Escaping the *Echo Trap*: On Credit Assignment Failure in Multi-turn LLM Self-Reflection

Linxuan Du[1]*, Guangquan Xue[1]*, Xiaobo Liang[1], Qipeng Huang[1], Yuyang Ding[1], Xinyu Shi[1], Yijun Zhang[2], Ji Qi[2], Wenpeng Zhu[2], Juntao Li[1], Min Zhang[1]

[1]Soochow University

[2]China Mobile (Suzhou) Software Technology Co., Ltd. Suzhou 215000, China

{lxdu, gqxue}@stu.suda.edu.cn, ljt@suda.edu.cn

{qiji, zhuwenpeng}@cmss.chinamobile.com, yijzhangme@gmail.com

Abstract

Despite the potential of multi-turn self-reflection to improve LLM reasoning, its effectiveness in practice is severely constrained by a failure mode we term the ***Echo Trap***. Specifically, this phenomenon gives rise to two coupled problems: (1) the model becomes limited by its inherent capabilities and tends to repeat earlier reflections to preserve reward signals; (2) once such "copy" behavior is reinforced, the model ceases to try new strategies, leading to exploration collapse. We hypothesize that this issue is largely driven by *imprecise credit assignment* during training, as standard GRPO assigns rewards at the trajectory level, making it difficult to distinguish which reflection steps contribute to improved outcomes. To address this limitation, we propose a tree-structured extension of GRPO for multi-turn self-reflection, which enables more accurate advantage estimation than standard trajectory-level GRPO. Through extensive experiments, we analyze the *Echo Trap* and demonstrate that our method effectively mitigates behavior collapse and improves performance across multiple benchmarks. Our code is available at https://github.com/Flowersea37/TRAE

1 Introduction

The capacity for reflection (Huang et al., 2024; Song et al., 2024) is fundamental to the continuous improvement of Large Language Models (LLMs). Reflection allows models to employ trial-and-error strategies to recover from execution failures and further enhance performance (Bensal et al., 2025; Shinn et al., 2023). Most Reinforcement Learning from Verifiable Rewards (RLVR) methods elicit *implicit reflection* by providing rule-based outcome rewards (Wen et al., 2025; Guo et al., 2025a), such as emergent 'Aha moments'. These moments guide reflective chain-of-thought (CoT) generation without process-level supervision. In contrast, *explicit reflection* methods adjust behavior using immediate feedback, overcoming the lack of process reward in RLVR and offering greater potential. However, explicit reflection is often hindered by *behavior collapse*, where the model typically improves its output only in the initial reflection round, while performance degrades over subsequent iterations. This phenomenon is commonly referred to as the ***Echo Trap*** problem.

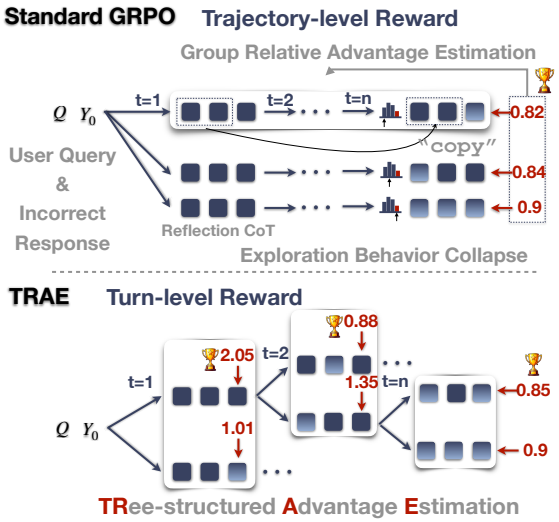

Figure 1: Standard GRPO relies on trajectory-level rewards and may suffer from behavior collapse. In contrast, TRAE achieves turn-level advantage estimation within trajectories using turn-level rewards estimated by tree-structured rollouts, enabling steady improvement while preserving revision capability.

Previous work addressing behavior collapse has primarily focused on improving each reflection turn independently and on early stopping to terminate iterations and prevent repetitive outputs (Kumar et al.; Jiang et al., 2025). However, these approaches largely avoid the core challenges of multi-turn reflection rather than enhancing the effectiveness of the reflection process itself. In Section 3, we analyze failures in multi-turn reflection and identify the main causes: 1) *Limited Refinement Capability*: The model's ability to improve

*Equal contribution

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35393–35405

outputs is constrained by its inherent capacity. After the most obvious errors in the first turn are corrected, further improvements require increasingly extensive exploration or rejected sampling. 2) *Copy Behavior Dominance*: During RL training, the model often defaults to repeating previous outputs. This occurs because repeating a partially correct response offers a reliable high-reward signal, especially when exploratory attempts fail to produce improved outputs. These entangled factors lead to behavior collapse, where the model's tendency to explore diminishes over successive reflection turns.

GRPO and its variants typically rely on trajectory-level rewards for policy optimization (Shao et al., 2024), which can hinder effective RL training, as illustrated in Figure 1. This training paradigm introduces a severe *credit assignment problem*: the model tends to overemphasize final outcomes while neglecting the contributions of critical intermediate decision steps. As a result, learning signals for reflective behaviors become coarse and misaligned, making it difficult to effectively encourage meaningful multi-turn self-reflection. In this work, we propose **TRAE**, a tree-structured extension of GRPO that introduces turn-level advantage estimation, enabling the policy to optimize cumulative discounted returns by accurately attributing credit to each reflection step. Furthermore, TRAE applies group-relative advantage estimation at each layer of the tree, which alleviates reward inconsistency across different time steps. This design allows for turn-level attribution of discounted returns to individual reflection steps, enabling the model to more accurately assess how each decision contributes to long-term performance.

