Dialectical Chain Distillation: Transferring Dialectical Reasoning from Teacher–Student Interactions to Small Language Models

Anonymous Submission

Abstract

While Large Language Models (LLMs) have become widely used in natural language processing, their deployment remains challenging in resource-constrained environments due to substantial computational requirements. Model compression techniques such as pruning, quantization, and knowledge distillation are commonly employed to reduce resource burden. However, these methods often compromise model robustness and multi-step reasoning ability. In this paper, we propose **Dialectical Chain Distillation (DCD)**, a novel knowledge distillation framework that enhances the reasoning capability of LLMs through structured teacher-student interactions. DCD constructs dialectical reasoning chains involving drafting, deep reasoning, verification, and finalization, which provide informative and interpretable supervision for training student models. Experimental results on AIME 24, GSM8K and GPQA Diamond demonstrate that DCD improves both reasoning accuracy and robustness compared to standard Chain-of-Thought distillation methods, highlighting its effectiveness in producing more reliable compressed LLMs.

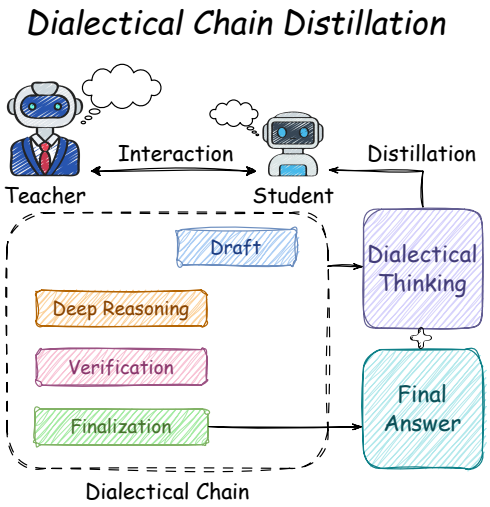


Figure 1: Overview of the proposed **Dialectical Chain Distillation (DCD)** framework. A teacher model engages in structured interactions with the student to generate dialectical reasoning chains comprising drafting, deep reasoning, verification, and finalization steps. These are distilled to enhance the student's internal reasoning patterns, leading to both accurate predictions and more interpretable, robust behavior.

1 Introduction

Large Language Models (LLMs) have exhibited remarkable performance in various natural language processing tasks, including reasoning, question answering, and code generation. Despite their impressive capabilities, these models encounter significant deployment challenges due to high computational and memory demands, limiting their usability on edge devices, mobile platforms, and other low-latency, resource-constrained environments. To overcome these barriers, model compression techniques, such as pruning, quantization, low-rank adaptation, and knowledge distillation have become crucial in enabling practical applications of LLMs.

However, compression techniques often inadvertently reduce model robustness, making them more susceptible to issues such as adversarial attacks, hallucinations, and factual inaccuracies. For instance, aggressive pruning, despite recent advances like the adaptive block-wise method Thanos [Ilin and Richtarik, 2025], can still disproportionately degrade reasoning quality and increase the likelihood of erroneous outputs. Similarly, while quantization methods such as GPTQ [Frantar *et al.*, 2022] and BiLLM [Huang *et al.*, 2024] substantially improve inference efficiency, they can distort internal representations, leading to brittle responses under adversarial or out-of-distribution conditions. Low-rank adaptation techniques, although effectively reducing model dimensionality, also risk undermining representational fidelity and generalization, further exacerbating robustness challenges.

Among these compression strategies, knowledge distillation (KD), particularly Chain-of-Thought (CoT) distillation, has emerged as a promising solution for even enhancing the reasoning capabilities and robustness of compressed models. Recent CoT distillation approaches, such as Symbolic Chain-of-Thought Distillation (SCoTD) [Li *et al.*, 2023] and DeepSeek distillation [Guo *et al.*, 2025], explicitly transfer structured reasoning processes from teacher models to

smaller student models, resulting in significant improvements in both interpretability and performance across complex reasoning tasks. Our work aligns with this promising direction by exploring an alternative yet complementary approach to CoT distillation.

