REVISITING MATHEMATICAL REASONING VERIFICATION WITH DEEP-RESEARCH-GROUNDED TRAJECTORY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown strong mathematical reasoning capabilities, yet reliably verifying existing reasoning traces remains challenging. A reliable verifier should detect genuine errors, localize the authoritative first error step, and avoid unsupported judgments on valid reasoning. We propose Deep-Research-Grounded Trajectory Distillation (DRGTD), a framework that combines structured verification supervision with conditional localization optimization. We first construct a multi-agent teacher–reviewer trajectory-construction pipeline centered on a Deep-Research Teacher. The Teacher performs source-grounded semantic analysis and selective external mathematical checks to produce structured verification trajectories. Fail-closed trajectory validation then retains supported information and compiles it into Decision, First-Error, Process, and Unified-Deployment supervision. We next apply two-stage trajectory distillation. Phase 1 performs multi-view capability learning, while Phase 2 aligns the learned behavior with the natural deployment interface. Finally, we freeze the SFT reasoning-trace-level CLEAN/ERROR decision gate and apply frozen-gate RL only to conditional first-error localization using pairwise hard negatives and frozen-reference KL regularization. On the primary Qwen2.5-Math-7B backbone, Strict Verification Accuracy improves from 29.33% for Base to 35.33% after SFT and 37.00% after RL. Base→SFT also improves Strict Verification Accuracy and Exact FE across 10 open-weight backbones. On three representative backbones, Pairwise PG + KL achieves higher Exact FE, FE—Detected, and Strict Verification Accuracy than PPO and Bandit-OPD. Teacher-side confirmatory evaluation shows a directional but not statistically significant improvement over native verification. Overall, our results support a controlled path from Deep-Research teacher verification and validated trajectory distillation to frozen-gate conditional localization optimization for compact mathematical reasoning verifiers.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong mathematical reasoning capabilities (Wei et al., 2022; Cobbe et al., 2021), but generating a seemingly reasonable solution and reliably verifying an existing reasoning trace remain distinct challenges. A reliable verifier not only needs to determine whether the reasoning is valid, but also needs to identify genuine errors and the first error step, rather than a downstream erroneous result. As the mathematical reasoning capabilities continue to improve, process-level supervision and fine-grained verification have thus become increasingly important (Uesato et al., 2022; Lightman et al., 2024).

However, the reliability of the verifier cannot be solely measured by its ability to retain correct reasoning. In our experiments, the Base model rarely rejected correct reasoning but allowed a large number of erroneous reasoning traces, showing a clear tendency toward conservative acceptance. After trajectory distillation, the model's ability to identify errors significantly improved, while the retention of correct reasoning only slightly changed.

Greater error sensitivity is still not sufficient to ensure reliable verification. Even strong models may miss subtle errors, misjudge correct steps, or mistake downstream errors for the first error step (Tyen

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2024; Jacovi et al., 2024; Jiang et al., 2024; Zheng et al., 2025). We call such unsupported or erroneous attributions *verification hallucinations*, and we build a Deep-Research Teacher that uses source-step analysis and selective external mathematical checks for verification judgments; external tools only provide evidence and do not directly determine the first error step (Yao et al., 2023; Gou et al., 2024a;b).

Even when teachers are constrained by external evidence, their freely generated reasoning should not be directly regarded as trustworthy supervision (Hsieh et al., 2023; Turpin et al., 2023; Wang et al., 2023; Paul et al., 2024). We represent the observable verification process as a structured verification trajectory, where fail-closed trajectory validation removes unverifiable information, and then it is compiled into complementary supervision. We refer to this transfer of validated information from Deep-Research Teacher trajectories into student supervision as Deep-Research-grounded trajectory distillation. Experiments show that this distillation simultaneously reduces missed errors and improves end-to-end first-error localization.

First-error localization is affected by upstream error detection behavior: if different verifiers identify different sets of errors, their conditional localization results are difficult to compare directly. To address this, we freeze the reasoning-trace-level CLEAN/ERROR decision after supervised training and optimize first-error localization only for reasoning traces already classified as ERROR, ensuring that reinforcement learning gains cannot come from changing sample routing. In our experiments, localization further improves while the upstream CLEAN/ERROR decision remains unchanged. On PRM300, this path increases the strict end-to-end verification accuracy of the 7B verifier from 29.33% for the Base model to 35.33% after SFT under a modest teacher-trajectory budget, and further to 37.00% with RL while keeping the upstream CLEAN/ERROR decision frozen.

Our main contributions are:

- **Deep-Research teacher verification.** We propose a Deep-Research Teacher integrating source-step semantic analysis, selective external mathematical verification, and evidence provenance; external tools merely provide evidence and do not directly determine the first error step.
- **Deep-Research-grounded trajectory distillation.** We represent the teacher's observable verification process as a structured verification trajectory, remove unverifiable information through fail-closed trajectory validation, and compile the retained information into complementary supervision for error identification, first-error localization, and deployment behavior.
- **Frozen-gate localization reinforcement learning.** We freeze the reasoning-trace-level CLEAN/ERROR decision after supervised training and optimize first-error localization only for reasoning traces already classified as ERROR, decoupling localization improvements from changes in upstream detection behavior.

2 Related Work

2.1 Mathematical Reasoning Verification

Existing work has gradually treated verification as a distinct problem within mathematical reasoning. Early methods mainly scored candidate answers or reasoning outputs (Cobbe et al., 2021). Subsequent studies have shown that process-level supervision, compared with outcome-only supervision, can provide more fine-grained verification signals (Uesato et al., 2022; Lightman et al., 2024), and have further developed automatic process supervision and progress-sensitive process reward models (Wang et al., 2024; Setlur et al., 2025; Zhang et al., 2025). At the same time, studies have found that even strong language models have difficulty reliably identifying reasoning errors and their first error steps (Tyen et al., 2024; Jacovi et al., 2024; Jiang et al., 2024; Zheng et al., 2025). Recent work further studies temporal consistency, richer verifier feedback, and cross-domain evaluation (Guo et al., 2025; Gao et al., 2025; Li et al., 2026). Our work further focuses on the relationship between correct-reasoning preservation, error detection, and first-error localization, and treats these as a unified end-to-end verification problem.

2.2 TOOL-AUGMENTED AND AGENTIC REASONING

Tool-augmented language models expand their reasoning capabilities through external computation, execution, or feedback. For example, ReAct, Toolformer, and PAL have respectively studied the interaction between reasoning and action, tool-use learning, and program-assisted reasoning (Yao et al., 2023; Schick et al., 2023; Gao et al., 2023). In mathematics, ToRA, CRITIC, and recent tool-augmented verification methods have further utilized external tools for problem solving, critique, self-correction, and verification (Gou et al., 2024a;b; Kang et al., 2026; Chang et al., 2026; Liu et al., 2025). Unlike these methods, we strictly limit tools to evidence sources bound to source steps and do not allow tool results to directly replace semantic judgments or determine the first error step.

