Enhancing Long-Chain Reasoning Distillation through Error-Aware Self-Reflection

Zhuoyang Wu[1*], **Xinze Li**[1*], **Zhenghao Liu**[1†], **Yukun Yan**[3],
Zhiyuan Liu[3], **Minghe Yu**[2], **Cheng Yang**[4], **Yu Gu**[1], **Ge Yu**[1], **Maosong Sun**[3]
[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]Software College, Northeastern University, Shenyang, China
[3]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[4]School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

Abstract

Large Language Models (LLMs) have demonstrated strong reasoning capabilities and achieved remarkable performance on mathematical reasoning tasks. Recently, distilling reasoning ability from long-form Chain-of-Thought (CoT) has emerged as a promising approach for enhancing small-scale models. Most existing studies treat small-scale models as students and leverage long-form CoTs generated by superior teacher models as training targets for Supervised Fine-Tuning (SFT) to distill reasoning ability. However, such long-form reasoning trajectories are often misaligned with the student's capacity, making it difficult for the student model to effectively internalize the provided reasoning steps. To address this limitation, we propose err**O**r-aware self-**R**eflect**ION** (ORION), a framework that adapts teacher CoTs through an error-aware self-reflection process. Specifically, ORION prompts the student model to refine teacher-generated trajectories by explicitly incorporating its own reasoning errors, thereby producing student-tailored reasoning trajectories for SFT. Experiments on multiple mathematical reasoning benchmarks demonstrate that ORION consistently improves student performance by more than 2% over all baselines. Further analysis shows that the CoTs constructed by ORION exhibit higher coherence and logical consistency, making them more effective supervision signals for SFT. All codes are available at https://github.com/NEUIR/ORION

1 Introduction

Large Language Models (LLMs) have demonstrated strong reasoning capabilities in mathematical problem-solving (Brown et al., 2020; Zhang et al., 2022). By leveraging the Chain-of-Thought (CoT) paradigm (Wei et al., 2022), LLMs can decompose mathematical problems into intermediate reasoning steps and solve them in a step-by-step manner (Fu et al., 2023; Li et al., 2023), thereby achieving substantially improved performance. However, for small-scale models, generating reliable and logically coherent CoTs remains challenging, especially when tackling complex mathematical tasks (Wang et al., 2025).

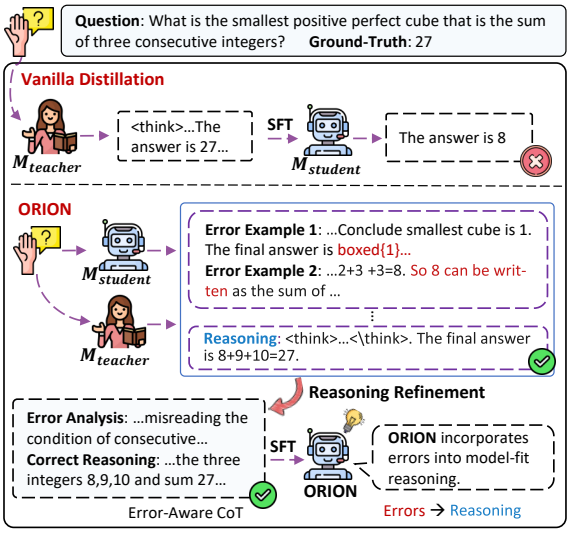


Figure 1: The Framework of Our ORION Model. ORION distills the reasoning capability of the teacher model ($\mathcal{M}_{\text{teacher}}$) by refining teacher-generated CoTs for training the student model ($\mathcal{M}_{\text{student}}$).

Recent studies have focused on distilling reasoning abilities from superior LLMs to enhance the mathematical reasoning performance of small-scale models (Zhou et al., 2021; Hsieh et al., 2023). These approaches typically regard small-scale models as students and stronger LLMs as teachers, transferring reasoning capabilities by training the student to imitate the teacher's underlying reasoning patterns (Li et al., 2024a). While such direct distillation strategies have shown promising results, subsequent work has further improved student reasoning by enabling learning from the student's own mistakes (An et al., 2023) or by lever-

[*] indicates equal contribution.
[†] indicates corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 1465–1482

aging more informative long-form reasoning trajectories (DeepSeek-AI et al., 2025). In particular, recent studies have explored error-aware learning paradigms, where a teacher model corrects and analyzes errors in the student's solutions (Jain et al., 2025; Yang et al., 2024b). The corrected solutions are then used as supervision targets to guide the student in avoiding similar mistakes during inference (Yang et al., 2024b; Li et al., 2024b). However, teacher-generated corrections are not always reliable and may introduce additional noise into the supervision signal. More recently, researchers have explored distillation from long-form reasoning trajectories that involve deeper exploration and explicit reflection, enabling the student model to internalize corrective reasoning patterns and thereby achieve further performance improvements (Ye et al., 2025; DeepSeek-AI et al., 2025). Nevertheless, the substantial capability gap between teacher and student models often hinders effective learning from such long and complex reasoning trajectories, ultimately limiting the effectiveness of reasoning distillation (Li et al., 2025b).

In this paper, we introduce err**O**r-aware self-reflect**ION** (ORION), a novel approach that delivers more effective supervision signals by leveraging student self-reflection to distill reasoning capabilities from a teacher model. As illustrated in Figure 1, ORION first exposes potential erroneous solutions generated by the student model by sampling with different decoding temperatures. Then, conditioned on its generated errors and the teacher-provided long-form CoTs, the student model reflects on these errors and refines the CoT results into student-tailored reasoning trajectories. Finally, we construct Supervised Fine-Tuning (SFT) data by pairing each query together with its sampled erroneous solutions as inputs, and using the refined CoTs as training targets.

Our experimental results demonstrate that ORION consistently outperforms all baselines across mathematical reasoning tasks of varying difficulty, highlighting its overall effectiveness. Further analysis shows that, compared to directly using long-form CoTs as training targets, ORION achieves lower and more stable entropy during training, indicating improved training stability. Moreover, ORION effectively reduces redundant reasoning patterns present in long-form CoTs, enabling student models to acquire more concise and higher-quality reasoning processes. Finally, we observe that ORION effectively corrects multiple types of errors produced by student models, with a particular strength in addressing reasoning errors. This further validates the effectiveness of the error-aware CoT refinement mechanism in providing higher-quality supervision for training.

2 Related Work

Large Language Models (LLMs) have proven effective in solving mathematical problems (Cobbe et al., 2021; Hendrycks et al., 2021). Advanced prompting strategies, such as Chain-of-Thought (CoT) (Wei et al., 2022), decompose the problem-solving process of LLMs into a series of intermediate steps, significantly enhancing their mathematical reasoning abilities (Li et al., 2023; Qin et al., 2023; Luo et al., 2023a). To further improve the performance of small-scale models, some studies fine-tune models with annotated solutions or rationales (Hsieh et al., 2023). However, this approach may lead to model overfitting to supervision signals, limiting the model's ability to learn genuine reasoning knowledge from such supervision (Luo et al., 2023b; Gudibande et al., 2023).

To construct high-quality SFT datasets, recent studies typically explore long-form reasoning distillation methods (Yang et al., 2025a; Ye et al., 2025). Specifically, these methods treat Reasoning Language Models (RLMs), such as DeepSeek-R1 (DeepSeek-AI et al., 2025), as teacher models and use their generated long-form CoTs as supervision targets to fine-tune student models, enabling them to learn not only the diverse mathematical solution knowledge but also the underlying reasoning patterns (Li et al., 2025a; Yang et al., 2025b). However, several studies have shown that student models are constrained by their limited capacity and reasoning capability, making it difficult for them to effectively acquire reasoning knowledge from the long-form CoTs produced by more powerful teacher models (Li et al., 2025b). Moreover, such distillation may introduce additional issues, including content repetition and excessively verbose rationales (Li et al., 2025b).