In this paper, we propose **Dialectical Chain Distillation (DCD)**, a novel knowledge distillation framework designed to further enrich the reasoning capabilities of compressed LLMs through structured dialectical thinking, as shown in Figure 1. Rather than seeking to replace or surpass current CoT distillation methods, our goal is to explore a novel variant of reasoning representation inspired by teacher-student interactions and dialectical reasoning. Specifically, DCD leverages structured interactions between teacher model and student model to generate reasoning chains that capture diverse viewpoints, logical conflicts, and reflective argumentation. Consequently, the student model trained via DCD learns not only to provide correct predictions but also to internalize dialectical reasoning strategies, enhancing its robustness and interpretability in handling nuanced, adversarial, or ambiguous inputs.

We empirically evaluate DCD across multiple benchmarks, including AIME 24, GSM8K, and GPQA Diamond. The results show that student models trained with DCD exhibit improved robustness and reasoning accuracy compared to original Instruct model.

Our work makes the following key contributions:

- We propose **Dialectical Chain Distillation (DCD)**, a new variant of CoT distillation that emphasizes dialectical reasoning and structured argumentation.
- We introduce a teacher-student interaction mechanism to generate dialectical reasoning chains, serving as rich supervisory signals for student models.
- We empirically validate the effectiveness of DCD in enhancing reasoning accuracy across diverse tasks, contributing valuable insights toward robust real-world model deployment.

2 Related Work

2.1 Model Compression

The growing size of large language models (LLMs), often containing billions of parameters, has dramatically improved performance across natural language processing tasks. However, this rapid scaling has introduced serious limitations in terms of computational and memory requirements, especially when considering deployment in resource-constrained environments such as edge computing platforms, mobile devices, or real-time systems. To address these challenges, researchers have developed a variety of model compression techniques, including pruning, quantization, and knowledge distillation.

Pruning methods compress models by eliminating redundant or insignificant parameters. Recent advancements include structured pruning approaches such as *LLM-Pruner* [Ma *et al.*, 2023], which identifies and prunes structurally redundant components within transformer architectures, substantially reducing computational requirements while retaining task performance. In addition, Sun et al. [Sun *et al.*, 2023] propose *Wanda*, a one-shot pruning strategy that ranks weights based on a combination of magnitude and input activation norms. Without any fine-tuning, Wanda achieves performance on par with retraining-based baselines, offering a highly practical pruning solution.

Quantization compresses models by reducing numerical precision of weights and activations from floating-point to lower bit-width representations. *GPTQ* [Frantar *et al.*, 2022] introduces a highly effective post-training quantization (PTQ) approach leveraging second-order information for improved accuracy. More recently, *AWQ* [Lin *et al.*, 2024], an activation-aware method, achieved state-of-the-art results in low-bit (INT4) quantization, highlighting the critical role of activation distributions in successful quantization.

Knowledge distillation (KD) is a classical approach for transferring semantic knowledge from a large teacher model to a compact student model. Early work by Hinton *et al.* framed KD as matching softened output logits, enabling small networks to inherit the teacher's generalization behaviour [Hinton *et al.*, 2015]. Subsequent studies, such as Stanford *Alpaca* [Taori *et al.*, 2023] and *Self-Instruct* [Wang *et al.*, 2023b], extend this paradigm by using powerful teacher models to generate synthetic instruction–response pairs that serve as supervised data for student training.

2.2 Distilling Thinking Patterns

The first wave of knowledge-distillation research focused exclusively on aligning a student's output distribution with that of a teacher, for example by matching softened logits or regressing hidden states [Ding *et al.*, 2023]. Subsequent analyses crucially revealed a key weakness in this strategy: when the supervision signal is limited to final answers, students often learn to mimic surface-level style while failing to acquire the teacher's deeper underlying reasoning skills [Taori *et al.*, 2023; Mukherjee *et al.*, 2023]. This insight has shifted attention toward distilling intermediate thinking patterns rather than just outputs.

A landmark step in this direction is Chain-of-Thought (CoT) prompting [Wei *et al.*, 2023], which elicits step-by-step explanations from large models. Li *et al.* discover that transferring these explicit traces enables a model an order of magnitude smaller to match its teacher on math and science benchmarks [Li *et al.*, 2022]. Building on this, Symbolic CoT Distillation (SCoTD) converts free-form chains into executable symbolic programs before distillation, reducing noise and further improving transfer efficiency [Li *et al.*, 2023].