We conduct extensive experiments and analyses of TRAE on six widely adopted benchmarks for mathematical reasoning. TRAE substantially improves the Qwen3-8B-Base model, raising average accuracy from 36.1% to **56.2%** and consistently surpassing the SFT, GRPO, and DAPO baselines. Furthermore, under iterative reflection, the base model's performance declines across three rounds, whereas TRAE steadily improves, achieving an average gain of **9.5%**, outperforming all baselines. These results demonstrate TRAE's effectiveness in both overall performance and multi-turn reflection. In summary, our contributions are as follows:

- We demonstrate that applying RL to multi-turn self-reflection introduces a substantial risk of behavior collapse, where exploration progressively degrades and the policy converges to suboptimal, repetitive strategies.
- We propose TRAE, a tree-structured variant of GRPO that incorporates turn-level advantage estimation. By enabling turn-level advantage estimation over multi-turn reasoning trajectories, TRAE effectively mitigates behavior collapse and allows performance to continuously improve across reflection rounds.
- We further demonstrate that TRAE preserves behavioral diversity, which is essential for achieving a better trade-off between exploration and exploitation.

2 Related Work

Self-correction. Prior studies have shown that LLMs can leverage self-correction capabilities without relying on external feedback to enhance reasoning quality. However, naively prompting LLMs for self-correction can reduce performance (Tyen et al., 2024; Qu et al., 2024; Huang et al., 2023). Several other approaches (Zhang et al., 2025b; Qu et al., 2024; Ye et al., 2023) have attempted to leverage supervised fine-tuning by imitating self-correction data filtered from human or stronger model. However, this approach often leads to degraded self-correction ability due to distribution mismatch. Reinforcement learning (RL) methods have been proposed to address these challenges (Kumar et al.; Ma et al., 2025; Guo et al., 2025a). However, naively applying RL methods can lead to behavior collapse where LLMs tend to generate the first response of high quality and minimal refinement from the first. To mitigate the challenge, Kumar et al. proposed a warm-up stage that constrains first-attempt outputs to remain close to the base model while optimizing second-attempt accuracy. Other approaches, such as self-verification (Jiang et al., 2025; Bensal et al., 2025), teach LLMs terminate self-correct at appropriate times. However, those methods primarily avoid behavior collapse without addressing it.

Credit Assignment Credit assignment refers to associate actions with their long-term consequencesm,which has been a crucial problem in RL for a long time (Pignatelli et al., 2023). Recent work has explored methods to improve credit assignment at the step level. For instance, GIGPO (Feng et al., 2025) extracts specific steps

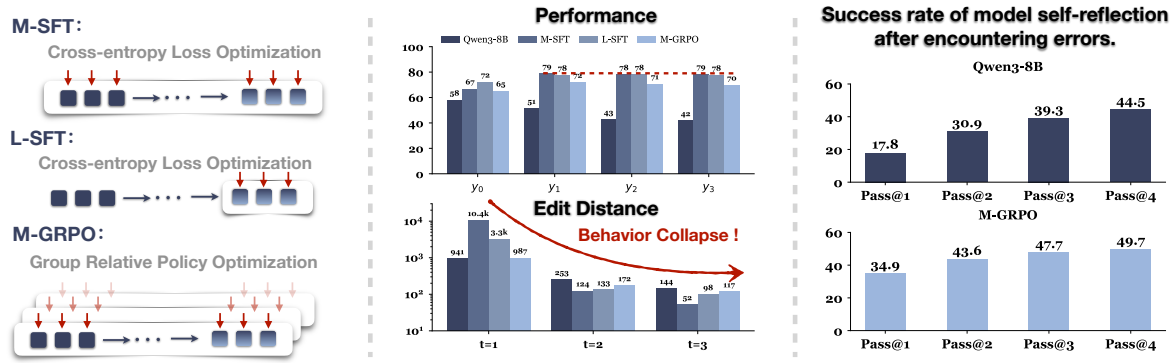

Figure 2: **Left**: Training methods for Qwen3-8B-Base and SFT and GRPO variants, along with their evaluation results on MATH-500. **Middle**: Detailed performance trends and edit distance changes. **Right**: Success rate of error correction when these models encounter incorrect samples.

from trajectories for targeted training, while MT-PPO (DESIGN, 2025) and TDRM (Zhang et al., 2025a) investigate how to balance process rewards and outcome rewards. Another line of research leverages Monte Carlo estimation to obtain more accurate advantage signals. VinePPO (Kazemnejad et al., 2024) and TreeOPO (Huang et al., 2025a) replace the value model in PPO with Monte Carlo estimation. TreeGRPO (Ji et al., 2025), TreeRPO (Yang et al., 2025), and TreePO (Li et al., 2025) employ tree-based rollouts to form groups within a tree structure, enabling a more global sequence-level credit assignment. However, these approaches do not explicitly address turn-level credit assignment in multi-turn interaction scenarios. Segment-level methods such as SPO (Guo et al., 2025b), TreeRL (Hou et al., 2025), and TEMPO (Tran et al., 2025) employ tree-based sampling to estimate segment-level advantages, primarily relying on value function differences across segments. In contrast, we adopt a tree-based sampling strategy and provide explicit turn-wise credit assignment for the model.

3 Preliminary Study

We first formalize the task of multi-turn self-reflection. As demonstrated in prior work (Jiang et al., 2025; Qu et al., 2024), this task requires models to perform iterative refinement, ideally ensuring that each successive modification leads to continuous performance improvement. Through empirical investigation, we observe "***behavior collapse***" during both SFT and GRPO training. Finally, we conduct an in-depth analysis of the underlying causes of this degradation and outline potential directions for improvement.

3.1 Multi-turn Self-reflection

We formalize multi-turn self-reflection as an iterative refinement process in which a model repeatedly evaluates and improves its own outputs over multiple turns. Given a query q and an initial response y_0, the policy π_θ generates a sequence of refined outputs $\{y_1, \ldots, y_n\}$. Each iteration y_i contains both a self-generated critique and a subsequent refinement based on y_{i-1}. Formally, the sequence must exhibit monotonic reward improvement, such that $\mathcal{R}(y_i) > \mathcal{R}(y_{i-1})$, where $\mathcal{R}(\cdot)$ denotes a reward function (e.g., rule-based correctness in mathematical reasoning). For simplicity, we treat the critique and its subsequent refinement as a single output block y_i in the remainder of our analysis.