Different from approaches that directly distill long-form reasoning trajectories from teacher models, another line of work constructs an SFT dataset by leveraging teacher models to correct or analyze erroneous reasoning trajectories generated by the student model (Tong et al., 2024; Pan et al., 2025). Specifically, An et al. (2023) propose an error-correction-based SFT framework, in which

incorrect reasoning produced by the student is used as input, and the student is trained to reproduce the corresponding corrections generated by the teacher. Tong et al. (2024) further prompt the LLM with potential errors collected from the teacher model's corrections, enabling the student to not only learn the correct answers but also avoid making similar mistakes during SFT. However, these methods predominantly rely on teacher-generated corrections and largely overlook how effectively the student model can internalize and benefit from such corrective supervision. To address this limitation, ORION explicitly collects student-generated errors and prompts the student to refine the teacher's long-form reasoning trajectories conditioned on these errors, resulting in student-tailored supervision that more effectively facilitates reasoning distillation.

3 Methodology

In this section, we present Error-Aware Self-Reflection (ORION), a method designed to refine teacher-produced long-form reasoning trajectories by leveraging the student model itself for self-reflection and refinement. As shown in Figure 2, we first describe how to distill knowledge from long-form reasoning trajectories using standard SFT (Sec. 3.1). We then introduce our error-aware data refinement framework (Sec. 3.2), which employs the student model itself to identify potential reasoning errors and refine the teacher-generated CoTs into higher-quality supervision signals.

3.1 Distilling Long Reasoning Capability into Student Models via SFT

Given a mathematical question q, the reasoning task requires the model to perform a series of reasoning steps to solve the problem. To enhance the mathematical reasoning ability of small-scale models, we treat one such model as the student model $\mathcal{M}_{\text{student}}$ and investigate two SFT strategies for optimizing its parameters: vanilla SFT and distillation using teacher-generated long-form reasoning trajectories.

Vanilla SFT Method. In the vanilla SFT paradigm, the student model is fine-tuned on human-annotated solution labels $\mathcal{D}_{\text{Raw}} = \{(q^1, L^1), \ldots, (q^n, L^n)\}$, which serve as ground-truth outputs for each input question q^i. The training objective minimizes the negative log-likelihood of the ground truth solution L^i:

$$\mathcal{J} = -\sum_{i=1}^{n}\sum_{t=1}^{|L^i|} \log P(L_t^i | L_{<t}^i, \text{Instruct}_{\text{QA}}(q^i); \mathcal{M}_{\text{student}}). \tag{1}$$

While this approach has demonstrated strong empirical performance in enhancing the problem-solving accuracy of student models (Hsieh et al., 2023), it often overfits to the training labels, thereby limiting the student model's ability to acquire genuine reasoning knowledge from supervision (Luo et al., 2023b; Gudibande et al., 2023). Consequently, recent research has increasingly focused on building high-quality datasets for more effective reasoning distillation (Wettig et al., 2024).

Long-form Reasoning Distillation. Unlike the standard SFT approach, the reasoning distillation method incorporates Chain-of-Thought (CoT) supervision to guide the student model through intermediate reasoning steps, rather than directly optimizing for the human-annotated solution labels (Hsieh et al., 2023). This paradigm encourages the student model $\mathcal{M}_{\text{student}}$ to learn not only the correct solutions but also the underlying logical structures and reasoning patterns exhibited by the teacher model $\mathcal{M}_{\text{teacher}}$.

Specifically, given a question q, we first regard a Reasoning Language Model (RLM), such as DeepSeek-R1 (DeepSeek-AI et al., 2025), as the teacher model $\mathcal{M}_{\text{teacher}}$ and prompt it to generate a long-form reasoning response o:

$$o = \mathcal{M}_{\text{teacher}}(\text{Instruct}_{\text{QA}}(q)). \tag{2}$$

We then construct the SFT dataset $\mathcal{D} = (q^1, o^1), \ldots, (q^n, o^n)$ by pairing each input query q^i with its corresponding $\mathcal{M}_{\text{teacher}}$-generated response o^i. The distilled reasoning response o^i is used as the supervision target for fine-tuning the student model $\mathcal{M}_{\text{student}}$, enabling it to acquire step-by-step reasoning abilities from the teacher's intermediate reasoning trajectories:

$$\mathcal{J} = -\sum_{i=1}^{n}\sum_{t=1}^{|o^i|} \log P(o_t^i | o_{<t}^i, \text{Instruct}_{\text{QA}}(q^i); \mathcal{M}_{\text{student}}), \tag{3}$$

where $|o^i|$ denote the token number of o^i. This distillation approach encourages the student model to mimic the reasoning behaviors of more capable teacher models, thereby enabling the student model to generate coherent and logically grounded CoT outputs. However, due to the reasoning capability gap between teacher and student models, directly

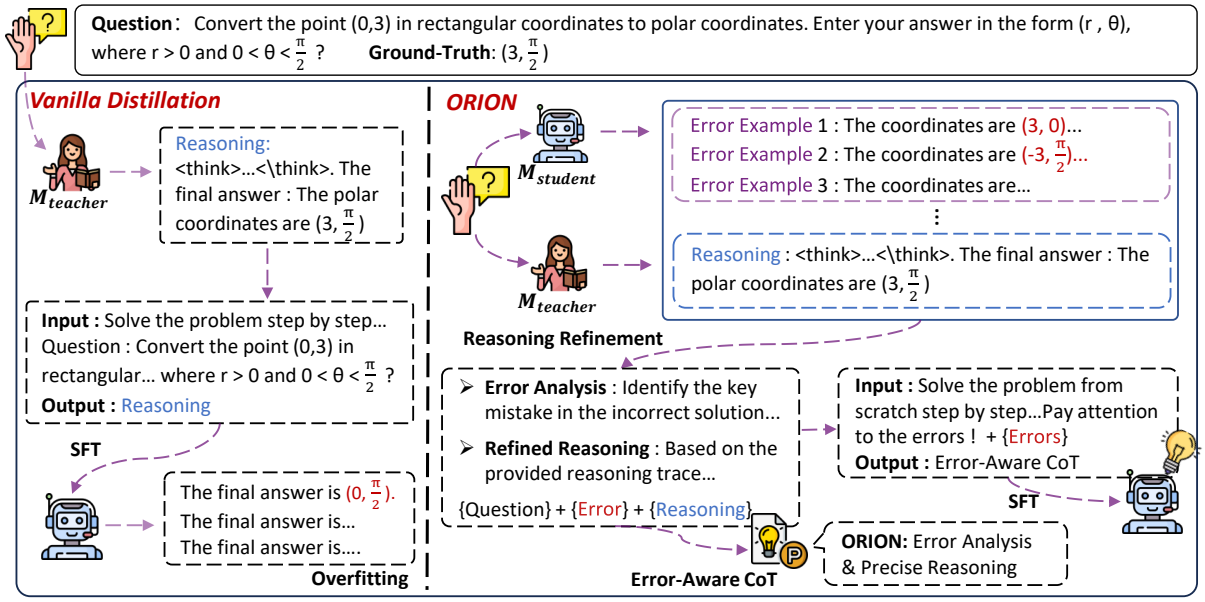


Figure 2: Illustration of Our ORION Model.

fine-tuning student models on long-form reasoning trajectories can be challenging. Direct distillation from the teacher may result in degenerate behaviors, such as repetitive outputs or unstable training dynamics (Yin et al., 2025; Li et al., 2025b). To mitigate these issues, ORION refines long-form CoTs through error-aware self-reflection, providing more targeted and effective training signals for SFT (Sec. 3.2).

3.2 Refining Long-Form Reasoning via Error-Aware Self-Reflection

To enhance the quality of long-form reasoning data for SFT, ORION introduces an error-aware reflection mechanism that leverages the student model's own mistakes and self-reflection to refine the teacher-provided long-form CoTs.

For the given dataset $\mathcal{D} = (q^1, o^1), \ldots, (q^n, o^n)$, where o^i is the teacher's output for question q^i, ORION first prompts the student model $\mathcal{M}_{\text{student}}$ to generate multiple candidate solutions Y^i for each q^i, facilitating self-exposure to errors. Based on these self-generated solution errors, the student model $\mathcal{M}_{\text{student}}$ is then prompted to refine the teacher-generated long-form reasoning through self-reflection, yielding a refined dataset $\tilde{\mathcal{D}}$. This dataset is subsequently used to enhance the reasoning capability of the student model via SFT (Eq. 3).