2.3 REASONING DISTILLATION AND STRUCTURED SUPERVISION

Reasoning distillation utilizes teacher-generated reasoning signals to enhance smaller models and has shown effectiveness with limited training data and structured reasoning supervision (Hsieh et al., 2023; Gu et al., 2024; Dai et al., 2024). However, freely generated chain-of-thought does not necessarily faithfully reflect the actual reasoning process of the model, and it may contain unfaithful explanations or explanations weakly related to the final decision (Turpin et al., 2023; Wang et al., 2023; Paul et al., 2024; Wadhwa et al., 2024). Therefore, we do not directly distill unrestricted teacher chain-of-thought, but only transfer structured verification information that has passed fail-closed trajectory validation.

2.4 REINFORCEMENT LEARNING FOR VERIFIER OPTIMIZATION

Reinforcement learning and preference optimization have been widely applied in mathematical reasoning and verifier training, including DPO, V-STaR, and optimization methods based on verifiable or process-level rewards (Rafailov et al., 2023; Hosseini et al., 2024; Shao et al., 2024; Wang et al., 2024; Setlur et al., 2025). Our focus is not on proposing new reinforcement learning objectives, but rather on restricting the behaviors that reinforcement learning can modify: freezing the reasoning-trace-level CLEAN/ERROR decisions after supervised training, and only optimizing the first-error localization in reasoning traces already classified as ERROR, thereby decoupling localization improvements from changes in upstream error detection.

3 PROBLEM FORMULATION

3.1 VERIFICATION AND FIRST-ERROR LOCALIZATION

Let x denote a mathematical problem and $r = (r_1, \ldots, r_T)$ a corresponding step-by-step reasoning trace. We study verification of the provided reasoning rather than independent solution generation. A reasoning trace is labeled CLEAN if it contains no substantive reasoning error, and ERROR if it contains at least one reasoning error. The verifier first makes a reasoning-trace-level CLEAN/ERROR decision:

$$\hat{y} = G(x, r), \qquad \hat{y} \in \{\textsc{Clean}, \textsc{Error}\}. \tag{1}$$

For an ERROR reasoning trace, verification additionally requires locating the first step at which an error is introduced. We define the *authoritative first error step* as

$$t^* = \min\{t \mid r_t \text{ introduces a reasoning error}\}. \tag{2}$$

Localization is invoked only when $\hat{y} = \textsc{Error}$:

$$\hat{t} = L(x, r \mid \hat{y} = \textsc{Error}). \tag{3}$$

A downstream error may be a consequence of an earlier failure; therefore, identifying such a later step does not constitute exact first-error localization.

3.2 VERIFICATION METRICS

We evaluate verification along three components. CLEAN Accuracy measures preservation of valid reasoning, Error Detection measures detection of erroneous reasoning traces, and Exact FE measures end-to-end first-error localization over all ERROR examples. $\mathrm{FE} \mid \mathrm{Detected}$ instead measures

conditional localization only among examples already detected as erroneous, while Balanced Decision combines CLEAN Accuracy and Error Detection. Complete definitions of these component metrics are provided in Appendix A.6.

Our primary aggregate metric is *Strict Verification Accuracy*. A CLEAN example is successful only if it is correctly accepted, whereas an ERROR example is successful only if the error is detected and its authoritative first error step is exactly localized. We define

$$\mathrm{StrictVerAcc} = \frac{N_{\mathrm{CorrectClean}} + N_{\mathrm{ExactFE}}}{N}, \tag{4}$$

where $N_{\mathrm{CorrectClean}}$ is the number of correctly accepted CLEAN examples, N_{ExactFE} is the number of ERROR examples whose authoritative first error step is exactly localized, and N is the total number of examples.

4 METHOD

Figure 1 summarizes our four-stage framework: Deep-Research teacher verification, fail-closed trajectory validation, multi-view trajectory distillation, and frozen-gate RL for conditional first-error localization.

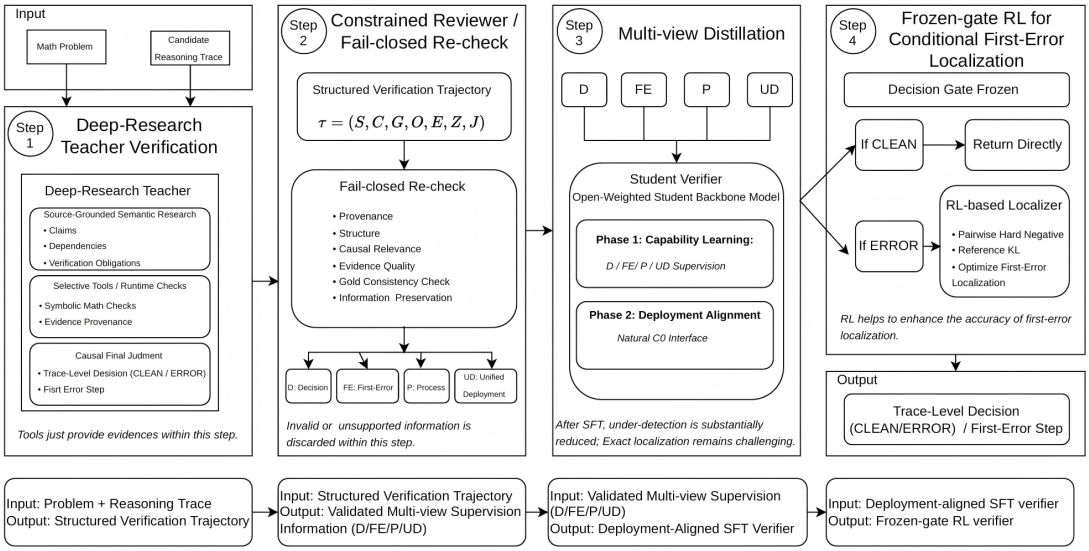

Figure 1: Method overview. Deep-Research teacher verification and fail-closed trajectory validation produce multi-view supervision, followed by frozen-gate RL for conditional first-error localization.

4.1 DEEP-RESEARCH TEACHER VERIFICATION

The Deep-Research Teacher employs two model invocations separated by a deterministic runtime. The first invocation reads the original problem and the complete reasoning trace, extracting semantic claims, causal dependencies, and verification obligations bound to the source steps, but does not predict or rank candidate first error steps. The model is responsible for semantic interpretation, while the runtime is responsible for preserving source identity and controlling the compilation and execution of subsequent checks.

During runtime, based on the structured research state and the immutable original input, it selectively constructs and executes mathematical checks bound to the source propositions, and records evidence provenance and its correspondence with the original propositions. Checks that cannot be faithfully mapped to the source propositions remain unresolved instead of being replaced by more easily verifiable proxy propositions; external tools only provide evidence and do not directly generate first-error-step labels.