3.2 Extensive Trials

Model Training To investigate the underlying challenges in training multi-turn self-reflection, we employ Qwen3-8B-Base as our base model and analyze its performance under both SFT and GRPO paradigms. Specifically, we utilize Qwen3-32B-FPB as the teacher model to synthesize 10K distillation data. Through rejection sampling, we curate training trajectories that begin with an erroneous initial response y_0 and are followed by a sequence of progressively corrected refinements $\{y_i\}$. To ensure computational feasibility and adhere to the 32k context length limit, the maximum reflection horizon is constrained to three turns, i.e., $i \in \{1, 2, 3\}$. We conduct a comparative analysis across three training configurations:

- **M-SFT**: The model is supervised on the entire multi-turn refinement trajectory $\{y_0, y_1, \ldots, y_n\}$, explicitly training the model to perform iterative self-reflection.

- **L-SFT**: Supervision is applied only to the final refined response y_n, ignoring intermediate reflection steps, serving as a control to evaluate whether explicit multi-turn supervision is necessary.
- **M-GRPO**: An RL-based approach that optimizes the policy with multi-turn group-relative rewards.

Model Inference During inference, we also perform three-step self-reflection to evaluate the outcome at each iteration. We compare outputs produced by different training methods and compute the edit distance, measured as the average edit distance between consecutive outputs y_i and y_{i-1}, to quantify the degree of modification introduced at each reflection step.

3.3 Analysis and Possible Improvements

Observation All results are shown in Figure 2, from which we can clearly observe: 1) *All training methods fail to sustain performance over multiple turns, with edit distance decreasing exponentially.* 2) *M-SFT shows severe repetition during inference: nearly all responses stop after reaching the maximum length. While the responses contain the complete content, they repeat continuously.* 3) *Error correction remains difficult even with repeated attempts: the success rate of correcting initially incorrect responses stays relatively low, indicating that meaningful improvements are sparse and often require substantial trial-and-error.*

We attribute these observations to several factors. First, during multi-turn training, M-SFT generates substantial redundant content, which biases the model toward repetitive outputs. In contrast, while M-GRPO also employs multi-turn training, it mitigates repetition by penalizing redundant content through negative feedback during optimization. However, despite suppressing textual repetition, M-GRPO fails to enhance overall performance and even underperforms the SFT baseline. This behavior likely stems from the inherent limitations of the Qwen3-8B-Base backbone, which lacks proficiency in multi-turn revision. Consequently, the training distribution becomes dominated by repetitive patterns, hindering the model's capacity to explore alternative strategies. Furthermore, our evaluation of error-correction capabilities reveals that pass@4 remains low even as the number of samples increases, suggesting that the model requires significant trial-and-error to achieve meaningful improvements. Because meaningful improvements are sparse, trajectory-level rewards become particularly problematic: when only a small subset of reflection turns actually contributes to the final outcome, assigning the same optimization signal to the entire trajectory can misattribute credit and reinforce unhelpful or repetitive revisions.

> ***Motivation:*** Based on the above analysis and findings, one possible approach to addressing multi-turn training issues is to optimize the model through turn-level advantage estimation, i.e., guiding the model to avoid behavior collapse.

4 Method

To address behavior collapse in multi-turn reflection, we propose a method that assigns distinct advantages to different turns within a trajectory, enabling targeted optimization at the turn level, as illustrated in Figure 3. Our approach consists of three steps: (1) designing a more accurate turn-level reward estimation for multi-turn reflection; (2) obtaining this estimation via tree-structured rollouts; and (3) assigning turn-specific advantages based on the rewards computed in the previous steps.

4.1 Turn Reward Estimation

In multi-turn reflection, each turn produces an intermediate response y, whose correctness can be explicitly evaluated. We therefore define the *immediate reward* for y as follows:

$$r^{\text{imm}} = \begin{cases} 1, & \text{if } y \text{ is correct,} \\ 0, & \text{if } y \text{ is incorrect,} \\ -1, & \text{if } y \text{ contains a format error.} \end{cases} \tag{1}$$

The immediate reward reflects a turn's local quality but not its impact on the overall outcome. To capture long-term effects, we use Tree-based Monte Carlo return to obtain more stable estimates of each turn's future reward. In a multi-turn trajectory, the standard Monte Carlo return at step t, denoted by G_t, is defined as:

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t} r_k, \tag{2}$$

where $\gamma \in [0, 1]$ is the discount factor, T denotes the terminal turn, and r_k is the reward obtained at

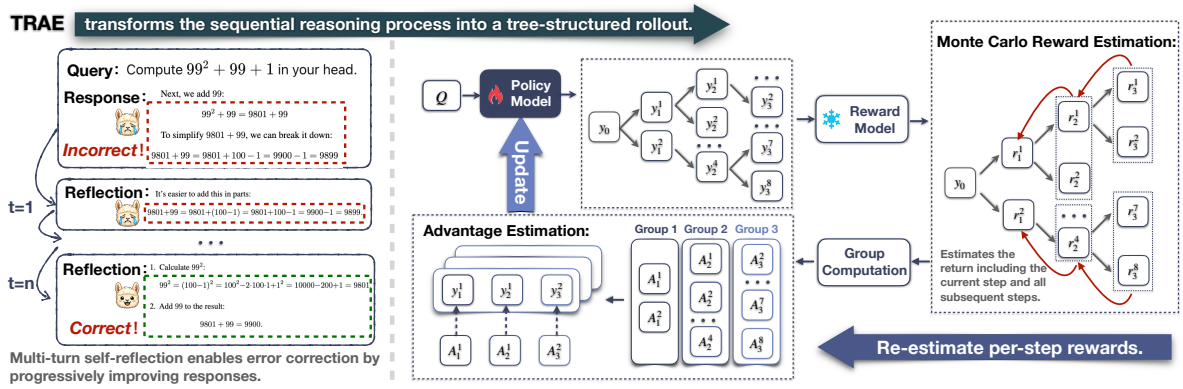

Figure 3: Illustration of TRAE framework. Our method utilizes tree-structured training with a redesigned advantage estimation. This ensures that each turn in the reasoning sequence receives an accurate estimate, addressing credit assignment challenges in multi-turn self-reflection.