Error Exposure via Response Sampling. To uncover typical reasoning errors made by the student model $\mathcal{M}_{\text{student}}$, we adopt a response sampling strategy inspired by (An et al., 2023).

For each input question q^i, we generate a diverse set of K candidate solutions $Y^i = \{y_1^i, \ldots, y_K^i\}$ by sampling from $\mathcal{M}_{\text{student}}$ across a range of temperatures τ:

$$Y^i \sim \text{Sample}_\tau(\mathcal{M}_{\text{student}}(\text{Instruct}_{\text{QA}}(q^i))). \quad (4)$$

Each sampled response $y_k^i \in Y^i$ is then post-processed using an answer extraction function $\text{Ans}(\cdot)$, and its correctness is evaluated according to the ground-truth label L^i. These responses that yield incorrect final answers are collected into an error set Y_{err}^i:

$$Y_{\text{err}}^i = \{y_k^i \mid \text{Ans}(y_k^i) \neq L^i\}. \quad (5)$$

The erroneous solutions $Y_{\text{err}}^i = \{y_1^i, \ldots, y_m^i\}$ serve as references for refining the teacher-generated CoT, allowing the student model to recognize failure patterns and adjust the reasoning trajectories.

Reasoning Refinement through Error-Aware Self-Reflection. To help the student model internalize more robust reasoning trajectories during SFT, we introduce a self-reflection mechanism that leverages its own erroneous reasoning outcomes to enhance the quality of the training data. Given an initial long-form reasoning-based SFT dataset $\mathcal{D} = \{(q^1, o^1), \ldots, (q^n, o^n)\}$, the student model $\mathcal{M}_{\text{student}}$ is encouraged to analyze its mistakes and refine the original teacher's CoT outputs for SFT.

For each mathematical query q^i in $\mathcal{D}$, we begin with an initial long-form CoT-style response o^i and a set of student-generated erroneous attempts

$Y^i_{\text{err}} = \{y^i_1, \ldots, y^i_m\}$. For each error example $y^i_k \in Y^i_{\text{err}}$, the student model $\mathcal{M}_{\text{student}}$ is prompted to refine the $\mathcal{M}_{\text{teacher}}$-generated reasoning outcome o^i by reflecting on its self-made error y^i_k, resulting in the refined reasoning outcome $\tilde{o}^i_k$:

$$\tilde{o}^i_k = \mathcal{M}_{\text{student}}(\text{Instruct}_{\text{Ref}}(q, y^i_k, o^i)), \tag{6}$$

where $\text{Instruct}_{\text{Ref}}$ denotes a self-reflection instruction that guides the student model $\mathcal{M}_{\text{student}}$ to analyze and refine o^i based on its error response y^i_k. This yields a set of refined reasoning responses for the mathematical query q^i:

$$\tilde{O}^i = \{\tilde{o}^i_1, \ldots, \tilde{o}^i_m\}. \tag{7}$$

To ensure the correctness of the refinements, we filter out any $\tilde{o}^i_k$ whose final answer does not match the ground-truth label L^i:

$$\tilde{\mathcal{D}}^i = \{(q^i, \tilde{o}^i_k) \mid 1 \leq k \leq m,\ \text{Ans}(\tilde{o}^i_k) = L^i\}. \tag{8}$$

Finally, we construct the refined SFT dataset $\tilde{\mathcal{D}}$ by aggregating all validated refinements $\tilde{\mathcal{D}}$ across all mathematical queries $\{q^1, \ldots, q^n\}$:

$$\tilde{\mathcal{D}} = \bigcup_{i=1}^{n} \tilde{\mathcal{D}}^i. \tag{9}$$

4 Experimental Methodology

This section first describes the datasets, baselines, and evaluation metric, followed by the implementation details of our experiments.

Datasets. We randomly sample 10,000 examples from OpenR1-Math-220k (Hugging Face, 2025) to construct the training set, which provides high-quality solution trajectories for mathematical reasoning. For evaluation, we use four math benchmarks spanning different difficulty levels. GSM-Hard (Gao et al., 2023) increases the computational difficulty of GSM8K (Cobbe et al., 2021). MATH500 (Godahewa et al., 2021) is a subset of 500 competition-style problems randomly sampled from the MATH dataset, covering a wide range of difficulty. AIME (AIM, 2025) and AMC (AMC, 2025) are widely used in U.S. high-school mathematics competitions featuring challenging problems across diverse domains, such as algebra, number theory, and probability.

Baselines. We compare ORION with both zero-shot models and SFT models. We first compare with two zero-shot baselines: vanilla LLM and Wrong-of-Thought (Zhang et al., 2024). For the vanilla LLM, we provide the mathematical question q to the model and ask it to directly generate a solution. Additionally, we compare ORION with Wrong-of-Thought (Zhang et al., 2024) in our experiments. The model first generates an initial answer, then identifies solution errors using a multi-perspective verifier, and regards these errors as signals to help LLMs prevent the repetition of the same mistakes. Furthermore, we utilize two SFT-based baselines, including SFT (Label) and SFT (Long-CoT) (DeepSeek-AI et al., 2025). Following previous work (Wang et al., 2024a), SFT (Label) and SFT (Long-CoT) fine-tune the LLMs using human-annotated labels and reasoning trajectories from the teacher models.

Evaluation Metrics. Following prior studies (Zhang et al., 2024), we adopt the Accuracy score as the evaluation metric to determine whether the mathematical QA models correctly generate the ground-truth answers. Specifically, we report Acc@1 and Acc@10, which represent the average accuracy over one and ten sampled answers, respectively (Liu et al., 2024).

Implementation Details. For all experiments, we use Qwen2.5-7b-Instruct (Yang et al., 2024a), Qwen3-8b (Yang et al., 2025a) and Llama3.1-8b-Instruct (Grattafiori et al., 2024) as backbone models. Additionally, we utilize DeepSeek-R1 (DeepSeek-AI et al., 2025) as the teacher model. We train each model for 3 epochs with a learning rate of 5×10^{-5} and gradient accumulation of 8, using LoRA (Hu et al., 2022) for efficient training. Our ORION model is built on TRL[1] and LLaMA Efficient Tuning[2]. Further experimental details and data statistics are provided in Appendix A.2, and prompt templates are in Appendix A.6.

5 Evaluation Results

In this section, we first evaluate the overall performance of ORION through the main experiences. Next, we conduct ablation studies to investigate the specific contribution of solution error and self-reflection components. Furthermore, we analyze the effectiveness of distilled models via ORION. Finally, some case studies are shown in Appendix A.9.

[1] https://github.com/huggingface/trl
[2] https://github.com/hiyouga/LLaMA-Factory

Model	AIME24		AMC23		Math500		GSM-H		Average	
	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10
Qwen3-8B										
Vanilla LLM	20.00	21.10	55.00	53.10	81.40	80.75	57.40	55.82	53.45	52.69
Wrong-of-Thought	16.67	16.90	55.50	54.85	79.83	80.14	58.16	57.95	52.54	52.46
SFT (Label)	16.67	18.83	52.50	51.55	80.95	79.82	57.13	56.55	51.81	51.69
SFT (Long-CoT)	23.33	22.50	57.50	56.90	82.90	81.40	59.27	58.82	55.75	54.91
ORION	**26.67**†‡§	**25.60**†‡§	**62.50**†‡§	**61.85**†‡§	**83.50**†‡	**82.95**†‡	**59.83**†‡	**59.75**†‡	**58.13**†‡§	**57.54**†‡§
Qwen2.5-7B-Instruct										
Vanilla LLM	10.00	6.20	47.50	44.00	69.80	69.80	55.72	54.34	45.76	43.59
Wrong-of-Thought	3.33	2.79	22.75	24.45	70.50	69.87	**59.39**	**58.91**	38.99	39.01
SFT (Label)	6.67	6.55	45.00	46.25	68.85	69.35	54.31	54.09	43.71	44.06
SFT (Long-CoT)	13.33	11.35	50.00	48.55	72.10	70.21	57.21	55.85	48.16	46.49
ORION	**16.67**†‡	**14.83**†‡§	**55.00**†‡§	**53.51**†‡§	**73.80**†‡	**72.30**†‡	58.86†§	58.24†§	**51.08**†‡§	**49.72**†‡§
Llama3.1-8B-Instruct										
Vanilla LLM	3.33	2.84	22.50	22.50	43.60	41.80	31.23	32.16	25.17	24.83
Wrong-of-thought	0.00	0.00	12.50	9.95	44.00	43.98	31.92	32.97	22.11	21.73
SFT (Label)	3.33	2.28	20.00	21.83	42.80	42.14	30.95	31.18	24.27	24.36
SFT (Long-CoT)	6.67	4.00	25.00	24.50	44.50	43.76	31.83	32.10	27.00	26.09
ORION	**10.00**†‡	**6.80**†‡	**30.00**†‡	**27.00**†‡§	**45.70**†‡	**44.78**†‡	**33.34**†‡§	**33.95**†‡§	**29.76**†‡§	**28.13**†‡§

