports KD performance on six (model-pair, task) cells. MDKD wins *every* cell, on both architecture families and across all three task types, despite the methods sharing identical optimization budgets, batch sizes, and KD training data.

7.5.4 Divergence Gain Analysis

Before presenting the broader analysis, we verify the core premise of MDKD: that divergence-aware sampling produces sequences that are simultaneously coherent and challenging for the student. We generate 512 responses on GSM8K using (a) standard teacher sampling and (b) MDKD sampling, then measure the student's cross-entropy loss on each set. For this analysis we use Qwen2.5-7B-Instruct as teacher and the base Qwen2.5-0.5B (no instruction tuning) as student.

As shown in Table 7.2, MDKD sacrifices only 1.9% accuracy while increasing the student's cross-entropy loss by 43.6%. The final-answer grader shows that MDKD sequences retain similar answer accuracy in this sample; it does not verify the correctness of every intermediate reasoning step. The sequences are systematically harder for the student to predict. Per-sample analysis reveals that 399 of the 512 samples are solved correctly by

Table 7.2: Divergence gain analysis on GSM8K (512 samples). MDKD sequences maintain near-teacher accuracy while increasing the student's cross-entropy loss by 43.6%, confirming that they target the student's blind spots.

Method	**Accuracy**	**Avg Length**	**Student CE Loss**
Teacher-only	83.0%	251.9	0.1778
MDKD	81.1%	248.6	0.2553
Delta	−1.9%	−3.3	**+0.0775 (+43.6%)**

both methods, 26 only by the teacher, and 16 only by MDKD, indicating that the quality degradation is minimal and largely confined to borderline cases. The rejection-sampling statistics further illuminate the mechanism: of 127,296 total token decisions, only 10.3% were accepted via the stochastic rejection step; the remaining 89.7% fell through to the fallback path. The fallback-policy ablation of Section 7.5.5 shows that this split does not undermine the method: the gain comes from the stochastic branch, and the fallback rule is interchangeable. The reading is that MDKD performs a gentle, continuous nudge toward divergent tokens rather than aggressively replacing them, preserving coherence while biasing the output toward harder content.

7.5.5 Interpreting the Acceptance Rate

The acceptance rate is not a hyperparameter; it is a measurement. By Theorem 7.1 the per-candidate acceptance probability equals $\mathrm{TV}(T_\tau, S_\tau)$, and by Corollary 7.3 the branch rate $1-(1-\delta)^m$ is a strictly increasing function of it. The 10.3% measured above for the 7B→0.5B pair is therefore the algorithm reporting that the student already covers most of the teacher's local probability mass. Teacher and student here are drawn from the same model family (Qwen2.5) with near-identical pre-training data, so agreement on the bulk of each next-token distribution is the expected behavior of a competent student, not a shortfall. A weaker or architecturally mismatched student would produce a higher rate; at $\tau = 1$ a student identical to the teacher produces a rate of exactly zero, since then $S_\tau = T_\tau$ and $\alpha \equiv 0$. Appendix E.2.1 reports a diagnostic sweep confirming this ordering. The name "maximum divergence" refers to the algorithm always moving toward the largest *available* residual; it is not a claim that the residual is large in magnitude.

One qualification applies throughout. The quantity δ is a *filtered* divergence: Algorithm 3 temperature-scales and nucleus-truncates the teacher, then renormalizes the stu-

Table 7.3: Fallback-policy ablation. Only the fallback rule varies; the teacher (Qwen2.5-14B-Instruct), student (base Qwen2.5-1.5B), data (1K GSM8K training samples), loss (reverse KL), and the full 1,319-problem test set are held fixed, and every configuration is trained for three epochs. The absolute accuracies are therefore not comparable with Table 7.5, whose 1K column trains for five epochs and reports 64.90 for the same pair; this ablation is a within-table comparison of fallback policies, not a restatement of the headline result. Every policy that retains the stochastic acceptance step performs comparably; disabling it does not.

Fallback policy	**GSM8K Acc. (%)**
$\arg\max_j \alpha_j$ (default MDKD)	70.20
Random (uniform over the m candidates)	68.84
Teacher-sample (redraw from T_τ)	71.42
No fallback (resample until accepted)	71.19
Fallback-only (stochastic acceptance disabled)	65.58

dent over the resulting support, so it absorbs the effect of τ and p_{nuc} as well as genuine model disagreement. A given pair therefore registers different rates at different temperatures: against the 14B teacher, the 1.5B student accepts 15.35% at $\tau = 1$ (measured with $m = 10$ candidates, as described in Appendix E.2.1) and 10.23% at the deployed $\tau = 0.7$ with $m = 5$. That last figure is a different measurement from the 10.3% above, which was recorded on a different pair; their near-coincidence is accidental.

This leaves one question. Theorem 7.1 is exact only for the stochastic branch, and that branch fires on a minority of steps; does the theory then describe only a small part of what the algorithm does? Table 7.3 answers it directly, varying only the fallback policy with everything else held fixed.

Every variant that keeps the stochastic step recovers the student into a tight 68.8–71.4% band, and pure rejection with no fallback at all matches default MDKD (71.19 against 70.20). The one variant that clearly degrades is fallback-only, which disables stochastic acceptance and loses roughly five points. The gain therefore comes from the stochastic residual-sampling step rather than from the particular fallback rule, which is interchangeable. Theorem 7.1 characterizes exactly the branch that produces the effect, and Corollary 7.3 fixes how often that branch is invoked, even though it fires on a minority of decoding steps.

A low rate does not imply a small effect. Because each next-token distribution is conditioned on the full prefix, a single accepted token early in a generation re-routes every

Table 7.4: In-domain transfer. *Left:* best Math-500 accuracy after s1K training (Qwen2.5-1.5B-Instruct student, Qwen2.5-14B-Instruct teacher). *Right:* UI-math distillation with 2K samples from the off-the-shelf 14B teacher; GSM8K measures within-domain transfer.

Model / Method	**Math-500 Acc.**
Student-IT	46.80%
SFT Student	33.60%
Supervised KD	40.60%
On-Policy KD (GKD)	47.20%
MDKD	**50.00%**

Model / Method	**UI-math Dev**	**GSM8K**
SFT student	27.35%	8.57%
Supervised KD	64.07%	44.20%
On-Policy KD (GKD)	69.66%	31.01%
SKD	56.69%	47.08%
MDKD	**70.66%**	**54.59%**

subsequent teacher conditional, and accepted tokens are in practice concentrated in the opening of each sequence. Appendix E.2.2 gives a worked example in which teacher and student produce near-identical, both-correct solutions while MDKD, accepting only 15.8% of tokens, follows an entirely different and still-correct derivation. This is the same effect the divergence-gain analysis measures in aggregate: a small fraction of switched tokens yields a +43.6% increase in the student's cross-entropy on the resulting sequences.

7.5.6 In-Domain Transfer and Regularization

We next ask whether distillation improves only the training task or whether it preserves broader competence when the student is evaluated on a different in-domain benchmark, since math KD can easily over-specialize. We evaluate two forms of in-domain transfer. First, we train on the long chain-of-thought s1K subset and evaluate on Math-500. Second, we train on UltraInteract-math (UI-math) and evaluate both on held-out UI-math and on GSM8K.

Table 7.4 (left) shows that s1K training is a difficult regularization test: SFT on 900 long-CoT examples reduces the 1.5B-Instruct student's Math-500 accuracy from 46.80% to 33.60%, and Supervised KD remains below the off-the-shelf student. GKD is approximately neutral, ending at 47.20%. MDKD is the only method that gives a clear positive transfer result, reaching 50.00% on Math-500, a +3.20 pp gain over the original student. The larger UI-math experiment (Table 7.4, right) gives the same conclusion under a more controlled shared-initialization protocol: after only 2K KD samples, MDKD is best on both the source dev set (70.66%) and the GSM8K transfer benchmark (54.59%, +7.51 pp ahead of the next-best GSM8K method). Together these results indicate that MDKD behaves as a *regularized* distillation objective rather than a narrow source-task optimizer.

Table 7.5: Scaling experiments: 14B teacher → 1.5B student on GSM8K. MDKD is the strongest method at both data sizes. Results shown at the best epoch for each method.

Method	1K data (5 epochs)	7K data (3 epochs)
SKD	31.77%	39.95%
Supervised KD	35.25%	41.77%
On-Policy KD (GKD)	54.59%	62.77%
MDKD (ours)	**64.90%**	**71.72%**
Base Student	8.26%	
Base Teacher	94.92%	

7.5.7 Scaling to Larger Teachers

To test how MDKD scales with teacher capacity, we distil Qwen2.5-14B-Instruct into a base Qwen2.5-1.5B student on GSM8K. This is a more challenging scenario: the student starts at only 8.26% accuracy (no instruction tuning) against a 94.92% teacher.

Table 7.5 shows that MDKD's advantage is dramatic with a larger teacher. With 1K training samples, MDKD (64.90%) outperforms On-Policy KD (54.59%) by over 10 points and Supervised KD (35.25%) by nearly 30 points, recovering 65% of the teacher–student gap. Scaling to 7K samples improves all methods, but the ranking is preserved: MDKD reaches 71.72%, recovering 73% of the gap. The richer probability distribution of a larger teacher provides more divergence to exploit, amplifying the benefit of MDKD's divergence-aware sampling.

7.5.8 Hyperparameter Analysis

Table 7.6 summarizes the MDKD hyperparameters used across tasks. For the main results we use a single shared MDKD configuration ($\tau = 0.7$, $p_{\text{nuc}} = 0.95$, $m = 5$, reverse KL) across all tasks, so that gains over the baselines cannot be attributed to per-task hyperparameter tuning. Two parameters merit discussion.

Distance metric. In our runs, reverse KL gave the best results among forward KL, JSD, and plain cross-entropy, and it is the divergence used in the main MDKD configuration. Reverse KL is mode-seeking: it encourages the student to concentrate probability mass on the teacher's top predictions rather than spreading it across the entire distribution,

Table 7.6: MDKD and baseline hyperparameters. Baseline parameters follow the original SKD repository; MDKD uses a single shared configuration across tasks.