Beyond linear CoTs, researchers have explored richer rationale supervision. Orca [Mukherjee *et al.*, 2023] and Orca 2 [Mitra *et al.*, 2023] fine-tune students on a mixture of explanations, critiques, and summaries generated by GPT-4, exposing the model to diverse reasoning styles. Wang et al. [Wang *et al.*, 2023a] introduce a dedicated "reasoning token" to unify rationale distillation across tasks, whereas Cheng *et al.* use contrastive alignment to penalize shallow rationales and reward causally relevant ones [Cheng *et al.*, 2024]. Complementary work on self-improvement loops lets the student critique and refine its own drafts [Ye *et al.*, 2023] or selectively re-reason only faulty segments for efficiency [Madaan *et al.*, 2023; Li *et al.*, 2024b].

Another strand of research moves from single-speaker

traces to multi-agent debate. Li *et al.* distill knowledge from structured arguments between teacher agents with differing viewpoints [Li *et al.*, 2024a], while Zhang *et al.* show that critique-and-revise dialogues improve factual grounding over plain CoTs [Zhang *et al.*, 2024]. Collectively, these studies demonstrate that exposing students to structured thinking, whether through linear explanations, iterative reflection, or adversarial debate, consistently enhances robustness and reasoning depth.

Motivated by this evidence, we introduce a dialectical distillation pipeline that packages *a draft, deep reasoning, verification,* and *final answer* into a single supervised trace. By transferring such multi-perspective reasoning chains, our method seeks to endow compact models with the robustness and interpretability that conventional output-level distillation cannot provide, while preserving the efficiency benefits of knowledge distillation.

3 Dialectical Chain Distillation

Figure 2 gives a high-level overview of our approach. Starting from a problem prompt, we first elicit a student draft, this draft is kept in the loop whether it is correct or not, creating a natural *thesis*. The teacher then produces a detailed chain of thought and, crucially, appends a noise-free verification tag— ``correct'' or ``incorrect'', obtained by comparing the draft answer with the dataset's gold label. This single sentence injects the necessary *antithesis*: it either confirms the draft or highlights the need for revision. The dialogue proceeds until a gated <|im_start|>answer token prompts the definitive solution, closing the *synthesis*. Because every token of this dialectical exchange is preserved, supervised fine-tuning on the resulting corpus teaches the compact student not only to match the teacher's answers but also to emulate its conflict-driven reasoning cycle, yielding the robustness and interpretability gains.

3.1 Teacher–Student Interaction Protocol

The cornerstone of Dialectical Chain Distillation (DCD) is an interaction protocol that constructs a *multi-perspective reasoning dialogue* between a large teacher model and a compact student model. Each interaction unfolds in four stages:

Draft Generation. Upon receiving a problem prompt q, the *student* produces a concise draft consisting of two parts: a short, rapidly generated reasoning sketch and a tentative answer:

$$d = (\text{sketch}_s,\ a_s).$$

Because the student relies solely on its current parameters, this draft may be only partially reasoned or outright incorrect, precisely the uncertainty we wish to exploit for learning.

Teacher Deep Reasoning. Next, the *teacher*, a substantially larger model with advanced reasoning capabilities, generates an extensive chain of thought:

$$r = (r_1, r_2, \ldots, r_k),$$

where each r_i is a fine-grained reasoning step produced without knowledge of the student's answer. This chain constitutes a long-form reasoning trajectory that often extends across hundreds or thousands of tokens, and provides a high-resolution view into expert-level problem solving.

Verification. Correctness is assessed by grounding the student's draft against the dataset's gold answer a_{gt}. Specifically, we invoke a lightweight *verification assistant*, architecturally identical to the student, whose sole input is the pair $(a_s,\ \ a_{\text{gt}})$. If the two values coincide, the verifier emits the fixed tag ``The draft solution is correct.'', otherwise it emits ``The draft solution is incorrect.'' Because the decision is anchored to a_{gt}, the tag provides a noise-free, binary signal that the subsequent prompt engineering encodes as part of the dialectical chain.