Table 1: Overall Performance. The **best** results among all methods are highlighted for clarity and emphasis. The symbols †, ‡, and § indicate statistical significance ($p < 0.05$) compared to Vanilla LLM, Wrong-of-Thought, and SFT (Long-CoT), respectively.

5.1 Overall Performance

As shown in Table 1, we compare the overall performance of ORION with baseline models across diverse mathematical reasoning tasks.

Overall, ORION outperforms all baseline models across datasets, demonstrating its effectiveness. Compared with the prompt-based error-aware learning method Wrong-of-Thought (Zhang et al., 2024), ORION achieves more than a 5% improvement, indicating its superior ability to incorporate self-generated errors into model learning. As the evaluation results show, simply incorporating solution errors into prompts during inference may degrade the student's performance. This degradation likely occurs because such errors act as noise that interferes with the reasoning process (Wang et al., 2024b). In contrast to Wrong-of-Thought, ORION provides a more effective mechanism for guiding student models to avoid repeating similar solution errors by refining long-form CoT trajectories from teacher models as supervision signals and then fine-tuning student models.

Regarding different SFT strategies, compared with the Vanilla student model, SFT (Label) yields identical performance, while SFT (Long-CoT) achieves improvements. This suggests that the reasoning trajectories derived from the CoT outputs of the teacher models, such as DeepSeek-R1 (DeepSeek-AI et al., 2025), provide richer reasoning patterns that can guide the student model to imitate during the SFT process. Building upon ORION, the supervision signals are further refined by the student models themselves through self-reflection and the incorporation of self-identified errors, resulting in an additional 2% performance gain. These results further demonstrate the effectiveness of ORION in providing more tailored and effective supervision for student models. Furthermore, to assess generalization, we evaluate ORION on different backbone models, including Qwen2.5-7B-Instruct, Qwen3-8B, and Llama3.1-8B-Instruct. The results consistently show notable improvements across all backbones, demonstrating the generalization ability of ORION in enhancing mathematical reasoning performance. In addition, we provide supplementary results with student models of different scales in Appendix A.3, and further evaluate ORION under different teacher models in Appendix A.4, confirming that our approach remains effective across both student and teacher configurations.

5.2 Ablation Study

As shown in Table 2, we conduct ablation studies to isolate the individual effects of the solution error and self-reflection components in ORION. We compare several ablated variants: ORION w/o Solution Error removes the incorporation of erroneous solutions during reasoning distillation, while ORION w/o Self-Reflection directly uses the long-form CoTs as supervisions without refinement and incorporates solution errors during training.

Overall, both Self-Reflection and Solution Error are effective in improving the quality of long-form

Model	AIME24		AMC23		Math500		GSM-H		Average	
	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10
Qwen3-8B										
SFT (Long-CoT)	23.33	22.50	57.50	56.90	82.90	81.40	59.27	58.82	55.75	54.91
ORION	**26.67**	**25.60**	**62.50**	**61.85**	**83.50**	**82.95**	**59.83**	**59.75**	**58.13**	**57.54**
w/o Solution Error	26.67	24.95	60.00	58.25	83.15	82.10	59.27	59.13	57.27	56.11
w/o Self-Reflection	20.00	20.90	57.50	56.50	82.75	81.25	58.86	58.34	54.78	54.25
Qwen2.5-7B-Instruct										
SFT (Long-CoT)	13.33	11.35	50.00	48.55	72.10	70.21	57.21	55.85	48.16	46.49
ORION	**16.67**	**14.83**	**55.00**	**53.51**	**73.80**	72.30	**58.86**	**58.24**	**51.08**	**49.72**
w/o Solution Error	16.67	10.00	52.50	52.90	73.10	**72.90**	58.24	58.20	50.13	48.50
w/o Self-Reflection	10.00	5.83	50.00	47.86	70.20	69.21	57.35	56.55	46.89	44.86
Llama3.1-8B-Instruct										
SFT (Long-CoT)	6.67	4.00	25.00	24.50	44.50	43.76	31.83	32.10	27.00	26.09
ORION	**10.00**	**6.80**	**30.00**	**27.00**	**45.70**	**44.78**	**33.34**	**33.95**	**29.76**	**28.13**
w/o Solution Error	6.67	5.50	30.00	26.50	45.40	44.34	33.13	32.65	28.80	27.25
w/o Self-Reflection	6.67	4.80	27.50	25.50	43.90	43.10	32.91	31.95	27.75	26.34

Table 2: Ablation Study. We evaluate distilled models trained with different strategies by removing the Solution Error and Self-Reflection components, to quantify the contribution of each component to ORION.

CoT-based supervision signals. Specifically, compared with the SFT (Long-CoT) model, ORION w/o Solution Error achieves over a 1% improvement, demonstrating the effectiveness of the self-reflection mechanism. This mechanism enables the student model to refine long-form CoTs by itself, thereby narrowing the gap between the supervision signal and the capability of the student model. over, the ORION w/o Self-Reflection model also outperforms SFT (Long-CoT), indicating that incorporating self-generated solution errors during the SFT process can guide the student model to avoid repeating potential mistakes. Further, comparing ORION w/o Solution Error with the full ORION, we observe additional gains in reasoning performance, suggesting that considering self-made errors during self-reflection brings complementary benefits. These findings highlight that the advantage of incorporating solution errors can be extended to the CoT refinement process, providing higher-quality and more tailored supervision signals for the student model.

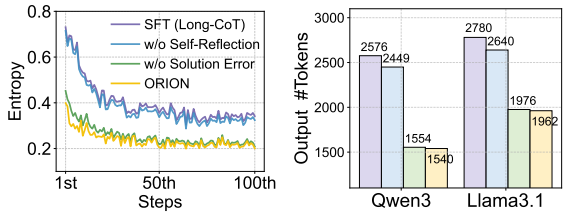
5.3 The Effectiveness of Distilled Models via Different Training Strategies

In this section, we evaluate the training stability and effectiveness of the distilled student models, as shown in Figure 3. Three models, SFT (Long-CoT), ORION w/o Solution Error, and ORION w/o Self-Reflection, are used as baselines for comparison.

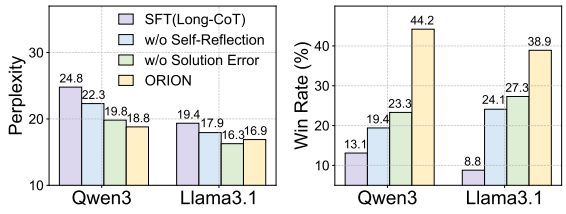
Training Stability. nalyze training stability by plotting the entropy scores in Figure 3(a). Compared with both SFT (Long-CoT) and ORION w/o Self-Reflection, the ORION and ORION w/o Solution Error models exhibit lower entropy scores during training, demonstrating their effectiveness in promoting a more stable training process. This phenomenon further confirms that the self-reflection mechanism benefits the student model's learning process during SFT. Moreover, when incorporating self-generated solution errors, the entropy scores decrease slightly, suggesting that these errors can be effectively utilized to refine long-form CoTs and better align them with the learning capacity of student models. Overall, these results highlight the crucial role of narrowing the gap between teacher-provided supervision and the learning ability of student models in SFT-based distillation.