turn k. In our work, the return of node v at turn t is approximated by the discounted mean of its child nodes, which we denote as $\hat{G}_v$.

$$\hat{G}_v = \sum_{k=t+1}^{T} \gamma^{k-t} \hat{r_k}, \qquad \hat{r_k} = \frac{1}{|\mathcal{C}(v,k)|} \sum_{c \in \mathcal{C}(v,k)} r_c^{\text{imm}}, \tag{3}$$

where $\mathcal{C}(v,k)$ denotes the set of child nodes of v at turn k. Finally, the overall reward assigned to node $v_{i,j}$,i.e.,the j-th node at the i-th level of the tree, is defined as:

$$r_i^j = r_v = r_v^{\text{imm}} + \hat{G}_v. \tag{4}$$

This formulation enables turn-level advantage estimation by accounting for both immediate correctness and long-term effects on subsequent reasoning.

4.2 Tree-structured Rollout

In standard GRPO, a query q is duplicated n times and rolled out independently to obtain n trajectories. In contrast, we construct a rollout tree rooted at the input question and collect trajectories as root-to-leaf paths, while maintaining the same sample efficiency. We define each node v as a tuple (h_v, t_v), where h_v denotes the conversation history and t_v is the reflection depth. The root node v_0 is initialized as $h_{v_0} = q$ and $t_{v_0} = 0$. Given a node v, we sample a response $y_v \sim \pi(\cdot \mid h_v)$ and create two child nodes u with

$$h_u = h_v \oplus y_v \oplus P, \quad t_u = t_v + 1 \tag{5}$$

where P is the reflection prompt and $\oplus$ denotes concatenation. The expansion proceeds recursively until $t_v = T$. After constructing the tree, we collect n trajectories from all root-to-leaf paths, where n equals the number of leaf nodes. For a leaf node l, the corresponding trajectory τ_l is defined as

$$\tau_l = h_l \oplus y_l, \tag{6}$$

where h_l denotes the accumulated conversation history of l and $y_l \sim \pi(\cdot \mid h_l)$ is the final response generated at the leaf.

4.3 Turn-level Credit Assignment

We first compute the Monte Carlo return for each node in the tree, following the procedure described in Section 4.1, proceeding from the bottom level to the top level. Next, we group nodes by their tree level and compute the advantage in the same way as GRPO (Shao et al., 2024). Let $\mathcal{L}(k)$ denote the set of nodes at depth (level) k, and let r_i^j denote the reward associated with node $v_{i,j}$, as defined in Section 4.1.

We normalize the reward within each level as follows:

$$\tilde{r}_i^j = \frac{r_i^j - \mu_i}{\sigma_i + \epsilon}, \qquad \mu_i = \frac{1}{|\mathcal{L}(i)|} \sum_{m \in \mathcal{L}(i)} r_m, \qquad \sigma_i = \sqrt{\frac{1}{|\mathcal{L}(i)|} \sum_{m \in \mathcal{L}(i)} (r_m - \mu_i)^2}. \tag{7}$$

where ϵ is a small constant added for numerical stability. The advantage assigned to the t-th token

in the response generated by node $v_{i,j}$ is denoted as A_i^j, and this advantage is shared by all tokens in the response.

$$\hat{A}_i^j = \tilde{r}_i^j. \tag{8}$$

Based on the above reward formulation, the optimization objective is defined as:

$$\begin{aligned}\mathcal{J}_{\text{TRAE}}(\theta) &= \mathbb{E}_{q\sim\mathcal{D}\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(q)} \\ &\quad \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\ell_{i,j,t}(\theta).\end{aligned}$$

$$\begin{aligned}\ell_{i,j,t}(\theta) = \min\Big(&r_{i,j,t}(\theta)\,\hat{A}_i^j, \\ &\text{clip}\left(r_{i,j,t}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\right)\hat{A}_i^j\Big).\end{aligned} \tag{9}$$

where

$$r_{i,j,t}(\theta) = \frac{\pi_\theta(y_{i,j,t} \mid q, y_{i,j,<t})}{\pi_{\theta_{\text{old}}}(y_{i,j,t} \mid q, y_{i,j,<t})}. \tag{10}$$

In summary, our tree-structured design enables turn-level advantage estimation during multi-turn self-reflection, ensuring that each reflection considers both the current turn's reward and its impact on future turns during optimization.

5 Experiment

We examine whether our approach enables stable reflection ability in multi-turn settings. To this end, we evaluate it across a variety of reasoning tasks and compare it against multiple baselines and existing methods (Section 5.2). Furthermore, we compared the effectiveness of self-reflection against other test-time scaling methods(Section 5.3), and we also analyzed the factors underlying the sustained performance gains observed with our self-reflection approach(Section 5.4). Finally, we verify that our method effectively mitigates behavior collapse by comparing accuracy and edit distance (Section 5.5).

5.1 Experiment Setup

Baselines We evaluate our method against three categories of approaches: (1) General methods, including SFT, GRPO, and DAPO (Yu et al., 2025); and (2) tree-based methods, such as TreePO (Li et al., 2025), and TreeRPO (Yang et al., 2025), which use tree structure during RL; and (3) other methods, such as TDRM (Zhang et al., 2025a), PAG (Jiang et al., 2025), Absolute-Zero-8B (Zhao et al., 2025), and R-Zero-8B (Huang et al., 2025b), which denote other approaches designed to improve reasoning ability.

Datasets Our training dataset consists of query–response pairs prepared in advance, containing 10k samples in total. The data is divided into two parts: the incorrect raw responses are sourced from (self-correxp2, 2025), while the correct raw responses are generated by Qwen3-8B-Base. The proportion of incorrect to correct responses is 1:1.