Finalisation. Finally, the dialogue terminates the dialectical thinking phase and requests a deterministic answer. Leveraging the verified reasoning, the teacher produces a definitive solution:

$$a^\star = \text{FinalAnswer}(r).$$

This protocol yields three complementary supervision signals:

1. *Answer Alignment* — the final answer $a^\star$, anchored to the dataset gold label, teaches the student to converge on ground-truth outcomes.
2. *Reasoning Alignment* — the teacher's long-form chain r exposes a token-level blueprint of expert problem solving, allowing the student to imitate fine-grained logical moves.
3. *Verification Feedback* — the binary tag (*correct* / *incorrect*) injects an explicit conflict–resolution cue, guiding the student to recognise when its initial intuition must be revised.

Why dialectical? Unlike traditional linear Chain-of-Thought prompting, our protocol explicitly incorporates stages of conflict and resolution. The interplay between the student's initial draft and the teacher's critique compels the student to actively engage with alternative viewpoints, internalizing the dialectical cycle of thesis, antithesis, and synthesis. This structured confrontation and reconciliation enhances the student's capacity for reflective thinking and robust self-correction.

3.2 Data Sourcing and Sample Construction

To instantiate the interaction protocol at scale, we construct our supervision corpus directly from the **s1k-1.1** benchmark [Muennighoff *et al.*, 2025].[1]. Specifically, we combine the original annotated data with a raw JSONL dump containing alternative student drafts. Below, we describe how these two components are integrated to produce the "draft–correct" and "draft–wrong" pairs used for distillation.

We begin by recalling that every record in **s1k-1.1** already contains four gold-quality elements:

(i) a problem statement q,

(ii) a *Gemini* chain-of-thought trajectory and its answer a_{G},

(iii) a long *DeepSeek-R1* chain-of-thought trajectory and its answer a_{D},

[1] https://huggingface.co/datasets/simplescaling/s1K-1.1

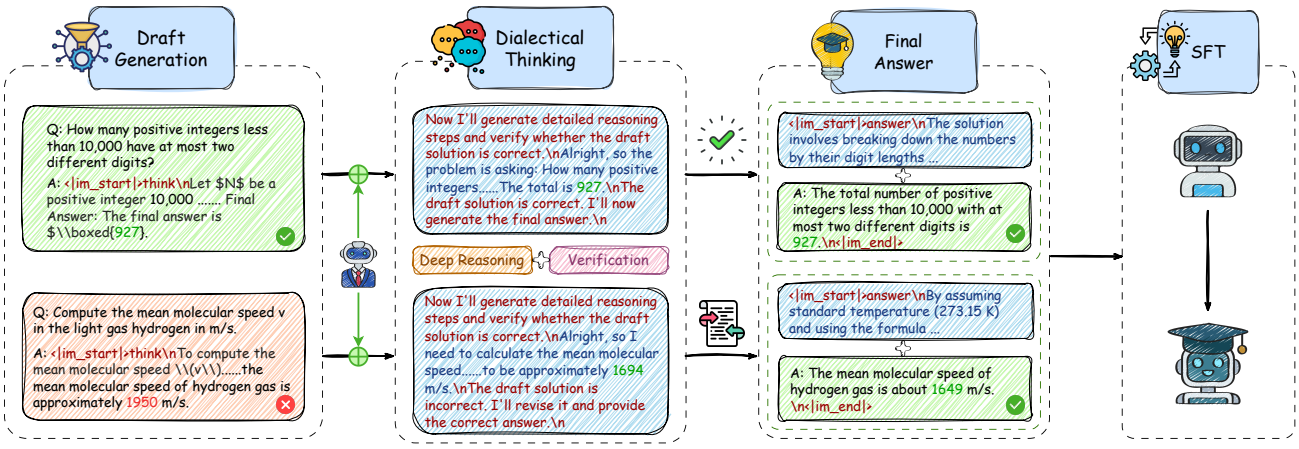


Figure 2: End-to-end workflow of **Dialectical Chain Distillation**. *Draft Generation* (left) creates two complementary cases: a draft that is already correct (green) and a draft that is wrong (red). Both drafts are passed to the *Dialectical Thinking* block where the teacher supplies a long, explicit reasoning chain. A mandatory self-verification sentence triggers either the `correct` or `incorrect` marker, forcing the model to confirm or revise the draft. The dialogue then transitions to the *Final Answer* stage, where a single `<|im_start|>answer` tag gates the definitive prediction. All tokens, including the binary verification cue, are retained and used to supervise a compact student model through standard SFT (right).