The second model invocation receives the original reasoning trace, the structured research state, and the accumulated evidence, and outputs the reasoning-trace-level CLEAN/ERROR decision and

the first error step. Downstream errors cannot be automatically regarded as the first error step; the Teacher must trace the error source along the causal dependencies and check whether earlier substantive errors exist in the preceding reasoning prefix. The complete protocol, execution budget, and evidence-recording details are provided in Appendix B.2.

4.2 Structured Trajectory Validation and Multi-View Distillation

The final judgment made by the teacher will omit much of the verification process that led to it, while freely generated reasoning may contain unsupported intermediate information. Therefore, we represent the observable verification process as a structured verification trajectory:

$$\tau = (\mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{O}, \mathcal{E}, \mathcal{Z}, \mathcal{J}),$$

where the components $\mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{O}, \mathcal{E}, \mathcal{Z}, \mathcal{J}$ respectively represent source coverage, semantic claims, causal dependencies, verification obligations, evidence, claim states, and the final causal judgment. The teacher trajectory is not directly regarded as trusted supervision; fail-closed trajectory validation only verifies existing information, and content that cannot be validated is excluded rather than regenerated. The reviewer is not allowed to re-solve the problem or generate a new FE label. The validation is carried out in a view-specific manner, so failure of process information does not necessarily invalidate otherwise usable decision or localization supervision. The complete trajectory schema and validation gates are provided in Appendix B.3 and Appendix B.1. The information that has passed validation is then compiled into four complementary views:

$$\mathcal{V} = \{D, F, P, U\},$$

where D is used for the reasoning-trace-level CLEAN/ERROR decision, F is used for first-error localization on ERROR reasoning traces, P retains the validated causal and evidence relationships, and U represents the unified behavior required at deployment.

On the basis of validated supervision, we adopt a two-stage distillation approach. In Phase 1, we jointly learn the decision, localization, process-reasoning, and unified-deployment capabilities of $D/F/P/U$ according to a controlled view-exposure policy; in Phase 2, without introducing new teacher trajectories or mathematical supervision, we align the unified behavior with the natural deployment interface. This design separates capability acquisition from deployment alignment. The specific optimization objectives and exposure settings are provided in Appendix A.3.

4.3 Frozen-Gate RL for Conditional First-Error Localization

SFT simultaneously learns the reasoning-trace-level CLEAN/ERROR decision and first-error localization. The subsequent goal of RL is to further optimize localization without changing the learned CLEAN/ERROR decision. Therefore, we freeze the SFT decision gate and restrict RL to conditional localization. Let G_ϕ denote the frozen SFT decision gate:

$$\hat{y} = G_\phi(x, r), \qquad \hat{y} \in \{\textsc{Clean}, \textsc{Error}\}.$$

If $\hat{y} = \textsc{Clean}$, the decision is directly returned; only the reasoning traces routed as Error are passed to the trainable localizer L_θ. For each validated Error reasoning trace, let $g = t^*$ be the authoritative first error step and h be the step currently preferred by the frozen SFT reference localizer. When $h \neq g$, (g, h) constitutes a task-specific hard negative, enabling the optimization to directly compare the authoritative first error step with the location currently favored by the model, rather than an arbitrary incorrect step. We adopt a localization objective with reference-policy regularization:

$$\mathcal{J}_{\mathrm{RL}}(\theta) = \mathbb{E}_{(x,r,g,h)} \left[\log \pi_\theta(g \mid x, r) - \log \pi_\theta(h \mid x, r) \right] - \beta D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) \,.$$

Training assigns exact credit only to the authoritative first error step, while the frozen reference policy is used to limit unnecessary localization-policy drift. Since G_ϕ remains frozen throughout the entire RL stage, RL can only affect first-error localization on reasoning traces that have been routed as Error, and cannot achieve improvement by altering the upstream CLEAN/ERROR decision. The complete RL training configuration and the frozen-gate evaluation protocol are provided in Appendices A.4 and A.6, respectively.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Teacher and trajectory data. Both Native Teacher and Deep-Research Teacher use DeepSeek V4 Pro (DeepSeek-AI et al., 2026). Therefore, the teacher-side comparison mainly reflects differences in the verification procedure rather than differences in the underlying model capabilities. Teacher performance is evaluated on a held-out 200-example confirmatory set and uses the same task definition and scoring method. For student supervision, the initial trajectory-generation stage handles approximately 1,500 mathematical reasoning tasks; after fail-closed trajectory validation, the retained trajectories are compiled into four types of supervision: Decision (D), First-Error (FE), Process (P), and Unified-Deployment (UD). Since Process supervision is the most selective view, this channel is subsequently expanded to 790 validated trajectories using the same frozen Teacher and validation procedure. The complete supervision composition is provided in Appendix A.2..

Student systems and backbones. We use Qwen2.5-Math-7B-Instruct (Yang et al., 2024) as the main student backbone. Base represents a pretrained backbone without verifier-specific training; SFT represents the model after multi-view trajectory distillation and natural-interface deployment alignment; SFT+RL further employs frozen-gate RL to optimize conditional first-error localization while keeping the reasoning-trace-level CLEAN/ERROR decision of SFT unchanged. Besides the main 7B backbone, we further repeat the Base→SFT→SFT+RL process on a total of 10 open-weight backbones to test the cross-backbone generalization of the method; and on 3 representative backbones, we compare our RL objective, PPO, and Bandit-OPD under the same frozen-gate setting. The specific backbones, training budgets, QLoRA configurations, and optimization settings are provided in Appendices A.3–A.5.

Evaluation. DEV151 is used for internal diagnostics and controlled ablations, and contains 24 CLEAN and 127 ERROR reasoning traces. We adopt the 300-trace subset of PRM800K introduced by Lightman et al. (2024) and evaluated by Jiang et al. (2024) as our main external benchmark, which we refer to as PRM300. The subset contains 85 CLEAN and 215 ERROR reasoning traces and is uniformly evaluated through the natural C0 deployment interface, without using PRM300 for checkpoint selection. All compared systems use frozen deterministic inference, prompt construction, output parser, and metric implementation. The complete evaluation, gold-blind parsing, and frozen-gate protocol are provided in Appendix A.6.

Metrics. We report CLEAN Accuracy, Error Detection, Exact FE, FE—Detected, and Balanced Decision. CLEAN Accuracy measures preservation of valid reasoning, Error Detection measures detection of erroneous reasoning traces, Exact FE measures end-to-end exact localization of the authoritative first error step, and FE—Detected measures conditional localization on examples already detected as ERROR. Our main aggregate metric is Strict Verification Accuracy: CLEAN examples are counted as successful only when they are correctly accepted, while ERROR examples must be both detected and exactly localized to the authoritative first error step to be counted as successful.

5.2 TEACHER VERIFICATION

First, we examined whether the Deep-Research verification procedure alters the verification behavior of the Teacher, under the condition of a fixed underlying language model. The Native Teacher and the Deep-Research Teacher were compared on the same held-out 200-example confirmatory set, using the same task definitions, parsing, and evaluation metrics.

Table 1: Confirmatory evaluation of the Native Teacher and Deep-Research Teacher on the held-out 200-example set. Higher is better for all metrics.