Parameter	**GSM8K**	**Summarization**
Shared across all methods		
Learning rate	1×10^{-5}	1×10^{-5}
Effective batch size	64	64
Precision	bf16	bf16
Baselines (SKD / GKD / Supervised KD)		
Teacher temperature	0.2	0.2
Distance metric	Forward KL	Forward KL
Teacher top-p	1.0	0.5
MDKD (ours)		
Teacher temperature (τ)	0.7	0.7
Distance metric	Reverse KL	Reverse KL
Nucleus top-p	0.95	0.95
Candidates (m)	5	5

which for a capacity-limited student is more efficient than the mode-covering behavior of forward KL. Because the baselines run forward KL, Table 7.7 re-runs each of them under reverse KL; MDKD still leads by more than six points, so its advantage is not explained by the choice of divergence alone, though temperature and truncation still differ between the methods.

Candidate pool size. Varying $m \in \{5, 15, 25, 50\}$ on GSM8K at fixed $\tau = 0.7$ and $p_{\text{nuc}} = 0.95$ over $N = 512$ problems, and comparing at each m against a teacher-only baseline decoded at the same temperature, MDKD raises mean student cross-entropy from the teacher-only value of 0.289 into the range 0.366–0.369, an increase of 26–28%, while keeping teacher accuracy within about 2 pp of the teacher-only rate for $m \leq 25$. Student cross-entropy stays in this narrow band across the sweep, so the difficulty gain over teacher-only sampling is stable in the pool size; the largest setting ($m = 50$) trades more teacher accuracy (-2.35 pp) for only a marginal change in cross-entropy. A small pool ($m = 5$) is therefore preferred not on cost grounds, since all candidates are scored in parallel, but because a larger pool buys no additional difficulty while giving up teacher accuracy. Corollary 7.3 implies that the stochastic branch fires more often as m grows, so the flat cross-entropy

Table 7.7: Loss-controlled baselines on GSM8K (Qwen2.5-14B-Instruct teacher, base Qwen2.5-1.5B student, 7K training samples). The forward-KL column reproduces the 7K results of Table 7.5; the reverse-KL column re-runs each baseline under MDKD's objective. MDKD is defined with reverse KL, so no forward-KL entry applies.

Method	**Forward KL**	**Reverse KL**
Supervised KD	41.77	31.10
On-Policy KD (GKD)	62.77	65.26
SKD	39.95	65.45
MDKD (ours)	n/a	**71.72**

indicates that the two branches produce tokens of comparable difficulty at this operating point, consistent with the fallback ablation of Section 7.5.5.

7.5.9 Discussion

Across two architecture families and three task families, MDKD demonstrates consistent effectiveness as a knowledge distillation method for autoregressive language models, obtaining the highest reported point estimate in every (model-pair, task) cell against Supervised KD, On-Policy KD, and SKD (Table 7.1); it is also the strongest method on in-domain transfer (Section 7.5.6) and its margin grows with teacher capacity (Section 7.5.7).

The divergence-gain analysis (Section 7.5.4) provides a mechanistic explanation consistent with Theorem 7.1: MDKD sequences are 43.6% harder for the student (in terms of cross-entropy) while maintaining 98% of the teacher's accuracy. Because the stochastic step samples exactly from the uncovered residual $[T - S]_+$, this "targeted difficulty" concentrates the gradient signal per training step on tokens the student does not yet cover, allowing the student to learn more efficiently from the same number of training samples; the fallback-policy ablation of Section 7.5.5 confirms that it is this branch, and not the fallback rule, that produces the gain. MDKD's largest gains arise on tasks with structured, multi-step outputs (math reasoning) and with larger teacher–student capacity gaps (14B$\rightarrow$1.5B vs. 7B$\rightarrow$0.5B), suggesting it is most valuable when the output space is large enough that the student has substantial room for disagreement.

The principal cost of MDKD is training-time overhead: it queries the teacher and a student snapshot at decode time during training, making it noticeably slower per epoch than Supervised KD on a fixed teacher-labeled corpus. This cost is not driven by hyperparameter search, since MDKD is robust to the candidate pool size m and uses a single

shared nucleus threshold across tasks; the remaining bottleneck is generation, and integrating MDKD with efficient inference engines would bring its wall-clock training cost much closer to that of Supervised KD. We return to this limitation and to the open problem of a sequence-level theoretical guarantee in Chapter 8.

Algorithm 3 MDKD_Generate: Divergence-aware response generation

function MDKD_GENERATE($x_{1:t}$, T, S, n, m, τ, p_{nuc})
    // $x_{1:t}$: prompt; T: teacher; S: student; n: max response length; m: candidates per step; τ: temperature; p_{nuc}: nucleus threshold
    $x \leftarrow x_{1:t}$
    for $i = t + 1$ to $t + n$ **do**
        $T_\tau(\cdot \mid x) \leftarrow \text{NucleusFilter}\big(\text{softmax}\big(T_{\text{logits}}(x)/\tau\big),\ p_{\text{nuc}}\big)$ ▷ Temperature-scaled, nucleus-filtered teacher
        $S_\tau(\cdot \mid x) \leftarrow \text{Renormalize}\big(S(\cdot \mid x),\ \text{supp}(T_\tau)\big)$ ▷ Renormalize student over nucleus support
        $\hat{x}_1, \ldots, \hat{x}_m \sim T_\tau(\cdot \mid x)$ ▷ Draw m candidates i.i.d.
        accepted $\leftarrow$ **False**
        for $j = 1$ to m **do**
            $\alpha_j \leftarrow \max\big(0,\ 1 - S_\tau(\hat{x}_j \mid x) \,/\, T_\tau(\hat{x}_j \mid x)\big)$
            if $\text{Uniform}(0, 1) < \alpha_j$ **then**
                $x \leftarrow x \circ \hat{x}_j$ ▷ Append accepted token
                accepted $\leftarrow$ **True**; **break**
            end if
        end for
        if not *accepted* **then**
            $j^* \leftarrow \arg\max_j \alpha_j$
            $x \leftarrow x \circ \hat{x}_{j^*}$ ▷ Fallback: largest relative shortfall
        end if
        if appended token is $\langle \text{EOS} \rangle$ **then break**
        end if
    end for
    return x
end function

Algorithm 4 MDKD Training Loop

function MDKD-TRAINING(S, T, $\mathcal{D}$, n, m, τ, p_{nuc}, B, η)
    // S: student, T: teacher, $\mathcal{D}$: dataset, n: max response length, m: candidates, τ: temperature, p_{nuc}: nucleus, B: batch size, η: learning rate
    for each mini-batch $\{x^{(i)}\}_{i=1}^{B}$ obtained via GET-MINI-BATCH($\mathcal{D}$) **do**
        $S_{\theta^-} \leftarrow \text{detach}(S_\theta)$ ▷ Frozen snapshot for generation
        for $i = 1$ to B **do**
            $\hat{y}^{(i)} \leftarrow \text{MDKD_GENERATE}(x^{(i)}, T, S_{\theta^-}, n, m, \tau, p_{\text{nuc}})$
        end for
        $\mathcal{L} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \sum_{t=|x^{(i)}|+1}^{|\hat{y}^{(i)}|} D\Big(T(\cdot \mid \hat{y}^{(i)}_{<t}) \,\|\, S(\cdot \mid \hat{y}^{(i)}_{<t}; \theta)\Big)$ ▷ D: KL, reverse KL, or JSD
        BACKPROPAGATION($\mathcal{L}$, θ_S) ▷ Gradients flow through S_θ only
    end for
    return S
end function

Chapter 8

Conclusion and Future Work

This thesis has investigated the fundamental limitations of classical Knowledge Distillation and proposed five novel methods that advance the field across three complementary paradigms: logit distillation, data distillation, and divergence-aware distillation. Below, we summarize the contributions, discuss their limitations, and outline directions for future research.

8.1 Summary of Contributions

8.1.1 Logit Distillation: Addressing the Capacity Gap, Checkpoint Search, and Noisy Teacher Problems

The first three contributions of this thesis (Chapters 3–5) addressed the optimization and information-theoretic bottlenecks of classical logit distillation.

Annealing Knowledge Distillation (Chapter 3). We proposed Annealing-KD, a technique that bridges the capacity gap between teacher and student without introducing auxiliary networks. By applying a dynamic temperature function exclusively to the teacher's logits, Annealing-KD constructs an information-theoretic curriculum: building on known properties of the temperature-scaled softmax, we showed that the Shannon entropy of the teacher's temperature-scaled distribution is a monotonically increasing function of temperature (strictly, whenever the logits are not all equal); so lowering the temperature

over training monotonically decreases the target's entropy, exposing the student to progressively sharper targets. This mechanism acts as a low-pass filter on the teacher's high-frequency targets, which we relate, by analogy with the spectral bias of neural networks, to a coarse-to-fine ordering of the supervisory signal. Experiments on CIFAR-10, CIFAR-100, and the GLUE benchmark demonstrated consistent improvements over Vanilla KD and TAKD. We further showed the effectiveness of Annealing-KD in hybrid LLM upcycling tasks.

Progressive Knowledge Distillation (Chapter 4). We proposed Pro-KD, a progressive training procedure that simultaneously addresses the capacity-gap and checkpoint-search problems. Rather than distilling from a single, fully converged teacher checkpoint, Pro-KD has the student follow the teacher's training trajectory from the beginning, learning from progressively more capable snapshots. Combined with the adaptive temperature schedule from Annealing-KD, this creates a doubly smooth curriculum: the teacher itself progresses from simple to complex representations, while the temperature annealing further softens each snapshot's outputs. Pro-KD eliminates the need for costly checkpoint search and consistently outperformed Vanilla KD, TAKD, Annealing-KD, and RCO on CIFAR-10, CIFAR-100, the GLUE benchmark, and SQuAD 1.1/2.0.

Continuation Knowledge Distillation (Chapter 5). We proposed Continuation-KD, a unified training framework inspired by the continuation method from optimization theory. Continuation-KD generalizes Annealing-KD by replacing its rigid two-stage structure with a single scheduled objective, and by introducing an annealed hinge-loss margin that suppresses gradient updates from small residual noise in the teacher's predictions. Its mechanism is the annealed hinge margin, which creates a region where residuals below $\Phi\, m$ produce zero gradient and zero Hessian, so noise-level differences from the teacher are not transferred to the student; a simple noise model gives the scale $m = K\sigma^2$. Two exact identities make this precise: a gradient identity and the Hessian decomposition $\nabla^2 \mathcal{L}_{\mathcal{T}} = H_S - \Phi\, C$, each isolating the teacher's contribution in a single term scaled linearly by the annealing factor. These characterize the objective's structure; the link to the observed empirical gains is not established by our experiments. Continuation-KD consistently outperformed Vanilla KD, TAKD, and Annealing-KD across both computer vision and NLP tasks.