(iv) the dataset ground-truth solution a_{gt} (with $a_{\text{G}} = a_{\text{D}} = a_{\text{gt}}$ by construction).

These ingredients allow us to fabricate both "draft–correct" and "draft–wrong" scenarios with no additional human annotation.

Generating student drafts. For every question q, we prompt an un-tuned copy of our target student model to produce a first-pass reasoning sketch and provisional answer a_s. Empirically, fewer than 35% of these drafts coincide with a_{gt}, giving us a natural supply of realistic errors.

Forming draft–correct pairs. To create positive supervision, we treat the Gemini solution a_{G} as a *correct* student draft. We pair it with the expert-level DeepSeek chain r and tag the draft as *correct*. These samples teach the student how a sound "first attempt" should look.

Forming draft–wrong pairs. Whenever $a_s \neq a_{\text{gt}}$, we retain the student's erroneous draft (sketch_s, a_s), reuse the same long chain r, and tag the draft as *incorrect*. These examples expose the student to error detection and subsequent revision.

Balancing the corpus. To maintain a realistic supervision signal, we aggregate all generated draft–correct and draft–wrong pairs. The resulting set exhibits a reasonably balanced distribution, comprising approximately 1,000 correct and 675 incorrect samples.

Resulting supervision set. Each assembled example is written as a single JSON line containing the question, the chosen draft (correct or wrong), the full DeepSeek reasoning chain, and the DeepSeek answer. The resulting file, comprises 1,675 training prompts with an average length of 10,905 tokens, long enough to preserve the complete dialectical trajectory that will be distilled in subsequent stages.

3.3 Prompt Engineering for Forced Reasoning

To guarantee that the distilled student model rehearses an explicit reasoning phase before committing to a final answer, we structure every training prompt with a sequence of *sentinel tags*. These tags delineate (i) whether the model is *thinking* or *answering*, and (ii) whether the draft has been judged correct or incorrect.

Forcing a reasoning turn. Immediately after the assistant role tag we insert the token pair:

`<|im_start|>assistant\n`**`<|im_start|>think`**

which acts as an unambiguous cue that the model must enter a *thinking* mode. Only once this phase ends may the model emit an answer.

Guiding the depth of thought. Immediately after the `Draft` delimiter we append a fixed instruction:

```
``Now I will generate detailed
reasoning steps and verify whether
the draft solution is correct.''
```

Placed right at the stage boundary, this sentence (i) forces the model to unfold a step-by-step line of reasoning and (ii) prepares it for the explicit verification cue that follows.

Encoding the verification outcome. When the teacher's reasoning chain concludes, we insert *one* of the following mutually exclusive markers, thereby signaling the transition from reasoning to synthesis:

- **Draft-Correct Tag:**
  ```
  ``The draft solution is correct.  I'll
  now generate the final answer.''
  ```
- **Draft-Wrong Tag:**
  ```
  ``The draft solution is **incorrect**.
  I'll revise it and provide the correct
  answer.''
  ```

Because this tag is selected only by comparing the draft answer with the dataset gold solution, it provides a noise-free, binary signal that cleanly separates confirmation from correction.

Answer emission. The model may exit the `think` phase only when it encounters the stage delimiter:

<|im_start|>answer

It then outputs the definitive answer and terminates the dialogue with a single <|im_end|> token.

Overall template. For brevity, we illustrate the layout on a draft–*correct* example. The only difference in a draft–*wrong* instance is the verification tag.

```
<|im_start|>system
You are Qwen, created by Alibaba Cloud.
...
<|im_end|>

<|im_start|>user
{question q}
<|im_end|>

<|im_start|>assistant
<|im_start|>think
{Draft}
Now I'll generate detailed reasoning steps
and verify whether the draft solution is
correct.
{DeepSeek long chain r_1,...,r_k}
The draft solution is correct.
I'll now generate the final answer.
<|im_start|>answer
{Final Answer}
<|im_end|>
```

3.4 Training Objective

Supervised fine-tuning. The distilled student is obtained by standard supervised fine-tuning (SFT). For each prompt x we minimise the token-level cross-entropy:

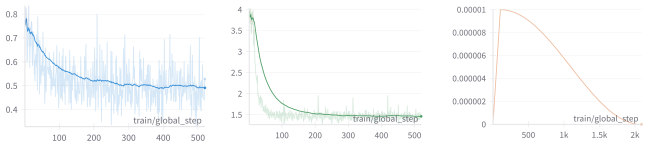
$$\mathcal{L}_{\text{SFT}} = -\sum_{t \in \mathcal{A}(x)} \log p_\theta\big(w_t \mid w_{<t}, x\big),$$

where $\mathcal{A}(x)$ indexes only those tokens that lie inside an `assistant` segment (starting at the first <|im_start|>think or <|im_start|>answer tag and ending at <|im_end|>). All system and user tokens are masked out, so the model is trained exclusively to reproduce (i) the reasoning chain, (ii) the verification tag, and (iii) the final answer.

4 Experiments

4.1 Datasets and Evaluation Settings

We systematically evaluate our proposed framework using three benchmarks: AIME 24 [MAA, 2024], GSM8K [Cobbe *et al.*, 2021], and GPQA Diamond [Rein *et al.*, 2023]. These benchmarks span tasks in mathematical reasoning and graduate-level scientific knowledge.



(a) Training loss (b) Gradient norm (c) LR schedule

Figure 3: Training dynamics for the proposed method.

AIME 24 dataset comprises 30 problems selected from the 2024 American Invitational Mathematics Examination (AIME) I and II contests. AIME is a prestigious high school mathematics competition recognized for its challenging problems that assess advanced mathematical reasoning. The dataset covers a wide range of topics, including algebra, geometry, and number theory, and typically requires multi-step reasoning to solve.

GSM8K (Grade School Math 8K) consists of 8,500 grade school-level mathematical word problems. For evaluation purposes, we use a 1,319-question test set with diverse, high-quality problems. Each problem typically takes 2–8 steps to solve and is designed to test a model's ability in step-by-step arithmetic reasoning. GSM8K is a standard benchmark for math reasoning in LLMs.

GPQA Diamond is a tough subset of the GPQA (Graduate-Level Google-Proof Q&A) benchmark, with 198 multiple-choice questions in advanced biology, chemistry, and physics. Created by experts, the questions resist simple search strategies and pattern-matching. Each question includes one correct answer and several strong distractors, making it a rigorous test of scientific reasoning.

Evaluation Protocol. All experiments are carried out using the LM-EVALUATION-HARNESS toolkit with inference facilitated by vLLM. For **AIME 24**, we report *pass@1* accuracy, defined as the proportion of problems correctly solved on the first attempt. **GSM8K** is evaluated under a 5-shot setting, utilizing the original few-shot exemplars provided by the evaluation harness. **GPQA Diamond** is assessed using a strictly zero-shot, multiple-choice configuration, with results reported as top-1 accuracy.

4.2 Training Details

Figure 3 visualizes the optimization trace of QWEN2.5-1.5B-INSTRUCT [Qwen *et al.*, 2025] student during supervised fine-tuning.

Training Configuration We fine-tune our model for 5 epochs with a batch size of 1, accumulating gradients over 16 steps for a total of 520 gradient updates. Training is conducted in bfloat16 precision with an initial learning rate of 1×10^{-5}. This rate is linearly warmed up over the first 5% of steps (26 steps) and then decays to zero following a cosine schedule over the remaining 494 steps. We employ the AdamW optimizer [Loshchilov and Hutter, 2017] with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and a weight decay of 1×10^{-4}. FlashAttention 2 is enabled for efficient attention computation, and we use liger kernel to activate mixed-precision kernels. On a single NVIDIA RTX 4090, training completes in approximately 46 minutes.

Model	AIME 24	GSM8K	GPQA-Diamond
Qwen2.5-1.5B-Instruct	0/30	54.66%	23.73%
+ Chain-of-Thought prompting†	0/30	63.46%	–
Qwen2.5-1.5B-Instruct + SFT on `s1K-1.1`	0/30	65.95%	24.24%
Qwen2.5-1.5B-Instruct + **DCD (ours)**	**2/30**	**67.02%**	**25.25%**

Table 1: Accuracy on three reasoning benchmarks (pass@1 for AIME 24, top-1 accuracy for GSM8K and GPQA-Diamond). †Chain-of-Thought prompting is applied only at inference time without additional training.