Model	CLEAN	Detection	Exact FE	Strict Ver. Acc.	Balanced
Native Teacher	78.00	**97.33**	62.67	66.50	87.67
Deep-Research Teacher	**84.00**	96.67	**64.00**	**69.00**	**90.33**

Table 1 shows that the Deep-Research Teacher does not simultaneously improve all metrics, but rather shifts the verification behavior toward a more balanced operating point. The CLEAN Accuracy increased from 78.00% to 84.00%, Exact FE from 62.67% to 64.00%, Strict Verification Accuracy from 66.50% to 69.00%, Balanced Decision from 87.67% to 90.33%, while Error Detection slightly decreased from 97.33% to 96.67%. Therefore, this procedure does not simply increase the tendency to predict ERROR, but rather maintains near-saturated Error Detection while better preserving valid reasoning and bringing a modest improvement to first-error localization. This is consistent with the design in Section 4.1: source-grounded claims, causal dependencies, and external evidence are used to constrain the final judgment, but tool outputs do not directly determine the authoritative first error step.

The paired comparison yielded a more conservative conclusion: The Deep-Research Teacher corrected 14 errors made by the Native Teacher and introduced 9 regressions. The McNemar's test resulted in $p = 0.4049$. Therefore, we interpret the teacher-side result as a modest directional improvement rather than a statistically significant gain. The earlier FAST40 and BLIND80 results were only reported as development-stage evidence in Appendix C.1; after completing this confirmatory evaluation, the Deep-Research Teacher was frozen for subsequent structured verification trajectory generation and student distillation.

5.3 Trajectory Distillation and SFT Ablations

We first evaluate trajectory distillation on the main external benchmark PRM300. Table 2 compares the pretrained Base verifier with the final deployment-aligned SFT verifier.

Table 2: Effect of trajectory distillation on PRM300.

Model	CLEAN	Detection	Exact FE	FE|Det.	Strict Ver. Acc.	Balanced
Base 7B	**65.88**	49.77	14.88	**29.91**	29.33	57.82
SFT 7B	63.53	**85.12**	**24.19**	28.42	**35.33**	**74.32**

The primary effect of trajectory distillation is to correct the under-detection of the Base verifier. Error Detection increased from 49.77% to 85.12%, Exact FE increased from 14.88% to 24.19%, Strict Verification Accuracy increased from 29.33% to 35.33%, Balanced Decision increased from 57.82% to 74.32%, while CLEAN Accuracy decreased only from 65.88% to 63.53%. Therefore, the higher CLEAN score of the Base is consistent with its conservative operating point and a large number of missed errors; SFT pushed the verifier toward a more balanced decision regime, where the gain in error detection was substantially greater than the loss in clean-reasoning preservation.

Although Exact FE substantially improved, the change in $\mathrm{FE} \mid \mathrm{Detected}$ was smaller, because what Base and SFT actually detected were different subsets of ERROR examples. As SFT substantially reduced under-detection, first-error localization became a more prominent remaining bottleneck, providing motivation for subsequent conditional localization optimization.

As an external reference, the SFT verifier achieved 35.33% Strict Verification Accuracy on PRM300, higher than the published results of GPT-4 Direct-Step (33.33%), GPT-4 Turbo Vanilla (34.00%), and GPT-4 Turbo Direct-Step (35.00%) (Jiang et al., 2024). Since these results come from different models and prompting procedures, we only consider them as non-controlled reference points rather than direct controlled baselines.

SFT ablations. Controlled ablations show that the effect of SFT comes from three aspects: supervision composition, deployment alignment, and training exposure. On DEV151, when using U alone, Strict Verification Accuracy / Exact FE was 25.83% / 20.47%; after adding explicit Decision and First-Error supervision ($D + F + U$), it improved to 31.79% / 26.77%; further adding validated Process supervision ($D + F + P + U$) reached 33.11% / 28.35%, indicating that D/F provided the main gain, while P provided additional and consistent improvement.

Meanwhile, natural-interface deployment alignment increased Strict Verification Accuracy on DEV151 from 21.85% to 30.46%, and Exact FE from 16.54% to 25.98%, suggesting that the capabilities learned through specialized supervision do not automatically become fully expressed under the deployment interface. Process scaling further indicates that supervision diversity and supervision

strength are different factors: when expanding the Process pool from 458 to 790 trajectories while maintaining 1,099 presentations, PRM300 Strict Verification Accuracy remained at 33.67%; after increasing the exposure to 1,895 presentations, it reached 35.33%. Therefore, successful trajectory distillation depends on what supervision is provided, how strong the exposure is during training, and whether these capabilities can be accessed under the deployment interface. Complete ablation results can be found in Appendices C.2–C.4.

5.4 Cross-Backbone SFT Generalization

To test whether the effect of trajectory distillation depends on the main Qwen2.5-Math-7B backbone, we further evaluated Base and SFT on 10 open-weight backbones, covering math-specialized, general-purpose, and code-oriented models. All models adopted the same PRM300 evaluation protocol as in Section 5.1. Table 3 reports the complete Base/SFT/SFT+RL results; this subsection focuses on the Base→SFT comparison.

Table 3: Cross-backbone evaluation on PRM300. Values within each cell are ordered as Base / SFT / SFT+RL. All values are percentages.

Backbone	CLEAN	Detection	Exact FE	Strict Ver. Acc.
Qwen2.5-Math-7B	65.88 / 63.53 / 63.53	49.77 / 85.12 / 85.12	14.88 / 24.19 / 26.51	29.33 / 35.33 / 37.00
Qwen2.5-Math-1.5B	42.35 / 35.29 / 35.29	45.58 / 85.12 / 85.12	10.70 / 17.21 / 20.00	19.67 / 22.33 / 24.33
DeepSeekMath-7B	36.47 / 28.24 / 28.24	60.47 / 88.37 / 88.37	7.91 / 20.00 / 21.40	16.00 / 22.33 / 23.33
Mathstral-7B	71.76 / 44.71 / 44.71	50.23 / 87.91 / 87.91	10.70 / 32.09 / 32.09	28.00 / 35.67 / 35.67
Qwen2.5-7B-Instruct	64.71 / 52.94 / 52.94	43.26 / 85.12 / 85.12	9.77 / 32.56 / 31.63	25.33 / 38.33 / 37.67
Qwen2.5-Coder-7B	37.65 / 37.65 / 37.65	84.65 / 89.77 / 89.77	20.47 / 29.30 / 30.70	25.33 / 31.67 / 32.67
DeepSeek-Coder-6.7B	10.59 / 36.47 / 36.47	70.23 / 87.44 / 87.44	5.58 / 14.88 / 14.88	7.00 / 21.00 / 21.00
OLMo-2-7B	11.76 / 47.06 / 47.06	96.74 / 80.47 / 80.47	16.74 / 17.21 / 15.35	15.33 / 25.67 / 24.33
Phi-4-mini	48.24 / 43.53 / 43.53	68.37 / 89.30 / 89.30	12.09 / 27.44 / 26.51	22.33 / 32.00 / 31.33
Granite-3.3-8B	58.82 / 49.41 / 49.41	40.47 / 86.98 / 86.98	4.65 / 33.49 / 28.37	20.00 / 38.00 / 34.33