8.1.2 Data Distillation: Citation-Grounded Question Answering

Efficient Citer (Chapter 6). Moving from white-box to black-box distillation, we applied the data distillation paradigm to the practically important task of citation-grounded question answering. We used ChatGPT as a teacher to synthesize a training corpus of answers enriched with inline citations from the MS MARCO dataset, and fine-tuned two compact student models: FIDCiter (3B parameters) and LLaMACiter (13B parameters). The teacher was used as a gold-conditioned rewriter, in that it received the human gold answer along with the query and passages, so this is data distillation from a rewriter rather than from an independently answering teacher, and the evaluation-time ChatGPT baseline does not receive the same inputs. LLaMACiter exceeded the ChatGPT teacher on human-rated informativeness on all four evaluation sets and FIDCiter on two of the four, and FIDCiter matched or exceeded it on in-domain citation quality, while using much shorter instruction templates; the teacher retained an edge on citation quality on the out-of-domain evaluation sets. These results show that a substantial part of the citation behavior can be distilled into efficient, deployable models.

8.1.3 Divergence-Aware Distillation: Active Knowledge Transfer for Generative Models

Maximum Divergence Knowledge Distillation (Chapter 7). Building upon our prior work on Backward KD in continuous domains, we proposed MDKD, which translates the active-divergence philosophy into the discrete, autoregressive setting of modern LLMs. At each decoding step, MDKD uses a rejection-sampling mechanism to preferentially select tokens where the student underestimates the teacher, constructing training sequences concentrated in regions where the student underweights teacher-supported tokens. We proved that the stochastic acceptance branch samples exactly from the positive residual $[T - S]_+$ with acceptance probability equal to the teacher–student total variation distance, and characterized the full finite-candidate sampler as a mixture of this residual branch and a deterministic fallback. An ablation confirmed that the stochastic branch is the source of the method's gain: replacing the fallback rule with a random or teacher-sampled choice leaves accuracy essentially unchanged, whereas disabling stochastic acceptance costs about five points. MDKD targets the distributional-coverage aspect of the generative distribution-mismatch problem; because its prefixes come from teacher proposals rather than the student's own policy, direct on-policy exposure-bias mitigation is left to future work.

8.2 Unifying Themes

Despite spanning three distinct paradigms, the contributions of this thesis are connected by several unifying themes.

Dynamic supervisory signals. A recurring motif throughout this thesis is the rejection of static training objectives in favor of dynamically evolving supervisory signals. Annealing-KD schedules the temperature, Pro-KD schedules the teacher's training state, Continuation-KD schedules the objective function itself, and MDKD dynamically constructs training data based on the current student's weaknesses. In each case, the key insight is the same: the supervisory signal should adapt to the student's current learning state rather than remaining fixed throughout training.

From passive to active distillation. The thesis traces a progression from passive distillation, where the student minimizes divergence over a fixed dataset, to increasingly active paradigms. Chapters 3–5 dynamically control the teacher's signal; Chapter 6 constructs new training data via teacher prompting; and Chapter 7 actively seeks out the student's blind spots through divergence-maximizing data construction. This progression reflects a broader trend in the field toward treating distillation as an interactive, adaptive process.

Bridging theory and practice. Each contribution is developed within an explicit conceptual frame: information theory (Annealing-KD), the capacity-gap bound (Pro-KD), continuation optimization (Continuation-KD), and rejection sampling (MDKD). We give formal results where the frame supports them (the entropy-monotonicity theorems of Chapter 3, the exact identities of Chapters 5 and 7) and explicit motivation where it does not (Pro-KD's use of the capacity-gap bound). All four methods remain simple to implement and compatible with standard training infrastructure. This balance between conceptual grounding and practical accessibility is a deliberate design choice throughout the thesis.

8.3 Limitations

While the proposed methods advance the state of the art across multiple settings, they have several limitations that should be acknowledged.

Scope of theoretical results. The theoretical analyses in Chapters 3 and 5 rigorously establish exact properties of the annealed objective (strict entropy monotonicity of the temperature-scaled teacher distribution, an exact gradient identity, and an exact Hessian decomposition). These are structural results: algebraic identities about the objective at a fixed parameter point. For the non-homogeneous Transformer students used in practice they do not, on their own, imply a lower gradient magnitude or a lower Hessian spectral norm, and the connection between these exact properties and the improved optimization and generalization we observe on GLUE and CIFAR is not established by our experiments. Bridging this gap, for example by proving that entropy- or scaling-controlled supervision yields provably tighter generalization bounds for the student, remains an open theoretical challenge.

Hyperparameter sensitivity. All proposed methods introduce new hyperparameters (maximum temperature $\tau_{\max}$ for Annealing-KD and Pro-KD, the stage lengths for Annealing-KD, the hinge margin schedule for Continuation-KD, and the candidate pool size m for MDKD). While we demonstrated robustness to these hyperparameters through ablation studies, the optimal settings can vary across tasks and model configurations. Developing principled, adaptive rules for setting these hyperparameters would improve the methods' usability.

Evaluation scope. The logit distillation methods (Chapters 3–5) were evaluated primarily on image classification (CIFAR-10/100) and natural language understanding (GLUE, SQuAD) benchmarks, with a smaller-scale generative evaluation of Annealing-KD on hybrid-LLM upcycling (Chapter 3). They have not been systematically evaluated on larger-scale generative tasks such as open-ended text generation, code synthesis, or mathematical reasoning, where the interplay between capacity gap and sequence-level generation may introduce additional challenges.

Computational overhead of MDKD. The rejection-sampling mechanism in MDKD requires evaluating both teacher and student probabilities at each decoding step during data construction, which increases the computational cost of generating training sequences relative to standard teacher-only generation. This cost is driven by the two distribution evaluations per decoding step, not by the candidate pool size: all m candidates are sampled and scored in parallel from those distributions, so increasing m requires no additional model forward passes and adds only negligible sampling and scoring work. Wall-clock scaling with m was not measured directly.

Data distillation quality ceiling. The Efficient Citer framework (Chapter 6) is inherently bounded by the quality of the teacher's generated data. If the teacher produces hallucinated or incorrectly attributed citations, these errors propagate directly to the student. While we mitigated this through post-hoc cleaning of the distilled dataset, a more principled approach to quality control during data distillation would strengthen the framework.

8.4 Future Research Directions

The work presented in this thesis opens several promising avenues for future research.

Unified adaptive distillation frameworks. Annealing-KD, Pro-KD, and Continuation-KD each adapt a different aspect of the distillation process (temperature, teacher state, and objective function, respectively). A natural extension is to develop a unified framework that simultaneously and jointly optimizes all three dimensions, potentially using meta-learning or bilevel optimization to adaptively determine the optimal schedule for each component based on the student's current learning state.

Scaling to foundation model distillation. The logit distillation methods developed in Chapters 3–5 were validated on models with up to hundreds of millions of parameters. Extending these techniques to the distillation of modern foundation models with tens or hundreds of billions of parameters, where the capacity gap is even more extreme, is an important direction. This may require combining the proposed methods with parameter-efficient fine-tuning techniques such as LoRA [60] and quantization-aware training.

End-to-end theoretical guarantees. As noted in the limitations, the theoretical analysis of this thesis establishes exact properties of the teacher's distribution but does not formally connect these to student generalization bounds. Future work could aim to prove that entropy-controlled or continuation-based curricula yield tighter PAC-Bayes or Rademacher complexity bounds for the distilled student, providing end-to-end theoretical guarantees for the proposed methods.

Combining MDKD with on-policy training. MDKD constructs divergence-aware training data using teacher-generated candidates. Combining MDKD with on-policy training, where the student generates partial sequences that are then completed via MDKD's

rejection sampling, could further reduce the distribution mismatch between training and inference. This hybrid approach would combine the exploration benefits of on-policy methods with the targeted supervision of divergence-aware data construction.

Cross-modal and cross-architectural distillation. The capacity gap is further exacerbated when teacher and student belong to different architectural families (e.g., distilling a Transformer teacher into a state-space model student). Extending the entropy-based curriculum of Annealing-KD and the continuation framework of Continuation-KD to cross-architectural settings, where the student may lack the structural mechanisms to represent certain patterns in the teacher's output, is a compelling direction for future investigation.

Quality-aware data distillation. The Efficient Citer framework demonstrated the effectiveness of data distillation for specialized tasks but relied on post-hoc cleaning to remove teacher errors. Future work could integrate quality estimation directly into the data distillation pipeline, using the student's own uncertainty estimates or external verification models to filter or reweight teacher-generated samples during dataset construction.

Alternative realizations of continuation optimization. In Chapter 5, we realized the continuation method by smoothing the objective with the hinge loss. However, the continuation framework is general and could potentially be realized through other smoothing mechanisms, such as Gaussian convolution of the loss surface, Moreau envelopes, or entropic regularization. Investigating these alternative realizations and their respective optimization landscapes is an interesting direction for future work.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[2] Eugene L Allgower and Kurt Georg. *Introduction to Numerical Continuation Methods*. SIAM, 2003.

[3] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, pages 233–242, 2017.

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[5] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in Neural Information Processing Systems*, 27, 2014.

[6] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[7] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 41–48, 2009.

[9] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

[10] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439, 2020.

[11] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[12] Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022.

[13] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[14] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[15] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[16] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[17] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11933–11942, 2022.

[18] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7028–7036, 2021.

[19] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[20] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5008–5017, 2021.

[21] Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts, 2026. arXiv:2601.22156.

[22] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online, August 2021. Association for Computational Linguistics.

[23] Zhihao Chi, Tu Zheng, Hengjia Li, Zheng Yang, Boxi Wu, Binbin Lin, and Deng Cai. NormKD: Normalized logits for knowledge distillation. *arXiv preprint arXiv:2308.00520*, 2023.

[24] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/, March 2023.