Effectiveness of Distilled Models. To evaluate the distilled models trained with different strategies, we analyze both the length (Figure 3(b)) and quality (Figures 3(c) and 3(d)) of the generated CoTs.

As shown in Figure 3(b), we prompt each distilled model to generate trajectories and measure their average lengths. The results indicate that ORION and ORION w/o Solution Error produce significantly shorter reasoning trajectories compared with other models. This finding suggests that the self-reflection mechanism helps eliminate redundant reasoning patterns, enabling the student model to learn more concise and efficient reasoning processes. This can help mitigate the overthinking phenomenon (Chen et al., 2024) and reduce inference costs. Next, we assess the quality of the generated CoTs. As shown in Figure 3(c), we collect the generated reasoning trajectories and use the vanilla LLMs as judges to compute the perplexity scores with respect to the ground-truth answers. The results show that ORION achieves the lowest perplexity among all models, demonstrating that

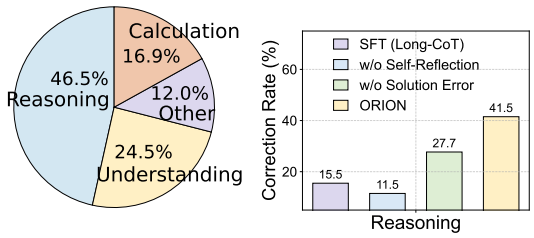


(a) Entropy During Training. (b) Response Length.

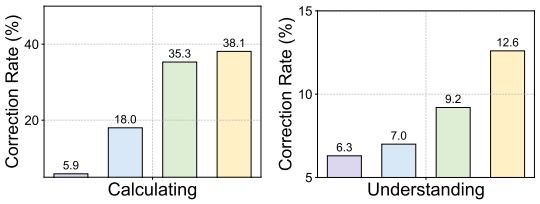


(c) Perplexity Score. (d) GPT Preference Rate.

Figure 3: Performance of Distilled Models under Different Training Strategies. We evaluate distillation entropy (Figure 3(a)), response length (Figure 3(b)), and CoT/final-response quality judged by Vanilla LLMs and GPT-4 (Figure 3(c) and Figure 3(d), respectively).



(a) Error Type Distribution. (b) Reasoning Error.



(c) Calculating Error. (d) Understanding Error.

Figure 4: Analysis of Distilled Models on Different Error Types (Qwen3-8B). Figure 4(a) shows the error-type distribution of vanilla LLMs, and Figures 4(b), 4(c), and 4(d) report the correction rates by error type.

its generated CoTs better align with the true reasoning process and can effectively guide the student model toward correct answers. Furthermore, as shown in Figure 3(d), we employ GPT-4 as an additional evaluator to assess the generated responses based on criteria such as correctness and clarity. Specifically, GPT-4 is prompted to select the best responses among all four models using the prompt templates provided in Appendix A.7. The evaluation results demonstrate that ORION achieves twice the win rate of other models, highlighting its effectiveness in distilling more capable and reliable reasoning models.

5.4 The Performance of Distilled Models on Different Error Types

To validate the effectiveness of our targeted error exposure strategy, we analyze error types and their correction outcomes, as illustrated in Figure 4. Specifically, we first prompt the student model to generate responses on the entire test set and sample 500 instances that are incorrectly answered for further analysis. These sampled instances are then categorized into four distinct error types: reasoning, calculation, understanding, and others, using the prompt template provided in Appendix A.8.

As shown in Figure 4(a), we present the distribution of error categories from the vanilla student model. The analysis reveals that reasoning errors constitute the majority of all mistakes, indicating that students struggle with reasoning-intensive problems. This observation underscores the importance of enhancing the reasoning ability of student models in mathematical problem-solving, such as distilling reasoning capabilities from RLMs. We then evaluate the effectiveness of different distilled models in correcting each error type, as shown in Figures 4(b), 4(c), and 4(d). For reasoning errors, ORION achieves the highest correction rate of approximately 41%, demonstrating its strong ability to help the student model overcome the most common error type. This improvement can be attributed to the refined supervision signals provided by the self-reflection mechanism, which guides the model to internalize reasoning patterns. Furthermore, compared with ORION w/o Solution Error, ORION achieves significantly higher correction rates on both reasoning and understanding errors, indicating that the inclusion of self-generated solution errors further enhances the model's ability to refine supervision signals and avoid repeating similar reasoning or comprehension mistakes.

6 Conclusion

This paper proposes the Error-Aware Self-Reflection (ORION) method, which distills the reasoning capabilities of teacher models into a student model via SFT. ORION leverages the student

model's self-reflection to refine the long-form CoTs generated by teacher models, thereby producing supervision signals that are better matched to the student's capacity and more learnable. Experimental results show that ORION outperforms all baselines across multiple mathematical tasks, validating the effectiveness of error-aware CoT refinement.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 62576082). This work is also supported by the AI9Stars community.

Limitation

Although ORION effectively improves the quality of SFT data by refining teacher-generated reasoning trajectories, the refinement-based approach may still introduce additional errors, as the LLM-driven refinement process is not fully controllable. Furthermore, ORION relies on sampling multiple reasoning trajectories from the student model to expose potential errors, which inevitably incurs extra computational overhead.

References

2025. AIME. aime problems and solutions. Accessed: September 23, 2025.

2025. AMC12. amc 12 problems and solutions. Accessed: September 23, 2025.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *ArXiv preprint*, abs/2310.20689.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *ArXiv preprint*, abs/2412.21187.

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. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, and Haowei Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv preprint*, abs/2510.12948.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *Proceedings of the ICML*, pages 10421–10430.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, and et al. Yibo Miao. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *ArXiv preprint*, abs/2410.07985.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *Proceedings of the ICML*, pages 10764–10799.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *ArXiv preprint*, abs/2105.06643.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *ArXiv preprint*, abs/2305.15717.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv preprint*, abs/2103.03874.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Proceedings of ACL Findings*, pages 8003–8017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Kushal Jain, Piyushi Goyal, and Kumar Shridhar. 2025. Undo: Understanding distillation as optimization. *ArXiv preprint*, abs/2504.02521.

Chengpeng Li, Zheng Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2023. Query and response augmentation cannot help out-of-domain math reasoning generalization. *ArXiv preprint*, abs/2310.05506.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G Patil, Matei Zaharia, et al. 2025a. Llms can easily learn to reason from demonstrations structure, not content, is what matters! *ArXiv preprint*, abs/2502.07374.

Xiaochuan Li, Zichun Yu, and Chenyan Xiong. 2024a. Montessori-instruct: Generate influential training data tailored for student learning. *ArXiv preprint*, abs/2410.14208.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025b. Small models struggle to learn from strong reasoners. *ArXiv preprint*, abs/2502.12143.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. 2024b. Learning from committee: Reasoning distillation from a mixture of teachers with peer-review. *ArXiv preprint*, abs/2410.03663.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. 2024. Are your llms capable of stable reasoning? *ArXiv preprint*, abs/2412.13147.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *ArXiv preprint*, abs/2308.09583.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023b. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *ArXiv preprint*, abs/2308.08747.

OpenAI, Josh Achiam, Steven Adler, and Sandhini Agarwal. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Zhuoshi Pan, Yu Li, Honglin Lin, Qizhi Pei, Zinan Tang, Wei Wu, Chenlin Ming, H Vicky Zhao, Conghui He, and Lijun Wu. 2025. Lemma: Learning from errors for mathematical advancement in llms. *ArXiv preprint*, abs/2503.17439.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of EMNLP*, pages 2695–2709.

Yongqi Tong, Dawei Li, Sizhe Wang, Yujia Wang, Fei Teng, and Jingbo Shang. 2024. Can llms learn from previous mistakes? investigating llms' errors to boost for reasoning. *ArXiv preprint*, abs/2403.20046.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. 2024a. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. *ArXiv preprint*, abs/2402.11651.