Benchmark We evaluate our method on widely used mathematical reasoning benchmarks, including MATH-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AIME 24, AIME 25, Olympiad (He et al., 2024), and AMC. These datasets span grade-school to Olympiad-level math across domains such as algebra, geometry, number theory, combinatorics, and probability, enabling a comprehensive evaluation of multi-turn reasoning and reflection.

Implementation Details We employ RL training library VERL (Sheng et al., 2024) to train our agent. The learning rate is set to 1×10^{-6}, with a batch size of 256 and a mini-batch size of 64. We restrict the length of a single response to fewer than 4096 tokens, with temperature 0.6 and top-p 0.95. ϵ clip-high is set to 0.4, and ϵ clip-low is 0.2. We construct a full binary tree of depth 3, which yields eight trajectories per iteration. The discount factor γ is fixed at 0.5. Training is conducted on eight NVIDIA A800 GPUs (80 GB each). For Qwen3-8B-Base, the training lasts for 3 epochs (117 steps in total), consuming approximately 40 hours.

5.2 Main Results

Table 1 presents the main results of our experiments on six math benchmarks. We analyze the performance from the following:

TRAE achieves substantial improvements across benchmarks. As shown in Table 1, TRAE achieves the highest average accuracy over six benchmarks, outperforming Qwen3-8B-Base by **+20.1** points and consistently surpassing SFT, GRPO, and DAPO baselines. TRAE attains the best result on AMC23 and ranks second on MATH-500, AIME24, GSM8K, and Olympiad. In contrast, Tree-based and Other methods tend to excel on individual benchmarks but generalize less consistently. Overall, these results demonstrate that TRAE delivers more robust and broadly applicable gains in mathematical reasoning.

TRAE achieves greater gains in self-reflection than General methods. As can be clearly ob-

Model	MATH-500	AIME24	AIME25	GSM8K	Olympiad	AMC23	Average
Tree-based Methods							
TreePO	**85.3**	27.8	-	-	**49.1**	55.5	-
TreeRPO	75.5	26.7	-	-	35.4	-	-
Other Methods							
TDRM	74.6	**36.7**	-	-	37.3	<u>62.5</u>	-
PAG	82.3	18.4	15.1	-	-	-	-
R-Zero-8B	82.0	16.4	**19.2**	**94.1**	48.9	61.7	<u>53.7</u>
Absolute-Zero-8B	76.6	18.2	<u>18.4</u>	92.0	47.8	<u>62.5</u>	52.5
General Methods							
Qwen3-8B-Base	58.0	10.0	10.0	72.3	29.3	37.5	36.1
w/ reflection (r=3)	41.8$_{\downarrow 16.2}$	6.67$_{\downarrow 3.3}$	6.67$_{\downarrow 3.3}$	49.8$_{\downarrow 22.5}$	20.7$_{\downarrow 8.6}$	20.0$_{\downarrow 17.5}$	24.2$_{\downarrow 11.9}$
SFT	71.8	20.0	10.0	84.3	38.7	52.5	46.2
w/ reflection (r=3)	78.4$_{\uparrow 6.6}$	16.7$_{\downarrow 3.3}$	13.3$_{\uparrow 3.3}$	91.1$_{\uparrow 6.8}$	41.3$_{\uparrow 2.6}$	52.5$_{\uparrow 0.0}$	48.8$_{\uparrow 2.6}$
GRPO	65.2	13.3	10.0	77.1	31.5	60.0	42.8
w/ reflection (r=3)	69.8$_{\uparrow 4.6}$	10.0$_{\downarrow 3.3}$	13.3$_{\uparrow 3.3}$	79.8$_{\uparrow 2.7}$	36.8$_{\uparrow 5.3}$	47.5$_{\downarrow 12.5}$	42.8$_{\uparrow 0.0}$
DAPO	62.0	20.0	6.67	75.5	31.7	47.5	40.5
w/ reflection (r=3)	69.6$_{\uparrow 7.6}$	23.3$_{\uparrow 3.3}$	10.0$_{\uparrow 4.0}$	83.6$_{\uparrow 8.1}$	35.8$_{\uparrow 4.1}$	50.0$_{\uparrow 2.5}$	45.5$_{\uparrow 5.0}$
Ours							
TRAE	70.2	23.3	10.0	82.3	37.0	57.5	46.7
w/ reflection (r=3)	<u>82.4</u>$_{\uparrow 12.2}$	<u>33.3</u>$_{\uparrow 10.0}$	16.6$_{\uparrow 6.6}$	<u>93.0</u>$_{\uparrow 10.7}$	<u>44.9</u>$_{\uparrow 7.9}$	**67.5**$_{\uparrow 10.0}$	**56.2**$_{\uparrow 9.5}$

Table 1: Accuracy on six mathematics benchmarks. **Bold** and <u>underline</u> denote the best and second-best results in each column, respectively. For Baselines and Ours, the colored deltas indicate the change introduced by reflection (r=3) relative to the corresponding non-reflection setting.

Method	MATH-500	AIME24	AIME25	GSM8K	Olympiad	AMC23	AVG
TRAE	70.2	<u>23.3</u>	10.0	82.3	37.0	57.5	46.7
w/ Majority@4	76.0	20.0	10.0	89.5	42.8	<u>67.5</u>	51.0
w/ Best-of-4-Skywork-Reward-V2	81.4	20.0	<u>13.3</u>	90.9	45.5	60.0	51.8
w/ Best-of-4-AceMath-7B-RM	<u>83.0</u>	20.0	10.0	92.8	<u>47.3</u>	65.0	53.0
w/ Pass@4	**88.2**	<u>23.3</u>	<u>13.3</u>	**95.2**	**55.6**	**72.5**	**58.0**
w/ Reflection (r=3)	82.4	**33.3**	**16.6**	<u>93.0</u>	44.9	<u>67.5</u>	<u>56.2</u>

Table 2: The result from different test-time scaling method, with **bold numbers** indiacting the best performance and <u>underline numbers</u> representing the second-best performance.

served in Table 1, the Qwen3-8B-Base model fails to perform effective self-reflection. In particular, its accuracy decreases across all six benchmarks, with the average accuracy dropping by 11.9%. In contrast, TRAE already achieves **46.7%** accuracy in the first round, surpassing almost all methods in General Methods. This improvement is further amplified after three rounds of reflection, yielding substantial gains on all six benchmarks and increasing the average accuracy by **9.5%** which is the largest gain among all reflection-based methods compared with the first round.