The cross-backbone results show that Base→SFT improved Strict Verification Accuracy and Exact FE across all 10 models, although the initial verification behaviors of different Base models varied greatly. For instance, the Strict Verification Accuracy of Qwen2.5-Math-7B increased from 29.33% to 35.33%, Qwen2.5-7B-Instruct from 25.33% to 38.33%, Mathstral-7B from 28.00% to 35.67%, while Granite-3.3-8B increased from 20.00% to 38.00%. Corresponding Exact FE also changed from 14.88% to 24.19%, 9.77% to 32.56%, 10.70% to 32.09%, and 4.65% to 33.49%. These results indicate that the verification behavior learned through Deep-Research-grounded trajectory distillation is not limited to a single backbone or a single model family.

This improvement does not simply push all models toward a higher ERROR prediction rate. Most initially under-detecting backbones showed a substantial improvement in Error Detection after SFT; however, OLMo-2-7B's Base already had 96.74% Error Detection and only 11.76% CLEAN Accuracy. After SFT, Detection was adjusted to 80.47%, while CLEAN Accuracy increased to 47.06%, and Strict Verification Accuracy simultaneously increased from 15.33% to 25.67%. Therefore, the more consistent cross-model trend is that SFT pushes different initial operating points toward more effective verification behavior, while enhancing end-to-end first-error localization capability, rather than simply learning a unified CLEAN/ERROR decision bias.

5.5 Frozen-Gate RL and Optimization Analysis

After completing SFT, we further examined whether frozen-gate RL could improve conditional first-error localization without changing the reasoning-trace-level CLEAN/ERROR decision. We refer to the complete RL configuration consisting of the pairwise hard-negative localization objective and frozen-reference KL regularization in Section 4.3 as *Pairwise PG + KL*. Since all RL systems reuse the frozen SFT decision gate, CLEAN Accuracy and Error Detection remain unchanged before and after RL. Therefore, any performance changes arise from localization behavior on reasoning traces already routed as Error.

On the primary Qwen2.5-Math-7B backbone, the deployment-aligned SFT verifier achieved 35.33% Strict Verification Accuracy on PRM300. With the same frozen decision gate but using a pre-RL conditional localizer, Strict Verification Accuracy reached 36.00%, and Pairwise PG + KL further increased it to 37.00%. Thus, $35.33\% \rightarrow 36.00\% \rightarrow 37.00\%$ separates the change introduced by the conditional localization interface from the additional gain produced by the RL update.

The component ablation on DEV151 provides a more direct analysis of the RL objective. The pre-RL localizer achieved 28.35% Exact FE and 33.11% Strict Verification Accuracy. Pairwise hard-negative optimization without KL increased these metrics to 33.86% and 37.75%, respectively, while the complete Pairwise PG + KL configuration further reached 35.43% Exact FE and 39.07% Strict Verification Accuracy. Paired analysis indicates that the main gain comes from the pairwise localization update, while frozen-reference KL provides a smaller additional stabilizing effect. Complete component metrics and paired statistics are reported in Appendix C.5.

To test whether the improvement can be obtained by using any RL optimizer, we compared Pairwise PG + KL, PPO, and Bandit-OPD on three representative backbones: Qwen2.5-Math-7B, Qwen2.5-7B-Instruct, and Mathstral-7B, under the same frozen-gate setting. Pairwise PG + KL achieved higher Exact FE, $\mathrm{FE} \mid \mathrm{Detected}$, and Strict Verification Accuracy than PPO and Bandit-OPD on all three backbones. On Qwen2.5-Math-7B, the Strict Verification Accuracy of Pairwise PG + KL, PPO, and Bandit-OPD was 37.00%, 36.33%, and 34.33%, respectively; on Qwen2.5-7B-Instruct, it was 37.67%, 37.00%, and 34.33%; and on Mathstral-7B, it was 35.67%, 35.00%, and 32.33%. Correspondingly, Pairwise PG + KL improved Exact FE and Strict Verification Accuracy by 0.93 and 0.67 percentage points, respectively, over PPO on all three backbones, with larger margins over Bandit-OPD. The complete optimizer comparison is reported in Appendix C.6.

As shown in Table 3, the benefits of frozen-gate RL exhibit stronger backbone dependence than those of SFT. Across the 10 backbones, SFT→SFT+RL further improved Strict Verification Accuracy on 4 backbones, remained unchanged on 2 backbones, and regressed on the remaining 4 backbones. Because the CLEAN/ERROR decision gate remains frozen throughout RL, these differences arise from conditional localization rather than changes in upstream detection behavior. Therefore, we do not treat RL as a universal stage that consistently improves all backbones; instead, cross-backbone transfer of trajectory distillation is more stable, while the localization gain of frozen-gate RL is more sensitive to the backbone and localization interface.

6 Discussion and Conclusion

Our results indicate that mathematical reasoning verification involves three related but distinct bottlenecks: verification grounding, reasoning-trace-level error detection, and causal first-error localization. The Deep-Research Teacher constrains verification grounding, while Deep-Research-grounded trajectory distillation improves error detection and first-error localization; Base→SFT improves Strict Verification Accuracy and Exact FE across 10 open-weight backbones, showing that this supervision transfer is not limited to a single model family. Frozen-gate RL further targets conditional localization: Pairwise PG + KL outperforms PPO and Bandit-OPD on three representative backbones, although its gains are more backbone-dependent than those of SFT. These results support using different interventions for distinct verification bottlenecks rather than optimizing an undecomposed verifier objective.

We introduced Deep-Research-Grounded Trajectory Distillation (DRGTD) for mathematical reasoning verification. The framework combines Deep-Research teacher verification, fail-closed trajectory validation, multi-view trajectory distillation, deployment alignment, and frozen-gate reinforcement learning for conditional first-error localization. On the primary Qwen2.5-Math-7B backbone, Strict Verification Accuracy increased from 29.33% for Base to 35.33% after SFT and 37.00% after RL. Across 10 open-weight backbones, Base→SFT improved both Strict Verification Accuracy and Exact FE, while frozen-gate RL further targeted localization without altering the learned reasoning-trace-level CLEAN/ERROR decision. Overall, the results show that structured and validated teacher trajectories can be transferred to compact verifiers to improve error detection and first-error localization, with subsequent optimization restricted to the remaining localization bottleneck.