Yanbo Wang, Yongcan Yu, Jian Liang, and Ran He. 2025. A comprehensive survey on trustworthiness in reasoning with large language models. *ArXiv preprint*, abs/2509.03871.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024b. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *ArXiv preprint*, abs/2403.05313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS*.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *ArXiv preprint*, abs/2402.09739.

An Yang, Anfeng Li, Baosong Yang, and et al. Beichen Zhang. 2025a. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, and Binyuan Hui. 2024a. Qwen2.5 technical report. *ArXiv preprint*, abs/2412.15115.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E Gonzalez, Bin Cui, and Shuicheng Yan. 2024b. Supercorrect: Advancing small llm reasoning with thought template distillation and self-correction. *ArXiv preprint*, abs/2410.09008.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E. Gonzalez, Bin Cui, and Shuicheng Yan. 2025b. Supercorrect: Advancing small LLM reasoning with thought template distillation and self-correction. In *Proceedings of ICLR*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *ArXiv preprint*, abs/2502.03387.

Huifeng Yin, Yu Zhao, Minghao Wu, Xuanfan Ni, Bo Zeng, Hao Wang, Tianqi Shi, Liangying Shao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. Marco-o1 v2: Towards widening the distillation bottleneck for reasoning models. *ArXiv preprint*, abs/2503.01461.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, and Xi Victoria Lin. 2022. Opt: Open pre-trained transformer language models. *ArXiv preprint*, abs/2205.01068.

Yongheng Zhang, Qiguang Chen, Jingxuan Zhou, Peng Wang, Jiasheng Si, Jin Wang, Wenpeng Lu, and Libo Qin. 2024. Wrong-of-thought: An integrated reasoning framework with multi-perspective verification and wrong information. In *Proceedings of EMNLP Findings*, pages 6644–6653.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2021. Bert learns to teach: Knowledge distillation with meta learning. *ArXiv preprint*, abs/2106.04570.

Dataset	Train	Test
OpenR1-Math-220k (2025)	10,000	-
GSM-Hard (2023)	-	1,319
MATH500 (2021)	-	500
AIME24 (2024)	-	30
AMC23 (2024)	-	40

Table 3: Data Statistics.

Models	Error Num	Correct Num	Ratio
Qwen2.5-7B	49,146	30,219	0.61
Qwen3-8B	38,470	27,135	0.71
Llama3.1-8B	85,874	51,896	0.60

Table 4: Refinement Statistics for Different Student Models.

Methods	AIME24	AMC23	MATH500	GSM-H
Llama3.2-3B-Instruct				
Vanilla	6.67	32.50	39.80	23.52
SFT	6.67	35.00	39.40	24.39
ORION	**10.00**	**40.00**	**43.60**	**26.71**
Qwen3-14B				
Vanilla	26.67	72.50	88.10	63.84
SFT	26.67	75.00	88.50	63.19
ORION	**30.00**	**75.00**	**89.70**	**64.92**

Table 5: Performance of ORION on Models of Different Sizes. We provide additional results on Llama3.2-3B and Qwen3-14B to examine how ORION generalizes to models with different parameter scales.

A Appendix

A.1 License

We provide the licenses for the datasets we use. OpenR1-Math-220k and MATH500 are licensed under the Apache License 2.0. AIME24 and AMC23 are not currently labeled with license types. GSM-Hard uses the MIT license. All of these licenses and agreements allow their data for academic use.

A.2 Additional Implementation Details

In this section, we provide the detailed data statistics in ORION. We build our training set by using 10,000 examples randomly sampled from OpenR1-Math-220k (Hugging Face, 2025) as seed data and then expand it to form our final training data. For evaluation, we use the GSM-Hard dataset containing 1,319 examples, MATH500 with 500 examples, AIME24 with 30 examples, and AMC23 with 40 examples. All statistics are shown in Table 3.

To clarify that the collected erroneous data can be successfully refined by the student model, we further report refinement statistics for each student model in Table 4. Specifically, Error Num denotes the number of erroneous cases sampled from the student model; Correct Num denotes the number of cases where the student produces a correct refinement when conditioned on the corresponding teacher CoT; and Correct Ratio is the proportion of refined samples that pass answer verification and are therefore included in the final training set. These statistics provide a direct view of how refinement contributes to ORION's training corpus. Overall, we observe a relatively high correction ratio across student models, indicating that the student-driven refinement process is effective at converting erroneous attempts into usable, verified supervision.

A.3 Additional Results with Different Student Models

To further evaluate the scalability and robustness of ORION, we conduct additional experiments on models of different sizes, including the smaller Llama3.2-3B (Grattafiori et al., 2024) and the larger Qwen3-14B (Yang et al., 2025a). All training settings remain consistent with those used for the 7B-scale experiments. As shown in Table 5, ORION yields consistent improvements across model scales, demonstrating that the proposed framework generalizes effectively to both lower-capacity and higher-capacity models.

A.4 Additional Results with Different Teacher Models

To examine whether ORION depends on a particular teacher model or a specific teacher–student performance gap, we further evaluate the framework using two additional teacher models of different scales: QwQ-32B and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025). These models represent substantially different capacities compared to the primary teacher model, DeepSeek-R1, used in the main experiments.

As shown in Table 7, ORION delivers consistent performance gains across all teacher configurations. The results obtained with QwQ-32B and DeepSeek-R1-Distill-Qwen-7B are comparable to those achieved with DeepSeek-R1, indicating that the effectiveness of ORION is not tied to a specific teacher model. This reinforces our claim that ORION is broadly applicable and remains effective regardless of the teacher model's scale or initial performance.

Methods	AIME24	AMC23	MATH500	GSM-H
Qwen2.5-7B-Instruct				
Vanilla	10.00	47.50	69.80	55.72
SFT	13.33	50.00	72.10	57.21
ORION	**16.67**	**55.00**	**73.80**	**58.86**
ORION (S)	16.67	52.50	73.40	58.52
Llama3.1-8B-Instruct				
Vanilla	3.33	22.50	43.60	31.23
SFT	6.67	25.00	44.50	31.83
ORION	**10.00**	30.00	**45.70**	**33.34**
ORION (S)	10.00	**32.25**	44.80	32.97

Table 6: Results for ORION under a Matched Data Budget. ORION (S) denotes a subset of ORION's refined training data, sampled to match the data volume of SFT (Long-CoT).

A.5 Effect of Data Expansion on Performance

To further examine whether the gains of ORION are merely due to training data augmentation, we conduct a controlled experiment with a matched data budget. Specifically, although ORION may generate multiple refined CoTs per question through error-exposure sampling, we uniformly subsample the refined outputs to construct an SFT set with the same number of training instances as the long-CoT distillation baseline (10,000 examples). We denote this setting as ORION (sample). As shown in Table 6, under an identical training data scale, ORION (sample) still consistently outperforms the long-CoT baseline and achieves performance comparable to the full ORION. These results indicate that the improvements primarily stem from the error-aware refinement that makes teacher reasoning trajectories more learnable for the student, rather than from simply increasing the amount of training data.

A.6 Prompt Templates Used in ORION

We detail the prompts employed across different stages of ORION. As shown in Figure 5, the Error Exposure stage uses prompts that instruct the model to solve problems, from which incorrect cases are collected. These errors are then utilized in the Self-Reflection stage, where the prompt templates in Figure 6 instruct the model to identify and correct solution errors while refining the long-form CoTs. Finally, during the Supervised Fine-Tuning stage, the prompt in Figure 7 leverages the refined trajectories with their corresponding solution errors as training supervision.