5.3 Comparison with Existing Test-Time Scaling Methods

Table 2 presents a systematic comparison between self-reflection and other test-time scaling methods on six benchmarks. We evaluate four test-time scaling strategies: majority voting, best-of-4, self-reflection, and pass@4. For the best-of-4 setting, we further assess performance under two reward models, Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025a) and AceMath-7B-RM (Liu et al., 2025b). We report and discuss the results from the following perspective:

TRAE's self-reflection-based test-time scaling yields substantially larger gains than other scal-

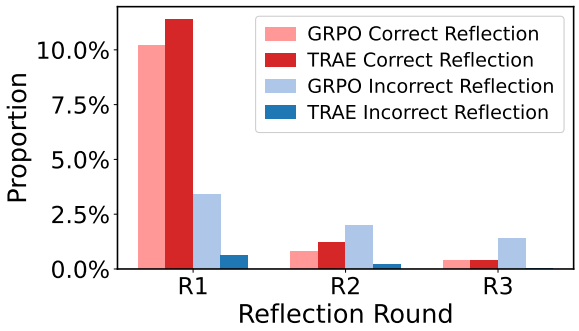

Figure 4: Reflection performance comparison between GRPO and TRAE on MATH-500, showing the rates of correct and incorrect reflections over three rounds.

	w/o MCR	w MCR	Δ
Epoch 1			
w/o reflection	66.0	66.0	+0.0
w/ reflection (r=3)	78.2	76.0	-2.2
Epoch 2			
w/o reflection	67.0	**69.4**	+2.4
w/ reflection (r=3)	76.8	**80.2**	+3.4
Epoch 3			
w/o reflection	71.8	70.2	-1.6
w/ reflection (r=3)	79.0	**82.4**	+3.4

Table 3: Ablation of MCR (Monte-Carlo Return) across training epochs. w/o reflection denotes accuracy under direct generation, while w/ reflection (r=3) reports accuracy after three reflection steps.

ing strategies. Table 2 shows that majority voting and best-of-4 provide limited gains. Self-reflection delivers a substantially larger improvement, reaching **56.2%** and trailing only pass@4. Notably, with self-reflection, TRAE outperforms Pass@4 on both AIME24 and AIME25, and ranks second only to Pass@4 on GSM8K and AMC23. These results indicate that TRAE exhibits strong reflective capability, consistently surpassing common test-time scaling baselines and approaching the upper bound represented by pass@4.

5.4 Analysis of Reflection

In Figure 4, we analyze Correct Reflection (an incorrect response corrected through reflection) and Incorrect Reflection (a correct response degraded by reflection) for GRPO and TRAE over three reflection rounds. Notably, TRAE consistently achieves more Correct Reflection than GRPO, highlighting its superior ability for error correction and stable refinement across multiple reflection rounds. In addtion, TRAE maintains an Incorrect Reflection rate below 1% throughout and reduces it to 0% by the third round, indicating strong preservation of previously correct outputs.

5.5 Ablation Studies

Effect of Monte Carlo Return Horizon To analyze the effect of Monte Carlo return estimation, we vary the discount factor in Equation 3, with $\gamma = 0$ yielding a myopic one-step return and $\gamma = 0.5$ corresponding to a multi-step discounted Monte Carlo return. All experiments use the training data in Section 5.1, run for three epochs, and keep other hyperparameters fixed. As shown in Table 3, the myopic return achieves comparable or slightly higher accuracy in the first epoch, but fails to improve consistently thereafter. In contrast, multi-step Monte Carlo return estimation yields steady gains across epochs, especially under reflection, with improvements of up to $+3.4$ accuracy points. These results indicate that extending the Monte Carlo return horizon provides richer credit assignment signals and supports sustained training improvements.

From Entropy Stabilization to Effective Reflection Figure 5 compares test-time accuracy and edit distance across GRPO, GRPO with clip-higher, and TRAE. As shown in Figure 5, increasing the clip-higher ratio (Yu et al., 2025) from 0.2 to 0.4 effectively mitigates entropy collapse during training and slightly increases edit distance, however, it brings no clear accuracy improvement, indicating that entropy stabilization alone is insufficient for effective reflection. In contrast, when combined with clip-higher, TRAE enables substantially larger revisions and achieves a steady accuracy gain. This suggests that our method translates stabilized policy entropy into more effective reflective updates rather than merely encouraging superficial changes.

6 Conclusion

We propose TRAE, a GRPO variant that enables robust multi-turn self-reflection through tree-structured reasoning and turn-level reward assignment. Our empirical analysis reveals the Echo Trap, where conventional multi-turn training fails to assign accurate credit to corrective actions, leading to training collapse. To address this issue, TRAE explicitly models turn-level rewards that capture both immediate gains and downstream effects. As a result, TRAE mitigates behavior collapse and achieves consistent performance improvements across reflection turns. Overall, our approach enhances self-reflection performance and

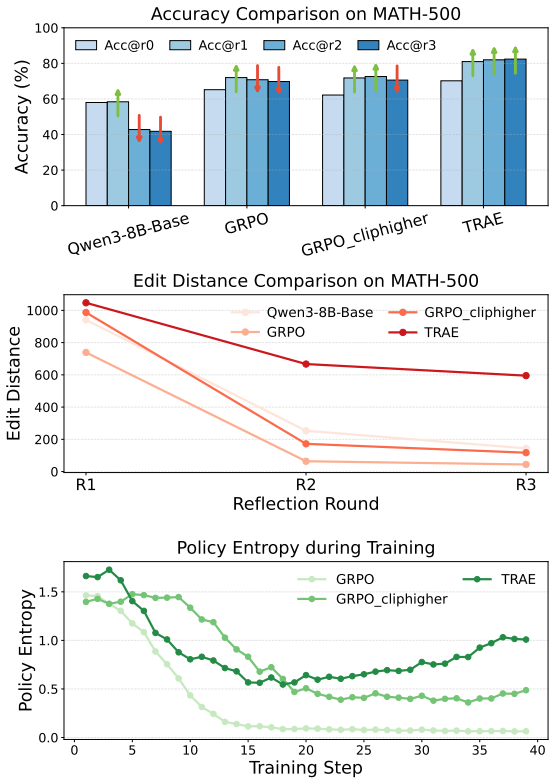

Figure 5: Results of multi-round reflective training methods on MATH-500.

provides a scalable path toward more effective reasoning in complex, multi-step scenarios.