[25] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019.

[26] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[27] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[28] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy, July 2019. Association for Computational Linguistics.

[29] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[30] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. arXiv:2110.14168.

[31] Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

[32] Lahav Dabah and Tom Tirer. On temperature scaling and conformal prediction of deep classifiers. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 11813–11845. PMLR, 2025.

[33] D.F. Davidenko. On a new method of numerical solution of systems of nonlinear equations. *Dokl. Akad. Nauk SSSR*, 88:601–602, 1953.

[34] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35, 2022.

[35] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[37] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, 2023.

[38] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[39] Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. Distillation ≈ Early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network. *arXiv preprint arXiv:1910.01255*, 2019.

[40] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[41] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, 2019.

[42] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*, 2019.

[43] Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*, 2017.

[44] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.

[45] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[46] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[47] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*, 2023.

[48] Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[49] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, 2019.

[50] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[51] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[52] Caglar Gulcehre, Marcin Moczulski, Francesco Visin, and Yoshua Bengio. Mollifying networks. In *5th International Conference on Learning Representations*, 2017.

[53] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[54] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

[55] Md Akmal Haidar, Nithin Anchuri, Mehdi Rezagholizadeh, Abbas Ghaddar, Philippe Langlais, and Pascal Poupart. RAIL-KD: RAndom Intermediate Layer mapping for knowledge distillation. *arXiv preprint arXiv:2109.10164*, 2021.

[56] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[58] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

[59] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[60] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[61] Tao Huang, Yuan Zhang, Mingkai Zheng, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge diffusion for distillation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[62] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[63] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[64] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[65] Aref Jafari, Ivan Kobyzev, Mehdi Rezagholizadeh, Pascal Poupart, and Ali Ghodsi. Continuation KD: Improved knowledge distillation through the lens of continuation optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5260–5269, 2022.

[66] Aref Jafari, Mehdi Rezagholizadeh, and Ali Ghodsi. Improved knowledge distillation by utilizing backward pass knowledge in neural networks. *arXiv preprint arXiv:2301.12006*, 2023.

[67] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2493–2504, 2021.

[68] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[69] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[70] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1345–1354, 2019.

[71] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[72] Mehran Kardar. *Statistical Physics of Particles*. Cambridge University Press, 2007.

[73] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.

[74] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

[75] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[77] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2, 1989.

[78] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*. Citeseer, 2012.

[79] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, 2023.

[80] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[81] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4313–4324. PMLR, 2020.

[82] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need II: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[83] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

[84] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[85] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[86] Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise?, 2020. arXiv:2003.02819.

[87] Amir M Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. A comprehensive survey on knowledge distillation. *Transactions on Machine Learning Research*, 2025. arXiv:2503.12067.

[88] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

[89] Paul Micaelli and Amos Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[90] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2381–2391, 2018.

[91] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5191–5198, 2020.

[92] Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*, 2023.

[93] Hossein Mobahi and John Fisher III. A theoretical analysis of optimization by gaussian continuation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015.

[94] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*, 2023.

[95] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[96] OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt/, 2022.

[97] Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pages 4951–4960. PMLR, 2019.

[98] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3967–3976, 2019.

[99] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[100] Quora. First Quora dataset release: Question pairs. https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs, 2017.

[101] Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. arXiv:2412.15115.

[102] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, pages 5301–5310, 2019.

[103] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, 2018.

[104] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[105] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[106] Mehdi Rezagholizadeh, Aref Jafari, Puneeth S. M. Saladi, Pranav Sharma, Ali Saheb Pasand, and Ali Ghodsi. Pro-KD: Progressive distillation by following the footsteps of the teacher. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4714–4727, 2022.

[107] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[108] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

[109] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740, 2020.

[110] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[111] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

[112] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017. arXiv:1703.00810.

[113] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[114] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[115] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[116] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[117] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

[118] Marzieh Tahaei, Aref Jafari, Ahmad Rashid, David Alfonso-Hermelo, Khalil Bibi, Yimeng Wu, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. Efficient citer: Tuning large language models for enhanced answer quality and verification. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4443–4450, 2024.

[119] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

[120] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.

[121] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[122] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations (ICLR)*, 2020.

[123] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015. arXiv:1503.02406.

[124] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[125] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1365–1374, 2019.

[126] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

[127] Vladimir Vapnik. *Statistical learning theory*. John Wiley and Sons, 1998.

[128] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(61):2023–2049, 2015.

[129] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[130] Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. Softmax is not enough (for sharp size generalisation). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 61190–61211. PMLR, 2025.

[131] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[132] Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The mamba in the llama: Distilling and accelerating hybrid models, 2025. arXiv:2408.15237.

[133] Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard Zemel. Adversarial distillation of bayesian neural network posteriors. In *International Conference on Machine Learning (ICML)*, pages 5190–5199, 2018.

[134] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

[135] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[136] L.T. Watson. Theory of globally convergent probability-one homotopies for nonlinear programming. *SIAM Journal on Optimization*, 11(3):761–780, 2000.

[137] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[138] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[139] Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi Chen, Xinggang Wang, Hongyang Chao, and Han Hu. TinyCLIP: CLIP distillation via affinity mimicking and weight inheritance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21970–21980, 2023.

[140] Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024.

[141] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

[142] Shaotian Yan, Kaiyuan Liu, Chen Shen, Bing Wang, Sinan Fan, Jun Zhang, Yue Wu, Zheng Wang, and Jieping Ye. Distribution-aligned sequence distillation for superior long-CoT reasoning, 2026. arXiv:2601.09088.

[143] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[144] Mingyu Yang, Mehdi Rezagholizadeh, Guihong Li, Vikram Appia, and Emad Barsoum. Zebra-Llama: Towards extremely efficient hybrid models, 2025. arXiv:2505.17272.

[145] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. In *European Conference on Computer Vision (ECCV)*, pages 53–69, 2022.

[146] Zhaojian Yu, Xin Zhang, Ning Shang, Yangyu Huang, Can Xu, Yishujie Zhao, Wenxiang Hu, and Qiufeng Yin. WaveCoder: Widespread and versatile enhanced instruction tuning with refined data generation. *arXiv preprint arXiv:2312.14187*, 2024.

[147] Zony Yu, Yuqiao Wen, and Lili Mou. Revisiting intermediate-layer matching in knowledge distillation: Layer-selection strategy doesn't matter (much), 2025. arXiv:2502.04499.

[148] Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311*, 2023.

[149] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[150] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4791–4800, 2019.

[151] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

[152] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3713–3722, 2019.

[153] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Making a miracl: Multilingual information retrieval across a continuum of languages. *arXiv preprint arXiv:2210.09984*, 2022.

[154] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[155] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11953–11962, 2022.

[156] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

APPENDICES

Appendix A

Annealing Knowledge Distillation

A.1 Experimental parameters of the image classification tasks

In this section, we include more detail of our experimental settings of section 3.4.2 in the paper. For the baseline experiments, we used the same experimental setup as [91]. We performed two series of experiments based on ResNet and plain CNN neural networks on CIFAR-10 and CIFAR-100 datasets. Table A.1 illustrates the hyper-parameters used in these experiments. (BS = batch size, EP1= number of epochs in phase 1 (for the baselines, this is the number of training epochs), EP2 = number of epochs in phase 2, LR = learning rate, MO = momentum, WD = weight decay, τ_{max} = maximum temperature). Annealing-KD's two phases together use the same total number of training epochs as each baseline ($128 + 32 = 160$), so all methods in Table A.1 are compared under a matched training budget.

A.2 BERT Experiments

In these experiments, RoBERTa-large (24 layers) and DistilRoBERTa (6 layers) are used as the teacher and student models respectively. Also, RoBERTa-base (12-layer) is used as the teacher assistant for the TAKD baseline. For Annealing KD, we use the maximum temperature of 7 and the learning rate of 2e-5 for all the tasks. We trained the student model for 14 epochs in phase 1, and 6 epochs in phase 2. The phase-1 length follows the

Table A.1: Hyper-parameters of CIFAR-10 and CIFAR-100 experiments

Model	Type	Training method	BS	EP1	EP2	LR	MO	WD	$\tau_{\max}$
ResNet	Teacher(110)	from scratch	128	160	N/A	0.1	0.9	0.0001	N/A
	TA(20)	KD	128	160	N/A	0.1	0.9	0.0001	N/A
	Student(8)	from scratch	128	160	N/A	0.1	0.9	0.0001	N/A
	Student(8)	KD	128	160	N/A	0.1	0.9	0.0001	1
	Student(8)	TAKD	128	160	N/A	0.1	0.9	0.0001	1
	Student(8)	**Annealing KD (ours)**	128	128	32	0.1	0.9	0.0001	10
CNN	Teacher(10)	from scratch	128	160	N/A	0.1	0.9	0.0001	N/A
	TA(4)	KD	128	160	N/A	0.1	0.9	0.0001	N/A
	Student(2)	from scratch	128	160	N/A	0.1	0.9	0.0001	N/A
	Student(2)	KD	128	160	N/A	0.1	0.9	0.0001	1
	Student(2)	TAKD	128	160	N/A	0.1	0.9	0.0001	1
	Student(2)	**Annealing KD (ours)**	128	128	32	0.1	0.9	0.0001	10

maximum temperature ($\tau_{max} = 7$ gives 14 epochs and $\tau_{max} = 14$ gives 28), and phase 2 is 6 epochs throughout. Each baseline was trained for the sum of the two phases on the same task, so every method in these tables is compared under a matched training budget. Table A.3 illustrates the details of the hyper-parameters of the experiments. Also, Table A.6 illustrates the hyper-parameter values of BERT-small experiments in detail. Also, we did two ablation studies. In the first one, we tried to fine-tune the maximum temperature in annealing KD and check the performance improvement compared with using the general value of 7. As it is illustrated in Table A.4, we can get more improvement with selecting the maximum temperature parameter more carefully. The second ablation is about comparing the effect of mean square error and KL-divergence loss functions on the final results of the experiments when they are used as the loss function of the first phase. Table A.5 shows the results of this ablation.