Methods	AIME24	AMC23	MATH500	GSM-H
QWQ-32B				
Qwen2.5-7B-Instruct				
Vanilla	10.00	47.50	69.80	55.72
SFT	16.67	50.00	73.10	55.54
ORION	**16.67**	**52.50**	**75.40**	**57.72**
Llama3.1-8B-Instruct				
Vanilla	3.33	22.50	43.60	31.23
SFT (Long-CoT)	6.67	25.00	45.80	32.79
ORION	**10.00**	**30.00**	**46.20**	**33.17**
DeepSeek-R1-Distill-Qwen-7B				
Qwen2.5-7B-Instruct				
Vanilla	10.00	47.50	69.80	55.72
SFT	13.33	50.00	71.20	57.41
ORION	**16.66**	**52.50**	**73.40**	**59.39**
Llama3.1-8B-Instruct				
Vanilla	3.33	22.50	43.60	31.23
SFT	6.67	25.00	44.80	33.05
ORION	**6.67**	**27.50**	**48.40**	**35.58**

Table 7: Performance of ORION with Different Teacher–student Configurations. We additionally evaluate ORION using QwQ-32B and DeepSeek-R1-Distill-Qwen-7B as teacher models to assess its robustness under varying teacher scales.

A.7 Prompt Templates Used for Evaluating the Quality of ORION

As shown in Figure 8, we present the prompt templates used to evaluate the quality and educational value of the CoTs synthesized by four methods: SFT (Long-CoT), ORION w/o Solution Error, ORION w/o Self-Reflection, and ORION. The evaluation is conducted using GPT-4 (OpenAI et al., 2023) as the evaluator, which assesses each generated reasoning trajectory based on its clarity, correctness, and pedagogical usefulness. We employ consistent prompt templates across all methods to ensure fair comparison and reproducibility.

A.8 Prompt Templates Used for Solution Error Categorization

As shown in Figure 9, we present the prompt templates used to classify the solution errors generated by GPT-4 (OpenAI et al., 2023). The prompt instructs the evaluator model to classify each incorrect response into one of four predefined categories: Reasoning Error, Understanding Error, Calculation Error, or Other.

A.9 Case Study

In this section, we present a detailed case study across two tables to intuitively demonstrate the core process and the resulting effectiveness of ORION.

As shown in Table 8, we illustrate the internal mechanism of our method. It presents the error exposure and self-reflection process. Building on this, Table 9 contrasts the final response generated by ORION with those from baseline methods, showing the superior quality and clarity of its output.

First, we examine a relatively simple math problem to analyze the behavior of ORION during the error exposure phase. Despite the low complexity of the problem, sampling multiple responses reveals a diverse range of errors, including a hallucination for general concepts ("lower class soldiers (8) is more than the available (10)"), misunderstandings of the problem statement ("exceeding their respective limits") and basic computational mistakes ("C(8,10)=90"). Therefore, we argue that it is crucial to expose as many diverse erroneous responses as possible for each problem and to use them as guidance to avoid similar pitfalls in the final reasoning. Then, we present a complete example of the Self-Reflection process. Given a mathematical problem, an incorrect solution, and a standard long-form CoT from an RLM, the model generates an Error-Aware Chain-of-Thought that not only analyzes the key mistakes in the solution error but also refines the standard reasoning trajectory to make it better aligned with the model's own capacity.

As shown in Table 9, we contrast the final responses to illustrate the superior reasoning capability of the full ORION model. The ORION w/o Self-Reflection provides a direct solution but lacks any analysis of potential pitfalls, resulting in an incorrect response. The ORION w/o Solution Error produces an analysis of possible errors; however, due to the absence of error examples in its training, it ultimately fails to deliver a correct answer. In stark contrast, the complete ORION model exhibits self-awareness by first identifying a potential failure point, noting that the solution could be flawed ("because the expression for n was not properly simplified or checked for integer solutions."). Building on this crucial analysis, it then presents a clear, refined solution, demonstrating its ability to anticipate and preempt common errors.

Instruction for Error Exposure

Please reason step by step, and put your final answer within boxed{}.

Problem: {Problem}

Figure 5: The Prompt Templates Used for Error Exposure.

Instruction for Reasoning Refinement

You are given a math problem, an incorrect solution, and a long reasoning trace. Your task has two steps:

1. *Error Analysis*
- Identify the key mistake in the incorrect solution.
- Explain briefly why this mistake is wrong.

2. *Refined Reasoning*
- Based on the provided long reasoning trace, extract only the *essential and correct steps*.
- Keep the reasoning clear, concise, and logically coherent.

Problem:: {Problem}

Incorrect Solution:: {Error}

Long Reasoning Trace (to refine):: {Long_Reasoning_trace}

Output Format (strict)

Error Analysis
- [Briefly explain the key error]

Correct Reasoning
[Step-by-step refined reasoning, clear and concise, avoiding the previous error]
Final Answer: \\boxed{{...}}

Figure 6: The Prompt Templates Used for Reasoning Refinement.

Instruction for Supervised Fine-Tuning

Instruct: Please reason step by step, and put your final answer within boxed{}.

Input: *Problem*: {Problem} + *Incorrect Solution*: {Error}

Output: {Error-Aware Chain-of-Thought}

Figure 7: The Prompt Templates Used for Supervised Fine-Tuning.

Instruction for GPT-Based Score

You are given three different reasoning chains that attempt to solve the same math problem. Your task is to evaluate these four reasoning chains based on the following three criteria and chose the best reasoning chain:

1. *Correctness*: Whether the reasoning is logically valid and leads to the correct final answer.
2. *Clarity*: Whether the reasoning is clear, well-structured, and easy to follow.
3. *Completeness*: Whether the reasoning covers all necessary steps and avoids unjustified leaps.

Only output the final choice without explanations, in the following format:

Best Reasoning Chain: {A/B/C/D}

Figure 8: The Prompt Templates Used in GPT Preference Evaluation.

Instruction for Error Type Categorization

You are given a mathematical problem, and an incorrect solution process. Your task is to analyze the incorrect solution process and categorize the type of error made. The possible error types are:

1. *Understanding Error* – The model misunderstood the problem or misinterpreted the instructions.
2. *Reasoning Error* – The model made an error in logical reasoning or approach.
3. *Calculation* – Arithmetic mistakes or errors in calculations.
4. *Others* – Any other type of mistake that doesn't fall into the categories above.

Based on the analysis, output only the most appropriate error type. Do not include any additional explanations or content.

Problem: {problem}
Incorrect Solution: {Incorrect Solution}

Figure 9: The Prompt Templates Used for Error Type Categorization.

Case 1: The Case of Error Exposure via Response Sampling

Question: The Greek army contained two types of soldiers: the upper class and the lower class soldiers. If there were a total of 5 upper class soldiers, and 10 lower class soldiers in a certain part of Athens, and the battle of Thermopylae demands a force of 4 upper class soldiers and 8 lower class soldiers, how many different battalions can be sent?

Ground Truth: 225

Error Response[0]: To solve this problem, we need to determine how many different battalions can be formed given the constraints on..., Since the requirement for **lower class soldiers (8) is more than the available (10)**, it is impossible to...**So the final answer is boxed{0}.**

Error Response[1]: To solve this problem, we need to determine how many different battalions can be formed given the constraints on..., Since both the upper class and lower class soldiers allow us to form exactly one battalion without ...**exceeding their respective limits...So the final answer is boxed{1}.**

Error Response[2]: To solve this problem, we need to determine how many different battalions can be formed given the constraints on the number of upper class and lower class soldiers available... For lower class soldiers, the number of ways to choose 8 from 10 is mistakenly calculated as **C(10,8)=90**, Then, the total number of battalions is $5 \times 90 = 450$. **So the final answer is boxed{450}.**

Case 2: The Case of Reasoning Refinement via Error-Aware Self-Reflection

Question: Let \$a, b, c\$ be distinct numbers such that the equations $x^2 + ax + 1 = 0$ and $x^2 + bx + c = 0$ have a common real root, and the equations $x^2 + x + a = 0$ and $x^2 + cx + b = 0$ also have a common real root. Compute the sum $a + b + c$.