7 Limitation

Due to computational constraints, our experiments are mainly conducted on Qwen3-8B-Base, which limits comprehensive comparisons across different model sizes and architectures. Consequently, the generality of our method to other model scales and series remains to be further validated. In addition, our evaluation focuses on mathematical problem-solving tasks, and it is unclear how well the method extends to other multi-turn agent settings such as search or tool-based reasoning. We also train the model using a relatively small dataset of 10k examples, leaving the effect of larger-scale training unexplored. Finally, the influence of the ratio between correct and incorrect initial responses in the training data is not systematically studied.

8 Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments.This work was supported by the National Science Foundation of China (NSFC No. 62576232)Key Laboratory of Data Intelligence and Advanced Computing, Soochow University. Yijun Zhang, Ji Qi, Wenpeng Zhu and Juntao Li are corresponding authors.

References

Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mozolevskyi, Muayad Ali, and Waseem AlShikh. 2025. Reflect, retry, reward: Self-improving llms via reinforcement learning. *arXiv preprint arXiv:2505.24726*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

AGENTS VIA TURN-LEVEL REWARD DESIGN. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level reward design.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. 2025b. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

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 arXiv:2103.03874*.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. 2025. Treerl: Llm reinforcement learning with on-policy tree search. *arXiv preprint arXiv:2506.11902*.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T Ash, and Akshay Krishnamurthy. 2024. Self-improvement in language models: The sharpening mechanism. *arXiv preprint arXiv:2412.01951*.

Bingning Huang, Tu Nguyen, and Matthieu Zimmer. 2025a. Tree-opo: Off-policy monte carlo tree-guided advantage optimization for multistep reasoning. *arXiv preprint arXiv:2509.09284*.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. 2025b. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2025. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. 2025. Pag: Multi-turn reinforced llm self-correction with policy as generative verifier. *arXiv preprint arXiv:2506.10406*.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. Training language models to self-correct via reinforcement learning, 2024. *URL https://arxiv.org/abs/2409.12917*, 2(3):4.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, and 1 others. 2025. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and 1 others. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025b. Acemath: Advancing frontier math reasoning with post-training and reward modeling. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3993–4015.

Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. 2025. S^2 r: Teaching llms to self-verify and self-correct via reinforcement learning. *arXiv preprint arXiv:2502.12853*.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. 2023. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *Advances in Neural Information Processing Systems*, 37:55249–55285.

selfcorrexp2. 2025. Llama-3.1 math chat dataset with initial incorrect responses. https://huggingface.co/datasets/selfcorrexp2/10k_llama31_first_wrong_math_chat_format. Accessed: 2026-01-3.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. 2024. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*.

Hieu Tran, Zonghai Yao, and Hong Yu. 2025. Exploiting tree structure for credit assignment in rl training of llms. *arXiv preprint arXiv:2509.18314*.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Yuanzhu Peter Chen, and Tony Mak. 2024. Llms cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, and 1 others. 2025. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. 2025. Treerpo: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*.

Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023. Selfee: Iterative self-revising llm empowered by self-feedback generation. *Blog post*, 1.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Dan Zhang, Min Cai, Jonathan Light, Ziniu Hu, Yisong Yue, and Jie Tang. 2025a. Tdrm: Smooth reward models with temporal difference for llm rl and inference. *arXiv preprint arXiv:2509.15110*.

Haoke Zhang, Xiaobo Liang, Cunxiang Wang, Juntao Li, and Min Zhang. 2025b. Unlocking recursive thinking of llms: Alignment via refinement. *arXiv preprint arXiv:2506.06009*.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*.

A Details of Multi-turn Reflection

To enable a more direct comparison of reflective capability across methods, we evaluate the outcomes of three consecutive reflection rounds for each approach. As shown in Table 4, SFT, GRPO, and DAPO generally exhibit an improvement followed by a decline, whereas our method maintains consistent gains across reflection rounds.

B Generalizability of TRAE

To demonstrate the generalizability of TRAE, we evaluate it from two perspectives: task generalization and model architecture generalization.

B.1 Task Generalization

For task generalization, we conduct an out-of-domain evaluation on MMLU-PRO, a widely used multi-domain benchmark beyond mathematical reasoning. As shown in Table 5, under the same setting and with up to three reflection rounds, TRAE exhibits a consistent monotonic improvement as the number of reflection rounds increases and achieves the best performance at every reflection depth. TRAE also consistently outperforms standard GRPO, suggesting that the learned self-reflection ability is not limited to mathematical reasoning tasks.

B.2 Model Architecture Generalization

To further validate the effectiveness of our method across models with different capabilities, we additionally evaluate TRAE on Llama 3.1-8B-Instruct and observe the same qualitative behavior on MATH-500 and GSM8K. As shown in Table 6, TRAE remains stable and consistently benefits from reflection, whereas the base model degrades as the number of reflection rounds increases.

C Comparison of Hyperparameters

To assess the effectiveness of the discount factor γ in the reward estimation stage while controlling for the effect of random seeds, we evaluate TARE under different choices of γ and different seed settings.

C.1 Effect of Monte Carlo Return Horizon

The results in Table 7 show that ignoring MCR ($\gamma = 0$) enables the model to converge quickly, but to a suboptimal solution, with limited potential for continued improvement during training. In contrast, incorporating MCR ($\gamma = 0.5$ and 1) yields more consistent performance gains as training progresses. Among these settings, $\gamma = 0.5$ achieves the best overall performance, suggesting that a balanced trade-off between immediate rewards and long-term returns (MCR) is particularly effective.