Table A.2: Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE tasks

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Batch Size	32	32	32	32	32	32	32	32	32
Max Seq. Length	128	128	128	128	128	128	128	128	128
Vanilla KD Alpha	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Gradient Clipping	1	1	1	1	1	1	1	1	1

Table A.3: Model specific Hyper-parameters for DistilRoBERTa on GLUE tasks

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Phase 1 epochs	14	14	14	14	14	14	14	14	14
Phase 2 epochs	6	6	6	6	6	6	6	6	6
Baseline epochs	20	20	20	20	20	20	20	20	20
τ_{max}	7	7	7	7	7	7	7	7	7

Table A.4: Ablation on DistilRoberta Annealing KD with temperature tuning

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Avg
Annealing KD	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	56.33	81.42
+ **temp tuning**	61.67	73.64	**91.99**	**89.26**	**93.34**	**92**	**91.72**	**85.14/85.22**	56.33	**81.67**
(max temperature)	7	7	8	14	14	11	14	14	7	-

Table A.5: Ablation on DistilRoberta Annealing KD with different loss functions

KD Method and Loss	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Avg
Annealing KD, MSE	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	56.33	81.42
Annealing KD, KL-div	62.56	70.75	90.84	89.01	93	91.32	91.42	85/84.75	56.33	81.13

Table A.6: Model specific Hyper-parameters for BERT-Small on GLUE tasks

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	4e-5	4e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Phase 1 epochs	28	28	14	28	14	14	28	28	14
Phase 2 epochs	6	6	6	6	6	6	6	6	6
Baseline epochs	34	34	20	34	20	20	34	34	20
τ_{max}	14	14	7	14	7	7	14	14	7

A.3 Multi-Seed Study of Annealing-KD

The results of Annealing-KD reported in Chapter 3 (Tables 3.3 and 3.5) are single-run results. To examine how sensitive the comparison between methods is to random initialization, we additionally repeated the DistilRoBERTa and BERT-Small experiments on the GLUE dev set with 10 different random seeds for each student method (training from scratch, Vanilla KD, TAKD, and Annealing KD). The teacher and the teacher assistant (TA) are single runs in this study. Tables A.7 and A.8 report the mean and the standard deviation (smaller print after the $\pm$ sign) over the 10 seeds for every task. As in the main chapter, MNLI is reported as matched/mismatched accuracy, and Score is the unweighted mean of the nine displayed task values using MNLI matched accuracy; the standard deviation of the Score is computed over the seeds.

Table A.7: Multi-seed DistilRoBERTa results for Annealing KD on the GLUE dev set (10 seeds). For the student models (From scratch, Vanilla KD, TAKD, Annealing KD), each cell reports the mean $\pm$ standard deviation over 10 random seeds; the Teacher and TA rows are single runs. MNLI is reported as matched/mismatched accuracy. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy (MNLI mismatched is reported but not aggregated). Best student result per column in bold.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI-m/mm	WNLI	Score
Teacher	66.52	88.45	93.91	91.95	96.10	94.71	91.79	90.52/90.55	56.34	85.59
TA	61.07	80.51	92.39	90.54	94.50	93.26	92.20	88.05/87.67	56.34	83.21
From scratch	60.69$\pm$0.90	69.17$\pm$2.33	89.96$\pm$0.97	88.57$\pm$0.20	92.27$\pm$0.29	91.33$\pm$0.30	91.43$\pm$0.07	84.08$\pm$0.10 / 83.85$\pm$0.17	56.34$\pm$0.00	80.43$\pm$0.29
Vanilla KD	60.38$\pm$0.22	69.24$\pm$2.27	89.92$\pm$0.87	88.41$\pm$0.16	92.78$\pm$0.42	91.56$\pm$0.22	91.68$\pm$0.06	84.29$\pm$0.15 / 83.97$\pm$0.05	56.34$\pm$0.00	80.51$\pm$0.28
TAKD	61.06$\pm$0.89	69.39$\pm$1.64	90.16$\pm$0.93	88.47$\pm$0.22	92.80$\pm$0.19	91.76$\pm$0.38	**91.81$\pm$0.08**	84.61$\pm$0.21 / 84.51$\pm$0.11	56.34$\pm$0.00	80.71$\pm$0.22
Annealing KD	**61.60$\pm$0.34**	**70.25$\pm$0.91**	**90.93$\pm$0.58**	**88.84$\pm$0.21**	**93.26$\pm$0.10**	**92.11$\pm$0.26**	91.78$\pm$0.02	**85.74$\pm$0.17** / **85.27$\pm$0.18**	56.34$\pm$0.00	**81.21$\pm$0.10**

Findings. We summarize the main observations from Tables A.7 and A.8.

- **The ranking of the methods agrees with the single-run results.** For both students, Annealing-KD obtains the highest average Score, followed by TAKD, Vanilla KD, and training from scratch. For DistilRoBERTa, the mean Scores are 81.21 ± 0.10 (Annealing-KD), 80.71 ± 0.22 (TAKD), 80.51 ± 0.28 (Vanilla KD), and 80.43 ± 0.29 (from scratch). For BERT-Small they are 76.18 ± 0.36, 75.28 ± 0.39, 74.83 ± 0.69, and 74.56 ± 0.52, respectively.

Table A.8: Multi-seed BERT-Small results for Annealing KD on the GLUE dev set (10 seeds). For the student models (From scratch, Vanilla KD, TAKD, Annealing KD), each cell reports the mean ± standard deviation over 10 random seeds; the Teacher and TA rows are single runs. MNLI is reported as matched/mismatched accuracy. F1 scores are reported for MRPC, Pearson correlations for STS-B, Matthews correlation for CoLA, and accuracy scores for all other tasks. Score is the unweighted mean of the nine displayed task values, using MNLI matched accuracy (MNLI mismatched is reported but not aggregated). Best student result per column in bold.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI-m/mm	WNLI	Score
Teacher	62.48	73.65	90.82	89.92	93.35	92.51	91.70	86.71/85.68	56.34	81.94
TA	58.71	69.68	91.72	89.64	92.78	92.15	91.51	85.05/85.10	56.34	80.84
From scratch	40.80±1.86	64.48±0.79	87.14±0.60	87.35±0.18	88.97±0.36	86.89±0.11	90.11±0.02	78.58±0.18 / 78.29±0.33	46.76±5.49	74.56±0.52
Vanilla KD	40.93±1.36	65.05±0.47	**87.28±0.78**	87.29±0.19	88.99±0.39	87.53±0.15	90.47±0.06	78.91±0.21 / 78.52±0.12	47.04±6.03	74.83±0.69
TAKD	40.95±0.90	64.91±0.54	87.03±0.69	87.13±0.19	88.92±0.55	87.14±0.17	90.28±0.06	78.73±0.25 / 78.68±0.16	52.39±4.15	75.28±0.39
Annealing KD	**41.77±0.76**	**65.49±1.80**	86.97±0.71	**88.48±0.17**	**89.58±0.17**	**88.71±0.27**	**91.43±0.05**	**79.87±0.04** / **79.44±0.11**	**53.28±3.65**	**76.18±0.36**

- **Task-level behavior.** Annealing-KD has the best mean on most individual tasks. The exceptions are QQP for DistilRoBERTa, where TAKD is marginally ahead (91.81 versus 91.78), and MRPC for BERT-Small, where Vanilla KD is slightly ahead (87.28 versus 86.97). On several tasks (for example RTE and CoLA for DistilRoBERTa) the differences between methods are of the same order as the seed-to-seed standard deviation, so they should not be over-interpreted.

- **The gain is larger for the smaller student.** The gap in mean Score between Annealing-KD and Vanilla KD is 0.70 points for DistilRoBERTa and 1.35 points for BERT-Small. For BERT-Small a part of this gap comes from WNLI, where the variation across seeds is large for all methods (standard deviations between 3.65 and 6.03); excluding WNLI, the gap in the mean over the remaining eight tasks is about 0.73 points.

Appendix B

Progressive Knowledge Distillation

B.1 Why Does Pro-KD Work?

In this section, first we state stopping the training procedure of the teacher at early epochs will improve Knowledge Distillation. Then, we argue that our method will exploit this fact without requiring any effort on finding the epoch at which early stopping the teacher leads to the optimum Knowledge Distillation. Finally, we will conclude that training the student together with the teacher leads to an important advantage that provides the student with the opportunity of exploiting the knowledge of the optimum teacher.

In knowledge distillation, soft labels coming from a pre-trained teacher contains some so-called *dark knowledge* [39] which gives a signal about the relative differences between the probabilities of the classes.

It is shown by [25] and [39] that early stopping during training of the teacher network improves the performance of Knowledge Distillation significantly; in other words, there is an optimal epoch at which if teacher training is stopped, the resulting teacher will act as a better teacher compared to the ones stopped at earlier or later epochs. Authors of [25] have shown this fact through experiments based on performing KD by using pre-trained teacher networks stopped at different epochs, and they also have indicated that changing the temperature cannot compensate lost Dark Knowledge. Authors of [39] have provided mathematical justification showing that neural network learns more useful information faster. They have used the methods proposed in papers [40, 97, 81], and the concept of Neural Tangent Kernel introduced by [64] in order to reach an asymptotic conclusion. This conclusion states that for infinite wide neural networks, gradient descend algorithm

searches over different direction with different pace; that is to say, the projection of the loss function in different eigenspaces evolves with different rate. Those rates can be calculated as follows [39]:

$$\begin{aligned}\langle (u_{t+1} - y), e_i \rangle &= \langle (I - \eta H^*)(u_t - y), e_i \rangle \\ &= \langle (u_t - y), (I - \eta H^*) e_i \rangle \\ &= (1 - \eta \lambda_i) \langle (u_t - y), e_i \rangle\end{aligned} \tag{B.1}$$

where u_t is the output of model, y is the true label, e_i is the eigenvector corresponding to the i^{th} eigenvalue(λ_i) of the static Gram matrix(H^*) of the network. The gram matrix of neural networks is a function of time in general; however, in infinitely wide neural networks, this matrix will be static and called Neural Tangent Kernel [64]. This static $n \times n$ matrix, n is the number of training samples, can be calculated as follows [39]:

$$H^* = \left(\langle \frac{\partial f(\theta, x_i)}{\partial \theta} . \frac{\partial f(\theta, x_j)}{\partial \theta} \rangle \right)_{i,j} \tag{B.2}$$

where $f(\theta, \cdot)$ is the output of our model, θ are the parameters of the neural network, and x_i is the i^{th} data sample.