4.3 Experimental Analysis

We assess the efficacy of our Dialectical Chain Distillation (DCD) framework through comparative evaluations of the student model before and after applying DCD fine-tuning. To contextualize our findings, we also include two baseline configurations: standard Chain-of-Thought (CoT) prompting applied at inference time (without additional training) and conventional supervised fine-tuning using the 's1K-1.1' dataset, which consists of high-quality reasoning chains but lacks explicit dialectical structures. The comprehensive evaluation results across three benchmarks are presented in Table 1.

The baseline model, QWEN2.5-1.5B-INSTRUCT, struggles particularly on the challenging AIME 24 benchmark, yielding minimal performance even when leveraging inference-time CoT prompting. Although CoT prompting notably improves the baseline performance on GSM8K, it remains insufficient for the more demanding reasoning tasks represented by AIME 24.

Applying standard supervised fine-tuning with the 's1K-1.1' dataset provides further incremental performance gains on GSM8K and GPQA-Diamond benchmarks. However, this method does not enhance the model's capability to solve the significantly more difficult AIME 24 problems.

In contrast, our proposed DCD approach consistently elevates the reasoning performance across all benchmarks. Crucially, it demonstrates unique effectiveness by enabling progress on the AIME 24 benchmark, underscoring the benefit of explicitly incorporating dialectical reasoning structures into training. The integration of drafting, deep reasoning, verification, and finalization stages within the DCD framework significantly improves the robustness and complexity of reasoning that smaller-scale models can handle.

5 Conclusion

In this paper, we present a novel knowledge distillation framework designed to improve the reasoning capability and robustness of compressed LLMs. By introducing structured teacher–student interactions and generating dialectical reasoning chains, DCD provides interpretable and informative supervision signals that extend the capabilities of conventional Chain-of-Thought distillation by incorporating dialectical structure. Experimental results across diverse reasoning benchmarks including AIME 24, GSM8K, and GPQA Diamond demonstrate that DCD substantially improves both accuracy and robustness over standard model compression methods. These results demonstrate the effectiveness of DCD in producing compact and robust LLMs suitable for deployment in resource-constrained environments. In the future, we plan to scaling up training on more diverse and larger-scale reasoning datasets, and extending its application to broader classes of reasoning and decision-making tasks.

References

[Cheng *et al.*, 2024] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models to domains via reading comprehension, 2024.

[Cobbe *et al.*, 2021] 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.

[Ding *et al.*, 2023] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations, 2023.

[Frantar *et al.*, 2022] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[Huang *et al.*, 2024] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*, 2024.

[Ilin and Richtarik, 2025] Ivan Ilin and Peter Richtarik. Thanos: A block-wise pruning algorithm for efficient large language model compression. *arXiv preprint arXiv:2504.05346*, 2025.

[Li *et al.*, 2022] Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. Explanations from large language models make small reasoners better, 2022.

[Li *et al.*, 2023] Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also" think" step-by-step. *arXiv preprint arXiv:2306.14050*, 2023.

[Li *et al.*, 2024a] Ming Li, Jiuhai Chen, Lichang Chen, and Tianyi Zhou. Can llms speak for diverse people? tuning llms via debate to generate controllable controversial statements, 2024.

[Li *et al.*, 2024b] Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16189–16211, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Lin *et al.*, 2024] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Ma *et al.*, 2023] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[MAA, 2024] MAA. American invitational mathematics examination - aime, February 2024. Accessed February 2024.

[Madaan *et al.*, 2023] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

[Mitra *et al.*, 2023] 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, 2023.

[Muennighoff *et al.*, 2025] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[Mukherjee *et al.*, 2023] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

[Qwen *et al.*, 2025] Qwen, :, 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.

[Rein *et al.*, 2023] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

[Sun *et al.*, 2023] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[Taori *et al.*, 2023] 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.

[Wang *et al.*, 2023a] Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. Making large language models better reasoners with alignment, 2023.

[Wang *et al.*, 2023b] 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, 2023.

[Wei *et al.*, 2023] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[Ye *et al.*, 2023] Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post, May 2023.

[Zhang *et al.*, 2024] Yichi Zhang, Zhuo Chen, Yin Fang, Yanxi Lu, Li Fangming, Wen Zhang, and Huajun Chen. Knowledgeable preference alignment for LLMs in domain-specific question answering. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 891–904, Bangkok, Thailand, August 2024. Association for Computational Linguistics.