ETHICS STATEMENT

This study involves mathematical reasoning verification. No personal or sensitive information is collected, nor are any human experiments conducted. Subject. The proposed framework aims to enhance the reliability of reasoning verification. However, neither the in-depth research teachers nor the refined verifiers can guarantee the accuracy of their judgment results. Particularly, this system may still overlook reasoning errors, wrongly reject valid reasoning, or misjudge the authoritative first error steps. External mathematical tools can provide additional evidence, but they cannot constitute a formal proof of the complete reasoning path. Therefore, the output of this validator should not be regarded as the final mathematical certification, nor should it be used as the sole basis for high-risk decisions. We clearly point out these limitations and evaluate the method under controlled and fixed protocols to reduce unverified claims about the model's reliability.

REPRODUCIBILITY STATEMENT

In the appendix, we provide detailed configurations for the Teacher-side, SFT (supervised fine-tuning), RL (reinforcement learning), and evaluation process. For the Teacher-side experiments, the underlying model was fixed under Native Teacher and Deep-Research Teacher conditions, and the API configuration, two-call verification protocol, tool/runtime interface, execution budget, evidence provenance, and fail-closed trajectory-validation procedure were recorded. For student training, we report the precise backbone model revision, QLoRA configuration, software environment, random seed, optimizer, learning rate, training steps, and supervision exposure. The RL configuration and frozen-gate evaluation protocol are also fully described. All student evaluations adopt frozen deterministic inference, prompt construction, tokenizer, output parser, and metric implementation, and candidate predictions are frozen before gold scoring. To accurately reproduce the Teacher-side outputs, the specified external DeepSeek API model version needs to be continuously provided. We will release all the data, code, model and other related resources in a public Github repository later.

AI USE STATEMENT

In this work, we used generative AI tools for tasks that will be disclosed in accordance with the ICLR 2027 AI Policy for Authors. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, analyses, or artifacts produced with the aid of generative AI.

REFERENCES

Qikai Chang, Zhenrong Zhang, Pengfei Hu, Jun Du, Jiefeng Ma, Yicheng Pan, Jianshu Zhang, Quan Liu, and Gao Jianqing. Thor: Tool-integrated hierarchical optimization via rl for mathematical reasoning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 21487–21510, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/245b7f8357fdc26797ed6714cda5bd94-Paper-Conference.pdf.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. Improve student's reasoning generalizability through cascading decomposed CoTs distillation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15623–15643, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.875. URL https://aclanthology.org/2024.emnlp-main.875/.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya

Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Dayiheng Liu, Chang Zhou, Wen Xiao, Tianyu Liu, and Baobao Chang. LLM critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14588–14604, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.753. URL https://aclanthology.org/2025.findings-acl.753/.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/fef126561bbf9d4467dbb8d27334b8fe-Paper-Conference.pdf.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 48362–48395, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d3cf1559a8795eb1ed2b3ad52409ac7d-Paper-Conference.pdf.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Jiacheng Guo, Yue Wu, Jiahao Qiu, Kaixuan Huang, Xinzhe Juan, Ling Yang, and Mengdi Wang. Temporal consistency for LLM reasoning process error identification. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22114–22129, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1205. URL https://aclanthology.org/2025.findings-emnlp.1205/.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4615–4634, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.254. URL https://aclanthology.org/2024.acl-long.254/.

Zhuoxuan Jiang, Haoyuan Peng, Shanshan Feng, Fan Li, and Dongsheng Li. Llms can find mathematical reasoning mistakes by pedagogical chain-of-thought. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 3439–3447. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/381. URL https://doi.org/10.24963/ijcai.2024/381. Main Track.

Minki Kang, Jongwon Jeong, and Jaewoong Cho. T1: Tool-integrated verification for test-time compute scaling in small language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 73413–73444, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/776a5f2c7d6dd4b0d83145fc044e2726-Paper-Conference.pdf.

Xuzhao Li, Xuchen Li, Shiyu Hu, Yongzhen Guo, and Wentao Zhang. VerifyBench: A systematic benchmark for evaluating reasoning verifiers across domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):31796–31804, 2026. doi: 10.1609/aaai.v40i38.40448. URL https://doi.org/10.1609/aaai.v40i38.40448.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12171–12186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.594. URL https://aclanthology.org/2025.acl-long.594/.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.882. URL https://aclanthology.org/2024.findings-emnlp.882/.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/98711dea460bdefe0e651ca23ec98ba2-Paper-Conference.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965. Curran Associates, Inc., 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ed3fea9033a80fea1376299fa7863f4a-Paper-Conference.pdf.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13894–13908, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.826. URL https://aclanthology.org/2024.findings-acl.826/.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. Investigating mysteries of CoT-augmented distillation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6071–6086, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.349. URL https://aclanthology.org/2024.emnlp-main.349/.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. SCOTT: Self-consistent chain-of-thought distillation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5546–5558, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.304. URL https://aclanthology.org/2023.acl-long.304/.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516,

Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.

APPENDIX

A ADDITIONAL EXPERIMENTAL DETAILS

A.1 TEACHER MODEL AND API CONFIGURATION

Both the Native Teacher and Deep-Research Teacher use the same underlying language model. The Native Teacher performs the verification task directly, whereas the Deep-Research Teacher augments the same model with the source-grounded research, external-tool verification, evidence tracking, and causal final-judgment procedure described in Section 4.1.

Table A.1 summarizes the configuration shared by the two teacher conditions. Protocol-specific verification behavior differs by design; the Deep-Research Teacher protocol and tool interface are described separately in Appendix B.2.

Table A.1: Shared teacher-model and API configuration.

Configuration	Setting
Underlying model	DeepSeek V4 Pro
API provider	Official DeepSeek API
Model identifier	`deepseek-v4-pro`
Controlled comparison	Same underlying model for Native and Deep-Research verification

Holding the underlying model fixed allows the teacher-side comparison in Section 5.2 to isolate the effect of the verification procedure rather than differences in base-model capability.

A.2 SUPERVISION COMPOSITION

The initial trajectory-generation stage processes approximately 1,500 teacher-processed tasks. After constrained re-checking, the retained supervision contains Decision, First-Error, Process, and Unified-Deployment views. Because Process supervision is the most selective view, this channel is subsequently expanded using additional trajectories generated by the same frozen teacher and admitted through the same validation procedure. Table A.2 summarizes the resulting supervision composition.

Table A.2: Supervision composition used for student training.

View	Post-re-check	Final	Supervision role
Decision (D)	1,173	1,173	CLEAN / ERROR decision
First-Error (F)	952	952	Conditional first-error localization
Process (P)	466	790	Causal and evidence-grounded verification process
Unified Deployment (U)	1,173	1,173	Natural deployment behavior

The 466 Process examples in the post-re-check column denote the Process-eligible population after constrained validation. This population should be distinguished from the frozen 458-trajectory Process training pool used as the reference condition in the controlled Process-scaling experiment of Appendix C.4. The final Process pool contains 790 unique validated trajectories and receives 1,895 Process-view training presentations in the exposure-matched SFT configuration.