Ground Truth: -3

Error Response: To solve for $a + b + c$, we need to analyze the given conditions step by step. Let the common real root of the equations $x^2 + ax + 1 = 0$ and $x^2 + bx + c = 0$ be r; substituting r into both equations gives $r^2 + ar + 1 = 0$ (1) and $r^2 + br + c = 0$ (2). Subtracting (1) from (2) yields $(b-a)r + (c-1) = 0$, so $r = \frac{1-c}{b-a}$ (3). Similarly, let the common real root of $x^2 + x + a = 0$ and $x^2 + cx + b = 0$ be s; substituting s gives $s^2 + s + a = 0$ (4) and $s^2 + cs + b = 0$ (5). Subtracting (4) from (5) yields $(c-1)s + (b-a) = 0$, so $s = \frac{a-b}{c-1}$ (6). Since $r = s$, equating (3) and (6) gives $\frac{1-c}{b-a} = \frac{a-b}{c-1}$; cross-multiplying gives $(1-c)(c-1) = (b-a)^2$, or $-(c-1)^2 = (b-a)^2$. Since a, b, c are distinct, $(b-a)^2 \neq 0$, so $(c-1)^2 = (b-a)^2$, meaning $c - 1 = \pm(b-a)$. Considering $c = b - a + 1$ and substituting back, we find $r = -1$ from (3). Substituting $r = -1$ into $x^2 + x + a = 0$ gives $1 - 1 + a = 0$, so $a = 0$. Then $c = b + 1$; checking consistency with $x^2 + cx + b = 0$ confirms validity, giving $b = -1$ and $c = 0$. **Thus, the final answer is boxed{-1}.**

Reasoning Refinement: *Error Analysis* : The incorrect solution **incorrectly assumed that the common roots α and β must be the same.** leading to a contradiction with the requirement that a, b, c are distinct. Specifically, it derived $\beta = \frac{1-c}{a-b}$ and $\alpha = \frac{c-1}{b-a}$, then concluded $\alpha = \beta$, leading to $\beta = 1$, $\alpha = 1$, $a = -2$, and $b = -1 - c$. This made a, b dependent on c, conflicting with the distinctness condition.

Correct Reasoning : Let's start by carefully defining the problem to avoid confusion. We need to find $a + b + c$ where a, b, c are distinct numbers, with two pairs of quadratic equations sharing common real roots. First, let's clarify which roots are common—they don't have to be the same across both pairs, **which was a key mistake in the incorrect solution.**

First, define the common roots properly: Let the common real root of $x^2 + ax + 1 = 0$ and $x^2 + bx + c = 0$ be α; this means α satisfies both equations. Separately, let the common real root of $x^2 + x + a = 0$ and $x^2 + cx + b = 0$ be β, so β satisfies these two equations. Note that α and β might not be equal—this distinction is crucial.

Next, use Vieta's Formulas to relate the roots and coefficients of each quadratic equation. For a quadratic equation $x^2 + px + q = 0$, the sum of roots is $-p$ and the product is q...

Now, let's connect these equations using the fact that α and β satisfy their respective pairs of equations. For α in both quadratics, subtracting the equations gives $(b-a)\alpha + (c-1) = 0 \implies \alpha = \frac{c-1}{b-a}$. For β, subtracting its pair gives $(1-c)\beta + (a-b) = 0 \implies \beta = \frac{b-a}{c-1}$. Notice that $\alpha \cdot \beta = 1$—interesting, but let's focus on α.

Let's simplify by testing specific values for α. Suppose $\alpha = 1$: substituting into $x^2 + ax + 1 = 0$ gives $1 + a + 1 = 0 \implies a = -2$. For $x^2 + bx + c = 0$, $1 + b + c = 0 \implies b = -1 - c$.

Now check the second pair with $a = -2$: $x^2 + x - 2 = 0$ factors to $(x+2)(x-1) = 0$, so β is 1 or -2. If $\beta = 1$, substituting into $x^2 + cx + b = 0$ gives $1 + c + b = 0$, which holds since $b = -1 - c$. If $\beta = -2$, we get $c = 1$ and $b = -2$, but this makes $a = b$, violating distinctness—so $\beta = 1$ is valid.

Ensuring a, b, c are distinct (e.g., $c \neq 1, -2$), the sum is $a + b + c = -2 + (-1 - c) + c = -3$.

Thus, the final answer is boxed{-3}.

Table 8: Case Study. Text in **red** indicates errors and their causes in the response, while text in **blue** marks key errors identified by ORION and the corresponding correct solutions.

Case 3: Comparison of Responses from Different Baseline Methods

Question: Please reason step by step, and put your final answer within boxed{}. If m and n are integers such that $3m + 4n = 100$, what is the smallest possible value of $|m - n|$?

Ground Truth: 3

Vanilla Response: To find the smallest possible value of $|m - n|$ given $3m + 4n = 100$, we solve for m in terms of n: $m = \frac{100-4n}{3}$. For m to be an integer, $100 - 4n$ must be divisible by 3. Since $100 \equiv 1 \pmod 3$ and $-4n \equiv -n \pmod 3$, **we require $n \equiv 0 \pmod 3$.** Let $n = 3k$. Substituting gives $m = \frac{100-12k}{3} = 33 - 4k$, so both m and n are integers: $m = 33-4k, n = 3k$. Then $|m-n| = |(33-4k)-3k| = |33-7k|$. Minimizing $|33 - 7k|$ reduces to choosing k nearest to $33/7 \approx 4.714$. For $k = 4$, $(m, n) = (17, 12)$ and $|m - n| = 5$. For $k = 5$, $(m, n) = (13, 15)$ and $|m - n| = 2$. **Hence, the smallest possible value is boxed {2}.**

ORION w/o Self-Reflection: Hmm, okay, let's try to express n in terms of m: $n = \frac{100-3m}{4}$. For n to be an integer, $100 - 3m$ must be divisible by 4. Since $100 \equiv 0 \pmod 4$, we need $3m \equiv 0 \pmod 4$, and because 3 and 4 are coprime, m must be a multiple of 4. Let's set $m = 4k$, giving $n = 25 - 3k$. Then $|m - n| = |4k - (25 - 3k)| = |7k - 25|$... To minimize this, k should be near $25/7 \approx 3.57$. Checking $k = 3$ gives $(m, n) = (12, 16)$ and $|m - n| = 4$; checking $k = 4$ gives $(m, n) = (16, 13)$ and $|m - n| = 3$. **Both are small, but the smaller one seems to be 4 when $k = 3$... Thus, the final answer is boxed{4}.**

ORION w/o Solution Error: ## Error Analysis: The main mistake in the original solution was incorrectly simplifying the equation $3m + 4n = 100$ when attempting to express n in terms of m, leading to non-integer values for m and n.

Correct Reasoning: We want to minimize $|m - n|$ given $3m + 4n = 100$. Solving for m, $m = \frac{100-4n}{3}$. Suppose n is chosen to make m an integer. Trying $n = 3k$, we get $m = \frac{100-12k}{3} = 33 - 4k$. Approximating, we take $m \approx 33 - 4k$. Then $|m - n| \approx |33 - 7k|$. **If $k = 5$, this gives $m = 13, n = 15$, so $|m - n| = 2$. Thus, the smallest possible value is boxed {2}.**

ORION: ## Error Analysis: The possible mistake was the incorrect handling of $3m + 4n = 100$ when solving for m, n. The substitution and simplification produced a wrong expression for n, leading to non-integer values. The error arose **because the expression for n was not properly simplified or checked for integer solutions.**

Correct Reasoning: We aim to minimize $|m - n|$ given $3m + 4n = 100$. Expressing n in terms of m, $n = \frac{100-3m}{4}$. For n to be integer, $100 - 3m$ must be divisible by 4. Since $100 \equiv 0 \pmod 4$, this requires $3m \equiv 0 \pmod 4$, so $m = 4k$. Substituting gives $n = 25 - 3k$, $m = 4k$. For non-negative integers, $0 \leq k \leq 8$. Then $|m - n| = |4k - (25 - 3k)| = |7k - 25|$. Evaluating: $k = 3 \Rightarrow |21 - 25| = 4$, $k = 4 \Rightarrow |28 - 25| = 3$, others yield larger values. **Thus the minimum is boxed {3}.**

Table 9: Case Study. Text in **red** indicates errors and their causes in the response, while text in **blue** marks key errors identified by ORION and the corresponding correct solutions.