Arguments provided by [39] state that a neural network learns more useful information faster than non-principal pieces of information about the input samples; however, they do not explain why continuing to train the teacher would decrease the level of that "dark knowledge". The information-bottleneck view of deep learning [123] motivated information-plane analyses that report [112] an initial fitting phase followed by a compression-like phase in some architectures and optimization settings, in which the mutual information between hidden representations and the input first rises and then falls while the information about the labels keeps rising. If such a compression phase occurs, one would expect the teacher's outputs to become more confident and its inter-class "dark knowledge" to diminish over the later epochs. However, subsequent work has shown that these phases are *not universal*: they depend on the activation function and the estimator used, and are absent in several standard settings [111]. We therefore treat the relationship between training time, teacher confidence, and distillation utility as an empirical hypothesis rather than an established mechanism. The empirical evidence for it in our setting is Table 4.1, which shows that the teacher checkpoint yielding the strongest student varies by task and by student architecture and is generally not the teacher's own best-performing checkpoint.

Pro-KD exploits this by exposing a single evolving student to a sequence of teacher checkpoints, checkpointing the student throughout. This removes the need to train a separate student from scratch for every candidate teacher checkpoint. It does *not* guarantee

that any Pro-KD student checkpoint is identical to a student trained only from the empirically optimal teacher checkpoint: the student state at step j depends on all preceding teacher checkpoints, so Pro-KD's advantage must be established by direct comparison with checkpoint-search baselines (as in the experiments of the main chapter) rather than by an equivalence argument.

In contrast with usual KD methods, a pre-trained teacher is not required by Pro-KD; instead, the teacher is trained alongside the student.

B.2 Hyper-parameters

In this section, we summarize the hyper-parameters used in our experiments.

Table B.1: Model specific Hyper-parameters for BERT-Small on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	5e-5	5e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
N (Teacher Epochs)	7	7	7	7	7	7	7	7	7
τ_{max}	7	7	7	7	7	7	7	7	7
n (Phase 2 Epochs)	10	10	10	10	10	10	10	10	10

Table B.2: Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Batch Size	32	32	32	32	32	32	32	32	32
Max Seq. Length	128	128	128	128	128	128	128	128	128
Vanilla KD Alpha	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Gradient Clipping	1	1	1	1	1	1	1	1	1

Table B.3: Model specific Hyper-parameters for DistilRoBERTa on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
N (Teacher Epochs)	5	5	5	5	5	5	5	5	5
τ_{max}	5	5	5	5	5	5	5	5	5
n (Phase 2 epochs)	10	10	10	10	10	10	10	10	10

Table B.4: Model specific Hyper-parameters for DistilRoBERTa on SQuAD

Hyper-parameter	SQuAD 1.1/2.0	SQuAD Teacher
Learning Rate	3e-5	1.5e-5
Batch Size	12	12
Max Seq. Length	384	384
Doc Stride	128	128
Weight Decay	-	0.01
N (Teacher Epochs)	3	3
τ_{max}	3	-
n (Phase 2 Epochs)	6	-

Table B.5: Model specific Hyper-parameters for BERT-Small on SQuAD

Hyper-parameter	SQuAD 1.1/2.0	SQuAD Teacher
Learning Rate	3e-5	3e-5
Batch Size	12	12
Max Seq. Length	384	384
Doc Stride	128	128
N (Teacher Epochs)	3	3
τ_{max}	3	-
n (Phase 2 Epochs)	5	-

Appendix C

Continuation Knowledge Distillation

C.1 Derivation of the Annealed MSE Hessian

This section details the formal derivation of the Hessian matrix for the annealed Mean Squared Error (MSE) objective utilized in the Continuation-KD framework (Chapter 5). This derivation culminates in the exact additive decomposition stated in Theorem 5.3, which cleanly separates the student-only curvature from the teacher-induced curvature.

Step 1: Define the Objective Function
Let $z_s(x;\theta) \in \mathbb{R}^K$ denote the pre-softmax logits of the student network parameterized by weights $\theta \in \mathbb{R}^d$. Let $z_t(x) \in \mathbb{R}^K$ denote the static logits of the pre-trained teacher network. The annealed MSE loss over K classes is defined as:

$$\mathcal{L}_{\mathcal{T}}(\theta) = \frac{1}{2}\sum_{k=1}^{K}\left(z_{s,k}(x;\theta) - \Phi(\mathcal{T})z_{t,k}(x)\right)^2 \tag{C.1}$$

Step 2: Compute the First Gradient
To find the gradient vector with respect to the student's weights θ, we apply the chain rule. Because the teacher's logits $z_{t,k}(x)$ and the scaling factor $\Phi(\mathcal{T})$ are static with respect to θ, their derivatives are zero. Taking the derivative of the squared term yields:

$$\nabla_\theta \mathcal{L}_{\mathcal{T}}(\theta) = \sum_{k=1}^{K}\left(z_{s,k}(x;\theta) - \Phi(\mathcal{T})z_{t,k}(x)\right)\nabla_\theta z_{s,k}(x;\theta) \tag{C.2}$$

where $\nabla_\theta z_{s,k}(x;\theta)$ is a $d\times 1$ column vector containing the partial derivatives of the k-th logit with respect to every parameter in θ. In compact notation using the Jacobian $J_\theta \in \mathbb{R}^{K\times d}$:

$$\nabla_\theta \mathcal{L}_\mathcal{T}(\theta) = J_\theta^T (z_s - \Phi(\mathcal{T})\, z_t) \tag{C.3}$$

Step 3: Compute the Second Gradient (Hessian)
The Hessian matrix $\nabla^2 \mathcal{L}_\mathcal{T}(\theta) \in \mathbb{R}^{d\times d}$ is obtained by taking the derivative of the first gradient with respect to θ. We apply the product rule, $\nabla(u\cdot v) = v(\nabla u)^T + u(\nabla v)$, to the terms inside the summation.

Let $u_k = (z_{s,k}(x;\theta) - \Phi(\mathcal{T}) z_{t,k}(x))$ (a scalar residual) and $v_k = \nabla_\theta z_{s,k}(x;\theta)$ (the $d\times 1$ gradient vector). The two components of the product rule are as follows. The derivative of the scalar residual is $\nabla_\theta u_k = \nabla_\theta z_{s,k}(x;\theta)$, so the first term is the outer product of the logit gradient with itself,

$$v_k\, (\nabla_\theta u_k)^T = \nabla_\theta z_{s,k}(x;\theta)\, \big(\nabla_\theta z_{s,k}(x;\theta)\big)^T. \tag{C.4}$$

The derivative of the gradient vector is the Hessian of the k-th logit, $\nabla_\theta v_k = \nabla^2_\theta z_{s,k}(x;\theta)$, so the second term is

$$u_k\, \nabla_\theta v_k = \big(z_{s,k}(x;\theta) - \Phi(\mathcal{T})\, z_{t,k}(x)\big)\, \nabla^2_\theta z_{s,k}(x;\theta). \tag{C.5}$$

Substituting these components back into the summation yields the full Hessian:

$$\nabla^2 \mathcal{L}_\mathcal{T}(\theta) = \sum_{k=1}^{K} \Big[\nabla_\theta z_{s,k}(x;\theta)\, (\nabla_\theta z_{s,k}(x;\theta))^T + (z_{s,k}(x;\theta) - \Phi(\mathcal{T}) z_{t,k}(x))\, \nabla^2 z_{s,k}(x;\theta)\Big] \tag{C.6}$$

Step 4: Conversion to Matrix Notation
To simplify Equation C.6, we express the first term inside the summation using the Jacobian matrix of the student network, $J_\theta \in \mathbb{R}^{K\times d}$. The k-th row of J_θ contains the gradient of the k-th logit, $(\nabla_\theta z_{s,k}(x;\theta))^T$.

By the definition of matrix multiplication, the product $J_\theta^T J_\theta$ is exactly equivalent to the sum of the outer products of its rows:

$$J_\theta^T J_\theta = \sum_{k=1}^{K} \nabla_\theta z_{s,k}(x;\theta)\, (\nabla_\theta z_{s,k}(x;\theta))^T \tag{C.7}$$

Substituting this Jacobian product back into Equation C.6 yields the compact form:

$$\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) = J_\theta^T J_\theta + \sum_{k=1}^{K} \left(z_{s,k}(x;\theta) - \Phi(\mathcal{T}) z_{t,k}(x) \right) \nabla^2 z_{s,k}(x;\theta) \tag{C.8}$$

Step 5: Exact Additive Decomposition
We now factor (C.8) into a student-only term and a teacher–student coupling term by distributing the residual in the second summand:

$$\begin{aligned} \nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) &= J_\theta^T J_\theta + \sum_{k=1}^{K} z_{s,k}(x;\theta) \nabla^2 z_{s,k}(x;\theta) - \Phi(\mathcal{T}) \sum_{k=1}^{K} z_{t,k}(x) \nabla^2 z_{s,k}(x;\theta) \\ &= \underbrace{\left(J_\theta^T J_\theta + \sum_{k=1}^{K} z_{s,k} \nabla^2 z_{s,k} \right)}_{H_S(\theta)} - \Phi(\mathcal{T}) \underbrace{\sum_{k=1}^{K} z_{t,k} \nabla^2 z_{s,k}}_{C(\theta)} \end{aligned} \tag{C.9}$$

This is the exact decomposition stated in Theorem 5.3: $\nabla^2 \mathcal{L}_{\mathcal{T}}(\theta) = H_S(\theta) - \Phi(\mathcal{T}) \cdot C(\theta)$, where $H_S(\theta)$ is the student-only Hessian (independent of Φ) and $C(\theta)$ is the teacher–student coupling matrix (also independent of Φ). The teacher-dependent term is multiplied by $\Phi(\mathcal{T})$, so the Hessian *matrix* is affine in $\Phi(\mathcal{T})$ at a fixed θ. This constrains the matrix but not its spectral norm: because $C(\theta)$ need not be positive or negative semidefinite, and its eigenspaces need not align with those of $H_S(\theta)$, the decomposition implies neither monotonicity nor linearity of the spectral norm as $\Phi(\mathcal{T})$ is varied. The decomposition is used as a structural description of the objective (Remark 5.4). Two regularity conditions are implicit in the derivation above and are assumed wherever the decomposition is used: $z_s(x;\theta)$ is twice continuously differentiable in θ on a neighbourhood of the evaluation point, and x, $z_t(x)$, and $\Phi(\mathcal{T})$ are held fixed under differentiation. For networks with piecewise-linear activations the required second derivatives exist almost everywhere but not at activation boundaries, where a classical Hessian need not exist.