A.3 SFT Training and Exposure Budget

Optimization objectives. In Phase 1, training examples are sampled from the validated supervision views $\mathcal{V} = \{D, F, P, U\}$ according to a controlled exposure policy π. For a sampled view v with input $x^{(v)}$ and target $y^{(v)}$, we optimize

$$\mathcal{L}_{\mathrm{MV}}(\theta) = \mathbb{E}_{v\sim\pi}\left[-\log p_\theta\left(y^{(v)} \mid x^{(v)}\right)\right]. \tag{5}$$

Phase 2 performs deployment alignment by pairing the unified supervision with the corresponding natural C0 target. The paired objective is

$$\mathcal{L}_{\mathrm{pair}}(\theta) = \frac{1}{2}\mathcal{L}_U(\theta) + \frac{1}{2}\mathcal{L}_{C0}(\theta), \tag{6}$$

where $\mathcal{L}_U$ and $\mathcal{L}_{C0}$ are the mean target-token cross-entropy losses of the unified and natural-interface branches, respectively.

The final supervised verifier is trained in two consecutive stages. Phase 1 performs multi-view capability learning using validated $D/F/P/U$ supervision, while Phase 2 continues from the Phase 1 checkpoint under the natural deployment interface. Table A.3 summarizes the final training budget.

Table A.3: Training budget for the final supervised verifier.

Stage	Steps	P exposure	Purpose
SFT Phase 1	1,573	1,895	Multi-view capability learning
SFT Phase 2	391	–	Natural deployment alignment
Total SFT	1,964	–	Final deployment-aligned verifier

Phase 1 uses the expanded pool of 790 unique validated Process trajectories. Phase 2 introduces no new teacher trajectories or mathematical supervision; it continues optimization from the Phase 1 checkpoint using authoritative targets under the natural inference interface.

A.4 RL Training Configuration

The final reinforcement-learning stage starts from the exposure-matched SFT verifier and optimizes only conditional first-error localization. The trajectory-level CLEAN/ERROR decision remains frozen throughout RL. Table A.4 summarizes the final configuration.

For mislocalized examples, the current SFT prediction serves as the hard negative and the authoritative first error step as the positive action. The update therefore targets the model's actual competing localization rather than an arbitrary negative position. BEFORE/AFTER contextual evidence is retained from the frozen SFT reference, while RL credit is restricted to the choice of which candidate step should be treated as the FIRST error. EARLY- and LATE-direction corrections are balanced during training to avoid systematically shifting localization in one direction.

Table A.4: Final configuration for conditional first-error reinforcement learning.

Component	Final configuration
Starting policy	Final exposure-matched SFT verifier
Decision behavior	Frozen SFT CLEAN/ERROR gate
Localization data	952 validated ERROR examples
Reference localization	222 EARLY / 508 EXACT / 222 LATE
Correction target	Gold FE vs. current SFT-predicted FE
Credit assignment	FIRST-only localization preference
Optimization	Pairwise exact contextual-bandit policy optimization
Reference regularization	KL to frozen SFT localization policy
KL coefficient	$\beta = 0.02$
Learning rate	2×10^{-6}
Training epochs	1
Optimizer steps	238

A.5 Student Backbone and SFT Optimization Configuration

Our primary student verifier is initialized from `Qwen/Qwen2.5-Math-7B-Instruct`, using the frozen model revision `ef9926d75ab1d54532f6a30dd5e760355eb9aa4d`. All headline Base, SFT, and SFT+RL results in the main paper use this backbone. **Base** denotes the pretrained checkpoint without verifier-specific training; **SFT** denotes the same backbone after the two-stage multi-view distillation and deployment-alignment procedure in Section 4.2; and **SFT+RL** further applies the conditional first-error localization stage in Section 4.3 while preserving the frozen SFT trajectory-level CLEAN/ERROR decision.

We use parameter-efficient QLoRA training (Dettmers et al., 2023) with 4-bit NF4 quantization and double quantization. LoRA adapters are applied to the attention projections (`q_proj`, `k_proj`, `v_proj`, and `o_proj`) and MLP projections (`gate_proj`, `up_proj`, and `down_proj`). Table A.5 summarizes the frozen primary-student training configuration.

Phase 1 performs multi-view capability learning using the validated $D/F/P/U$ supervision. The final matched-exposure configuration keeps the 790-example Process corpus fixed while increasing Process-view presentations from the fixed-budget setting to 1,895. This approximately restores the per-trajectory Process exposure density of the earlier 458-example training pool while leaving the non-Process supervision sequences unchanged.

Phase 2 continues from the Phase 1 adapter for 391 optimizer steps using the frozen natural-interface alignment schedule. No new teacher trajectories or mathematical supervision are introduced in this stage. The optimizer and scheduler are reset at the phase boundary while retaining the same model and adapter configuration. The final two-stage SFT budget is therefore 1,964 optimizer steps.

The frozen software environment uses `transformers==4.57.6`, `peft==0.20.0`, `accelerate==1.14.0`, and `bitsandbytes==0.50.2`.

Table A.5: Frozen optimization configuration for the primary `Qwen2.5-Math-7B-Instruct` student verifier.

Configuration	Setting
Primary backbone	`Qwen/Qwen2.5-Math-7B-Instruct`
Frozen model revision	`ef9926d75ab1d54532f6a30dd5e760355eb9aa4d`
Maximum sequence length	4,096 tokens
Quantization	4-bit NF4 with double quantization
LoRA rank	$r = 16$
LoRA scaling	$\alpha = 32$
LoRA dropout	0.05
LoRA target modules	`q/k/v/o_proj`, `gate/up/down_proj`
Micro-batch size	1
Gradient accumulation	4
Learning rate	2×10^{-5}
Optimizer	`paged_adamw_8bit`
Learning-rate schedule	Cosine
Warmup ratio	0.05
Weight decay	0.0
Maximum gradient norm	1.0
Random seed	48
Phase 1 optimizer steps	1,573
Phase 1 unique Process trajectories	790
Phase 1 Process presentations	1,895
Phase 2 optimizer steps	391
Total SFT optimizer steps	1,964
Phase-boundary behavior	Optimizer and scheduler reset before Phase 2

A.6 Evaluation and Parsing Protocol

All student evaluations use frozen inference and parsing procedures that are separated from training and checkpoint selection. The primary deployed interface is the natural C0 verification interface, under which the model receives the mathematical problem and the complete candidate reasoning trace and returns a trajectory-level verification judgment together with an first-error location when applicable.

Evaluation sets and interfaces. DEV151 is used as an internal development and diagnostic set and contains 151 reasoning traces, including 24 CLEAN and 127 ERROR examples. It supports the specialized Decision (D), First-Error (F), Unified (U), and natural C0 interfaces used in controlled ablations. Headline student results are reported on PRM300, which contains 300 reasoning traces, including 85 CLEAN and 215 ERROR examples. PRM300 is evaluated only through the natural C0 deployment interface.

Table A.6: Frozen student-evaluation interfaces used in the main experiments.