C.2 Effect of Seed

As shown in Table 8, the performance remains stable across different seed settings, indicating that our results are robust to variations in seed choice.

D Prompt Engineering

In this section, we present the prompt templates used during training and inference. Our framework employs different prompts for response generation and self-reflection, each designed to support a specific stage of the reasoning process.

During training, we use three types of prompts. Figure 6 shows the prompt used for initial response generation. Figure 7 presents the prompt for single-turn self-reflection, where the model revises its previous answer based on the original question and the initial solution. Figure 8 illustrates the prompt for multi-turn self-reflection, where the model has access to the full reflection history from previous turns. This design allows the model to iteratively refine its reasoning and produce more accurate solutions over multiple rounds.

Model	MATH-500	AIME24	AIME25	GSM8K	Olympiad	AMC23	AVG
Baselines							
SFT	71.8	20.0	10.0	84.3	38.7	52.5	46.2
w/ reflection (r=1)	78.0	20.0	13.3	91.7	41.5	50.0	49.0
w/ reflection (r=2)	78.2	16.7	13.3	91.4	41.5	50.0	48.5
w/ reflection (r=3)	78.4	16.7	13.3	91.1	41.3	52.5	48.8
GRPO	65.2	13.3	10.0	77.1	31.5	60.0	42.8
w/ reflection (r=1)	72.0	13.3	13.3	84.3	37.1	55.0	45.8
w/ reflection (r=2)	70.8	13.3	13.3	83.2	37.7	52.5	45.1
w/ reflection (r=3)	69.8	10.0	13.3	79.8	36.8	47.5	42.8
DAPO	62.0	20.0	6.67	75.5	31.7	47.5	40.6
w/ reflection (r=1)	70.8	20.0	10.0	84.1	35.7	50.0	45.1
w/ reflection (r=2)	70.2	23.3	10.0	84.4	36.0	52.5	46.0
w/ reflection (r=3)	69.6	23.3	10.0	83.6	35.8	50.0	45.3
Ours							
TRAE	70.2	23.3	10.0	82.3	37.0	57.5	46.7
w/ reflection (r=1)	81.0	30.0	**16.6**	92.5	43.7	65.0	54.8
w/ reflection (r=2)	82.0	**33.3**	**16.6**	**93.2**	44.6	**67.5**	**56.2**
w/ reflection (r=3)	**82.4**	**33.3**	**16.6**	93.0	**44.9**	**67.5**	**56.2**

Table 4: Reflection scores for each round, with **bold numbers** indiacting the best performance and underline numbers representing the second-best performance.

Prompt

{question} The finally answer should be enclosed in boxed {{}}.

Figure 6: Prompt for Response Generation.

MMLU_PRO	Qwen3-8B-Base	GRPO	TRAE
direct	**32.5**	38.4	46.2
w/ reflection (r=1)	27.5	41.6	50.4
w/ reflection (r=2)	27.4	**43.3**	52.8
w/ reflection (r=3)	26.8	43.0	**53.8**

Table 5: Reflection scores for each round, with **bold numbers** indicating the best performance and underlined numbers representing the second-best performance.

	MATH-500		GSM8K	
	Llama3.1-8B-Instruct	TRAE	Llama3.1-8B-Instruct	TRAE
direct	47.0	50.0	83.2	83.5
w/ reflection (r=1)	45.0	50.8	76.0	86.1
w/ reflection (r=2)	43.6	51.4	75.2	87.1
w/ reflection (r=3)	43.2	51.6	73.6	87.3

Table 6: Comparison of direct generation and multi-round reflection on MATH-500 and GSM8K. TRAE consistently improves performance with reflection, while the vanilla Llama3.1-8B-Instruct model degrades as reflection rounds increase.

	$\gamma = 0$	$\gamma = 1$	$\gamma = 0.5$
Epoch 1			
direct	66.0	65.4	66.0
w/ reflection (r=3)	78.2	75.4	76.0
Epoch 2			
direct	67.0	67.8	69.4
w/ reflection (r=3)	76.8	77.6	80.2
Epoch 3			
direct	71.8	68.2	70.2
w/ reflection (r=3)	79.0	78.2	82.4

Table 7: Results on MATH-500 under different γ values across training epochs. w/ reflection (r=3) reports accuracy after three reflection steps.

MATH-500	**random seed(ours)**	**fixed 0**	**fixed 42**	**fixed 100**
direct	70.2	70.8	68.8	72.4
w/ reflection (r=1)	81.0	81.4	80.2	80.6
w/ reflection (r=2)	82.0	82.4	81.6	81.4
w/ reflection (r=3)	82.4	82.6	81.6	82.0

Table 8: Results of different random seed settings on TARE. "Random seed" means that the seed is sampled randomly for each training run, whereas "fixed" means that a predefined seed is used consistently throughout training.

Prompt

You are given a math problem and a initial solution to the problem. Your task is to generation a critique to the solution which should contain the analysis to the solution, the judge to the solution and some instruction about how to correct the solution if it is wrong or make it better if it is correct and then generate a new solution refined from the initial solution that given to you, and you should refer from the critique because you can get some advice from it. The new solution should solve the problem step by step. The final answer must be enclosed in boxed {{}}. You must output in the following format: <critique>your critique to the solution</critique> <response>the new solution refined from the initial solution</response>

Figure 7: Prompt for Self-Reflection.

Prompt

The solution you need to reflect is in the previous step and enclosed in <response></response>. Your task is to generation a critique to the solution, which should contain the analysis to the solution, the judge to the solution and some instruction about how to correct the solution if it is wrong or make it better if it is correct and then generate a new solution refined from the previous solution, and referred from the critique. The new solution should solve the problem step by step. The final answer must be enclosed in boxed {{}}. You must output in the following format: <critique>your critique to the solution</critique><response>the new solution refined from the initial solution</response>

Figure 8: Self-Reflection Prompt Incorporating Previous Turn Context.