Appendix D

Data distillation: Efficient Citer

D.1 Training details

To fine-tune the FIDCiter (FID-style Flan-T5) model, we set the batch size to 8 and the input (per encoder) / output sequence length to 256/1800 tokens, respectively. We set the learning rate to 5e-5 with a linear decay scheduler and perform warmup during training. We train the model using Adam for 6 epochs, with a total of 5 provided contexts used in the training process. Training on 8 V100 GPUs took approximately 60 hours.

To fine-tune the LLaMACiter model, we set the batch size to 512 and the sequence length to 1024 tokens. The learning rate was set to 3e-5 with a linear decay scheduler and warmup, as for Flan-T5. The AdamW optimizer was used and the model was trained for 5 epochs. Five contexts were provided for each sample during training, on 8 V100 GPUs for approximately 16 hours.

D.2 Inference parameters

At inference we use the Hugging Face `generate` function. For FIDCiter we set `num_beams` = 5 and `top_p` = 0.95; for LLaMACiter we set `num_beams` = 4 and `top_k` = 40.

D.3 Data collection prompt

For collecting the data samples, we passed the queries and the passages in the MS MARCO dataset as well as the gold answer of each sample and asked the model to generate a new answer by considering the given gold answer and mentioning the passages as the references to the generated parts of the answer. Here you can find the used prompt:
prompt: I will give a question and several context texts about the question as well as the gold answer to the question. By considering the given contexts, the gold answer regenerates an answer to the question. Also, mention the reference of parts of your answer based on the given contexts within brackets [] in IEEE format. Your answer must be similar to the given gold answer.

D.4 Automatic evaluation metric from ALCE Benchmark

There are a few automatic evaluation methods for attributed question answering tasks proposed in the literature [148, 47] In this work, for the automatic evaluation we use correctness, citation recall and precision on ELI5 and ASQA datasets from ALCE [47] benchmark. Below is a brief description of each metric.

- **Correctness:** For ELI5 dataset, InstructGPT is used to generate three "sub-claims," and a natural language inference (NLI) model called TRUE is employed to check if the model output entails these sub-claims. This method, known as claim recall, proves to be a more accurate measure of correctness compared to existing metrics.
- **Citation quality:** Citation quality is assessed using two metrics: citation recall and citation precision. Citation recall determines if the output is fully supported by the cited passages, while citation precision identifies any irrelevant citations. ALCE employs the TRUE10 model, fine-tuned on NLI datasets, to automatically determine if the cited passages entail the model generation. The human evaluation demonstrates a strong correlation with the automatic evaluation procedure. For citation recall, each statement's recall is calculated by checking if there is at least one citation and if the concatenated passages entail the statement, using an NLI model.

D.5 Example of our human evaluation

Tables D.1, D.2, and D.3, demonstrate an example of the human evaluation process for a data sample from the MIRACL dataset on ChatGPT, FLAN-T5, and LLaMA models. The sample has a query and five corresponding passages. The evaluators divided the answer of each model into cited sections and evaluated informativeness, citation quality and logic of each section separately. They scrutinized each section within the cited passages to determine the citation quality. The final score was calculated by dividing the number of passages that accurately cited the section by the total number of passages referencing it. You can find the query and the list of the passages of this sample below:

Query: What is the wavelength of visible light? Passages: [1] The visible spectrum is the portion of the electromagnetic spectrum that is visible to the human eye. Electromagnetic radiation in this range of wavelengths is called visible light or simply light. A typical human eye will respond to wavelengths from about 380 to 740 nanometers. In terms of frequency, this corresponds to a band in the vicinity of 430–770 THz.

[2] Electromagnetic radiation with a wavelength between 380 nm and 760 nm (400–790 terahertz) is detected by the human eye and perceived as visible light. Other wavelengths, especially near infrared (longer than 760 nm) and ultraviolet (shorter than 380 nm) are also sometimes referred to as light, especially when the visibility to humans is not relevant. White light is a combination of lights of different wavelengths in the visible spectrum. Passing white light through a prism splits it up into the several colors of light observed in the visible spectrum between 400 nm and 780 nm.

[3] Light, or visible light, is a very narrow range of electromagnetic radiation of a wavelength that is visible to the human eye, or 380–750 nm which equates to a frequency range of 790 to 400 THz respectively. More broadly, physicists use the term "light" to mean electromagnetic radiation of all wavelengths, whether visible or not.

[4] Light is electromagnetic radiation within a certain portion of the electromagnetic spectrum. The word usually refers to visible light, which is the visible spectrum that is visible to the human eye and is responsible for the sense of sight. Visible light is usually defined as having wavelengths in the range of 400–700 nanometers (nm), or 4.00×10^{-7} to 7.00×10^{-7} m, between the infrared (with longer wavelengths) and the ultraviolet (with shorter wavelengths). This wavelength means a frequency range of roughly 430–750 terahertz (THz). The main source of light on Earth is the Sun. Sunlight provides the energy that green plants use to create sugars mostly in the form of starches, which release energy into the living things that digest them. This process of photosynthesis provides virtually all the energy used by living things. Historically, another important source of light for humans

has been fire, from ancient campfires to modern kerosene lamps. With the development of electric lights and power systems, electric lighting has effectively replaced firelight. Some species of animals generate their own light, a process called bioluminescence. For example, fireflies use light to locate mates, and vampire squids use it to hide themselves from prey.

[5] Blue Light is a range of the visible light spectrum, defined as having a wavelength between 400–495 nm. This short wavelength means that blue light is a type of high-energy visible light, defined as having a wavelength between 400 and 450 nm. Violet, indigo, and some blue-green light are other types of high-energy visible light.

Segments	Informativeness	Fluency	Citation quality
The wavelength of visible light ranges from about 380 to 740 nanometers, with frequencies in the vicinity of 430-770 THz[1][2][4].	1	1	1
This electromagnetic radiation is the only range that our human eyes can detect[1][3].	0	1	0.5
Specifically, light is defined as electromagnetic radiation within the wavelength range of around 400 to 700 nanometers or frequency range of roughly 430-750 THz[4].	1	1	1

Table D.1: ChatGPT output of the sample example. The answer to the given query is broken into 3 pieces

Segments	Informativeness	Fluency	Citation quality
The wavelength of visible light is between 380 nm and 760 nm.[2][4]	1	1	0.5

Table D.2: FLAN-T5 output of the sample example. The answer to the given query is broken into 1 section

Segments	Informativeness	Fluency	Citation quality
The wavelength of visible light is between 380 nm and 760 nm or 400–790 THz.[2][1]	1	1	0.5

Table D.3: LLaMA output of the sample example. The answer to the given query is broken into 1 section

Appendix E

Maximum Divergence Knowledge Distillation

E.1 Proofs of the Distributional Characterization

This appendix gives the full proofs of Theorem 7.1 and Corollary 7.3 from Chapter 7. Throughout, T and S denote the temperature-scaled, nucleus-filtered teacher and the correspondingly renormalized student distributions over their common support $\mathcal{V}_0 \subseteq \mathcal{V}$, with $T(v) > 0$ for all $v \in \mathcal{V}_0$. The single-proposal acceptance probability is $\alpha(v) = [\,1 - S(v)/T(v)\,]_+$, where $[z]_+ = \max(0, z)$.

Proof of Theorem 7.1

Step 1: Joint probability of proposal-and-acceptance. A token v is proposed with probability $T(v)$ and, conditioned on being proposed, accepted with probability $\alpha(v)$. Hence the joint probability of proposing *and* accepting v is

$$T(v)\,\alpha(v) = T(v)\Big[\,1 - \tfrac{S(v)}{T(v)}\,\Big]_+ = [\,T(v) - S(v)\,]_+. \tag{E.1}$$

The last equality holds in both regimes by direct verification: if $T(v) > S(v)$ then $\alpha(v) = 1 - S(v)/T(v)$ and $T(v)\alpha(v) = T(v) - S(v) = [\,T(v) - S(v)\,]_+$; if $T(v) \le S(v)$ then $\alpha(v) = 0$ and both sides are $0 = [\,T(v) - S(v)\,]_+$.

Step 2: Acceptance probability equals total variation (claim 1). Summing Eq. (E.1) over the support,

$$\Pr[\text{accept}] = \sum_{v \in \mathcal{V}_0} T(v)\,\alpha(v) = \sum_{v \in \mathcal{V}_0} [\,T(v) - S(v)\,]_+. \tag{E.2}$$

Because T and S are probability distributions on $\mathcal{V}_0$, we have $\sum_v (T(v) - S(v)) = 0$, so the positive and negative parts of $T - S$ have equal total mass:

$$\sum_v [\,T(v) - S(v)\,]_+ = \sum_v [\,S(v) - T(v)\,]_+ = \tfrac{1}{2} \sum_v |T(v) - S(v)| = \mathrm{TV}(T, S), \tag{E.3}$$

which establishes Eq. (7.3) and proves claim 1.

Step 3: Conditional law given acceptance (claim 2). For $\mathrm{TV}(T, S) > 0$, the conditional probability of selecting v given acceptance is the joint probability normalized by $\Pr[\text{accept}]$:

$$\Pr[v \mid \text{accept}] = \frac{T(v)\,\alpha(v)}{\Pr[\text{accept}]} = \frac{[\,T(v) - S(v)\,]_+}{\mathrm{TV}(T, S)}, \tag{E.4}$$

which is exactly the normalized positive residual $R(v)$.

Step 4: Conditional law given rejection (claim 3). The joint probability of proposing v and *rejecting* it is

$$T(v)\,(1 - \alpha(v)) = \min\{T(v), S(v)\}, \tag{E.5}$$

again by case analysis: if $T(v) > S(v)$ then $1 - \alpha(v) = S(v)/T(v)$ and the left side is $T(v) \cdot S(v)/T(v) = S(v) = \min\{T(v), S(v)\}$; if $T(v) \le S(v)$ then $1 - \alpha(v) = 1$ and the left side is $T(v) = \min\{T(v), S(v)\}$. Summing Eq. (E.5) gives

$$\Pr[\text{reject}] = \sum_v \min\{T(v), S(v)\} = 1 - \mathrm{TV}(T, S), \tag{E.6}$$

using the identity $\sum_v \min\{T(v), S(v)\} = 1 - \mathrm{TV}(T, S)$. For $\mathrm{TV}(T, S) < 1$, normalizing Eq. (E.5) yields

$$\Pr[v \mid \text{reject}] = \frac{\min\{T(v), S(v)\}}{1 - \mathrm{TV}(T, S)}, \tag{E.7}$$

which proves claim 3 and completes the proof. □

Proof of Corollary 7.3

Let $\delta = \mathrm{TV}(T, S)$. The m candidates $\hat{x}_1, \ldots, \hat{x}_m$ are drawn i.i.d. from T, and each is independently subjected to the stochastic acceptance test of Theorem 7.1, so each is accepted with marginal probability δ independently of the others. Therefore the probability that *no* candidate is accepted is

$$\Pr[\text{no acceptance}] = (1-\delta)^m, \tag{E.8}$$

and the probability that at least one candidate is accepted is $1-(1-\delta)^m$.

Condition on the event that at least one candidate is accepted. The algorithm appends the *first* candidate (in index order) that passes its test. By claim 2 of Theorem 7.1, every candidate that passes the test has the same conditional distribution $R(v) = [\,T(v) - S(v)\,]_+/\delta$; since the candidates are exchangeable and identically distributed, the index of the first acceptance is independent of the token identity, and the first accepted token is therefore distributed as R regardless of which index j first satisfies $r_j < \alpha_j$.

On the complementary event (probability $(1-\delta)^m$), the deterministic fallback selects $\arg\max_j \alpha_j$ among the m candidates, inducing some distribution F_m over $\mathcal{V}_0$ that depends on m, T, and S. Explicitly, writing $j^* = \arg\max_{j\in[m]} \alpha(\hat{x}_j)$,

$$F_m(v) \;=\; \frac{\mathbb{E}_{\hat{x}_{1:m}\sim T^{\otimes m}}\Big[\mathbf{1}\{\hat{x}_{j^*} = v\}\;\prod_{j=1}^{m}\big(1-\alpha(\hat{x}_j)\big)\Big]}{(1-\delta)^m}, \tag{E.9}$$

a finite sum over candidate tuples for small m. The factor $\prod_j(1-\alpha(\hat{x}_j))$ is the probability that the drawn tuple is rejected in its entirety, and it must not be omitted: F_m is the arg max law *conditioned on* total rejection, not the unconditional arg max law over m i.i.d. draws from T. Because the factor down-weights tuples containing high-α candidates, the two laws can differ substantially, and they bias F_m in opposite directions (Remark 7.4). Applying the law of total probability over the two mutually exclusive events,

$$P_{\mathrm{MDKD}}(v) = \big(1-(1-\delta)^m\big)\,R(v) + (1-\delta)^m\,F_m(v), \tag{E.10}$$

which is Eq. (7.6). □

Remark. The mixture weight $1-(1-\delta)^m$ is monotonically increasing in both the per-step total variation δ and the candidate count m: larger local teacher–student disagreement, or a larger candidate pool, increases the probability that the exact-residual stochastic branch (rather than the greedy fallback) determines the appended token. As $\delta \to 0$ (student matches teacher), the weight vanishes and the fallback dominates; as $\delta \to 1$, a single candidate suffices to trigger stochastic acceptance with high probability.

E.2 Acceptance-Rate Diagnostics

This appendix collects two diagnostics supporting Section 7.5.5.

E.2.1 Acceptance tracks divergence

To confirm that the acceptance rate reads out teacher–student divergence, we fix the teacher (Qwen2.5-14B-Instruct), run Algorithm 3 as an inference-only diagnostic on held-out GSM8K, and vary only the student. The diagnostic uses $m = 10$ candidates, $p_{\text{nuc}} = 0.95$, and $N = 512$ problems. We run at $\tau = 1$ because Algorithm 3 temperature-scales the teacher alone, so $\tau = 1$ is the setting at which a student with zero divergence yields exactly zero acceptance, giving a clean lower boundary. Note that the candidate count differs from the $m = 5$ of the deployed configuration, so the rates below are not directly comparable with those quoted in the main text.

Student	**Stochastic accept (%)**	**Fallback (%)**
Qwen2.5-14B-Instruct (identical to teacher)	0.00	100.00
Qwen2.5-1.5B-Instruct	15.35	84.65
Qwen2.5-1.5B (random init.)	100.00	0.00

Acceptance moves monotonically from 0.00% to 100% as divergence grows, the ordering Theorem 7.1 predicts. The two endpoints have different status. The 0.00% row is exact and deterministic rather than an empirical frequency: with the student identical to the teacher and $\tau = 1$ we have $S_\tau = T_\tau$, hence $\alpha \equiv 0$ and no candidate can ever be accepted; the run logged more than 153,000 token decisions, none with residual mass to accept. The 100% row is empirical saturation, since $(1 - \delta)^m > 0$ whenever both distributions are strictly positive on the shared support, so a rate of exactly 100% is unattainable in theory for finitely many candidates. The sweep is a consistency check rather than an independent measurement of divergence: the filtered divergences of the three students are not known separately, so what it shows is that the rate orders the students as expected, with the lower endpoint fixed exactly by theory.

E.2.2 A low acceptance rate can still re-route the trajectory

The example below was surfaced by scanning the GSM8K test set for cases in which the teacher and student answers agree while the MDKD answer differs; it is illustrative rather

than representative. Openings only; the accept/reject trace is `AFAFAAFFFFFFFFFAAFFFFFFFFFF`... (15.8% accepted, front-loaded).

Source	Opening
Question	"Jeff and Brad divide \$100; Jeff gets 4 times as much as Brad. How much does Jeff get?"
Teacher	"Let's denote the amount Brad gets as x dollars. Jeff gets 4 times as much, so Jeff gets ..."
Student	"Let's denote the amount Brad gets as x. Jeff gets 4 times as much, so Jeff gets ..."
MDKD	"To solve the problem of dividing 100 dollars between Jeff and Brad, where Jeff gets 4 times as much as Brad, follow these steps ..."

Teacher and student produce near-identical, both-correct solutions, so the student already covers the teacher's path, yet MDKD accepts only 15.8% of tokens, with its accepts clustered at the start of the trace, and produces an entirely different but still-correct derivation. The accepts, though few, determine the path.

E.3 Is the Fallback Branch Divergence-Targeting?

Remark 7.4 states that in the small-δ regime both branches of the mixture push generation toward the residual $[T - S]_+$, but that this is not a structural property. This appendix records the check behind that qualification. Write

$$\mathbb{E}_{F_m}[\alpha] \;=\; \sum_{v \in \mathcal{V}_0} F_m(v)\,\alpha(v), \qquad \text{against} \qquad \mathbb{E}_T[\alpha] \;=\; \sum_{v \in \mathcal{V}_0} T(v)\,\alpha(v) \;=\; \delta. \tag{E.11}$$

The fallback is more strongly directed at the residual than plain teacher sampling exactly when $\mathbb{E}_{F_m}[\alpha] > \delta$. Since α is supported on $\{v : T(v) > S(v)\}$ and increases with the shortfall $1 - S(v)/T(v)$, this is the natural scalar test. Throughout, F_m is computed from Eq. (E.9) by exact enumeration over all $|\mathcal{V}_0|^m$ candidate tuples, so no sampling error enters.

The claim fails in general. A two-token counterexample settles it in closed form. Let $\mathcal{V}_0 = \{a, b\}$ with

$$T = \left(\tfrac{1}{2}, \tfrac{1}{2}\right), \qquad S = \left(\tfrac{9}{10}, \tfrac{1}{10}\right), \qquad \text{so} \qquad \alpha(a) = 0, \quad \alpha(b) = \tfrac{4}{5}, \quad \delta = \tfrac{2}{5}.$$

Take $m = 2$. Each of the four ordered tuples has probability $\frac{1}{4}$, and the probability that a tuple is rejected in its entirety is $\prod_j (1 - \alpha(\hat{x}_j))$, namely 1 for (a, a), $\frac{1}{5}$ for (a, b) and (b, a), and $\frac{1}{25}$ for (b, b). The arg max returns a only on (a, a). Hence the normalizing constant is $\frac{1}{4}\big(1 + \frac{1}{5} + \frac{1}{5} + \frac{1}{25}\big) = \frac{9}{25} = (1 - \delta)^2$, as required, and

$$F_2 = \Big(\tfrac{25}{36},\ \tfrac{11}{36}\Big), \qquad \mathbb{E}_{F_2}[\alpha] = \tfrac{11}{36} \cdot \tfrac{4}{5} = \tfrac{11}{45} \approx 0.244 \ < \ \delta = \tfrac{2}{5}.$$

The fallback is therefore *less* concentrated on the residual than plain teacher sampling. The contrast with the unconditional arg max law is instructive: that law is $\big(\frac{1}{4}, \frac{3}{4}\big)$, giving $\mathbb{E}[\alpha] = \frac{3}{5} > \delta$. The two laws fall on opposite sides of δ, which is why the conditioning factor in Eq. (E.9) cannot be dropped.

What the analysis does and does not establish. The counterexample shows that no general statement is available. It does not follow that the opposite holds for small δ: expanding the conditioning factor gives $\prod_{j=1}^{m}(1 - \alpha_j) = 1 - \sum_{j=1}^{m} \alpha_j + O(\|\alpha\|^2)$, so its departure from one is *first* order in the acceptance scores, not second, and a small $\delta = \mathbb{E}_T[\alpha]$ does not imply that every $\alpha(v)$ is small. We therefore make no claim about the direction of F_m's bias in the deployed regime. The question is in any case empirically moot: the fallback-policy ablation in Section 7.5.5 shows that replacing the arg max with a uniform random choice or a plain teacher sample changes accuracy by at most a point or two, while disabling the stochastic branch costs about five. Whether F_m favors the residual is therefore not what the method depends on.

The counterexample is verified in exact rational arithmetic by `code/fallback_check.py`, accompanying this thesis.