Evaluation	Composition	Interface	Role
DEV151	24 CLEAN / 127 ERROR	D, F, U, and C0	Internal diagnostics and controlled ablations
PRM300	85 CLEAN / 215 ERROR	Natural C0	Headline evaluation

Deterministic generation. Evaluation uses deterministic greedy decoding with sampling disabled. The maximum context length is 4,096 tokens. For the specialized D, F, and U diagnostic interfaces, generation is capped at 24 new tokens. Under the natural C0 interface, the available completion budget is the portion of the 4,096-token context window remaining after the complete evaluation prompt has been encoded. When batch padding would artificially reduce the available completion budget for a longer example, that example is rerun individually with its full remaining context budget.

All systems compared under the same evaluation interface use the same frozen prompt construction, tokenizer, output parser, and metric implementation. No checkpoint is selected on PRM300, and evaluation does not update model parameters.

Gold-blind candidate extraction. Gold labels are not provided to the model during generation and are not used to construct the parsed candidate prediction. Evaluation payloads exposed to the model contain only the mathematical problem, candidate reasoning trace, and visible reasoning-step identifiers. The evaluation protocol, parser, model endpoint, and model-visible input hashes are frozen before inference.

Raw model generations are first converted into candidate predictions using the frozen parser, and the complete candidate prediction file is frozen before any gold scoring is performed. Gold labels are accessed only afterward for metric computation. Consequently, unresolved outputs are not repaired using the authoritative label or the correct first-error position, and no candidate prediction is modified in response to the gold annotation.

Output parsing. The specialized diagnostic interfaces use strict output contracts. The Decision view accepts only a trajectory-level `Decision: CLEAN` or `Decision: ERROR` response. The First-Error view accepts a single `First-Error: Step <id>` prediction whose step identifier must occur in the visible reasoning trace. The Unified view similarly resolves either an explicit no-error decision or one visible first-error step.

For the natural C0 interface, we first apply the frozen semantic extractor used by the historical deployment protocol. A resolved first-error prediction is accepted only if the predicted step identifier occurs in the visible source trace. If the primary semantic extraction remains unresolved, a frozen gold-blind recovery parser inspects the model's final-conclusion region for either an explicit no-error judgment or an explicit first-error step.

The recovery parser is fail-closed. A unique valid first-error identifier is accepted, and an explicit no-error judgment is interpreted as CLEAN. Predictions containing an invalid step identifier, conflicting no-error and first-error judgments, or multiple distinct first-error candidates remain unresolved. Outputs with no recoverable decision likewise remain unresolved. Unresolved predictions receive no correctness credit.

Metric computation. For a CLEAN example, the prediction is correct only when parsing resolves to no error. For an ERROR example, Error Detection follows the output contract of the corresponding interface. Under the Decision (D) interface, detection succeeds when parsing resolves to `Decision: ERROR`; under the Unified (U) and natural C0 interfaces, it succeeds when parsing resolves to a valid non-null error location. Exact First-Error succeeds only when the predicted location exactly matches the authoritative first error step.

Let N_{Clean} and N_{Error} denote the numbers of CLEAN and ERROR examples, respectively; let $N_{\text{CorrectClean}}$ denote correctly accepted CLEAN examples, $N_{\text{DetectedError}}$ detected ERROR examples, and N_{ExactFE} ERROR examples with the authoritative first errory step exactly localized. We compute

$$\text{CLEAN Accuracy} = \frac{N_{\text{CorrectClean}}}{N_{\text{Clean}}}, \qquad \text{Error Detection} = \frac{N_{\text{DetectedError}}}{N_{\text{Error}}},$$

$$\text{ExactFE} = \frac{N_{\text{ExactFE}}}{N_{\text{Error}}}, \qquad \text{FE} \mid \text{Detected} = \frac{N_{\text{ExactFE}}}{N_{\text{DetectedError}}},$$

$$\text{BalancedDecision} = \frac{\text{CLEAN Accuracy} + \text{Error Detection}}{2}.$$

Strict Verification Accuracy follows the end-to-end definition in Section 3.2 and is not repeated here.

Frozen-gate RL evaluation. The SFT+RL system reuses the trajectory-level C0 predictions of the frozen SFT verifier as an immutable decision gate. If the frozen gate predicts CLEAN, that decision is returned directly and the conditional localizer is never invoked. If the gate predicts ERROR, the same example is passed to the first-error localizer.

Pre-RL and post-RL localization are therefore evaluated on exactly the same set of gate-routed examples. The trajectory-level PRM gate decisions are frozen before localizer scoring. In addition, the pre-RL and post-RL localization candidate files are frozen before the PRM300 gold labels are accessed for scoring. Neither the gate decisions nor the localization candidates are subsequently repaired or altered using gold information.

As a result, CLEAN Accuracy and Error Detection are invariant by construction between the pre-RL and post-RL frozen-gate systems. Balanced Decision is therefore invariant as well. The RL intervention can affect only the conditional localization behavior on examples that were already routed as ERROR by the frozen SFT gate. Accordingly, changes after RL appear through localization-dependent quantities such as Exact First-Error, $\mathrm{FE} \mid \mathrm{Detected}$, and Strict Verification Accuracy, rather than through changes in the trajectory-level error-detection decision.

This frozen-gate design isolates the effect of reinforcement learning on first-error localization from changes in trajectory-level verification: the set of examples presented to the localizer is held fixed, while only the localization policy is changed between the pre-RL and post-RL conditions.

B TEACHER AND TRAJECTORY CONSTRUCTION DETAILS

B.1 CONSTRAINED RE-CHECK GATE DEFINITIONS

The constrained re-check validates information already present in each teacher trajectory without independently re-solving the underlying mathematical problem. Table B.1 summarizes the six ordered validation gates. Failed information is removed or downgraded rather than reconstructed through additional model generation.

Table B.1: Ordered fail-closed checks used to validate teacher verification trajectories.

Gate	Criterion	Validation	Failure action
G0	Provenance	Verify task identity, source/step correspondence, source hashes, alignment between teacher results and trajectory records, and references among source steps, claims, and evidence.	Exclude content whose provenance cannot be reconstructed.
G1	Structural integrity	Verify references among claims, dependencies, verification obligations, states, evidence, and final judgments. Planned operations are distinguished from tool calls that were actually executed.	Reject structurally unsupported fields or views.
G2	Causal relevance	Require retained process information to bear on the final decision. For ERROR examples, it must contribute to first-error localization; a downstream false statement alone is insufficient.	Remove or downgrade non-causal process supervision.
G3	Evidence quality	Verify that evidence was actually executed, remains bound to the intended source proposition, provides non-trivial support or contradiction, and is causally role-aligned.	Exclude weak, unbound, synthetic, or unsupported evidence.
G4	Gold consistency	Only after G0–G3 are fixed, compare the retained decision and, when applicable, first-error target with the authoritative training label. Gold is not used to generate replacement reasoning.	Reject inconsistent supervision views.
G5	Information preservation	Verify that compilation into the SFT record preserves validated claims, dependencies, states, evidence, and source references without silent loss or remapping.	Reject views whose validated information is not faithfully preserved.

B.2 Frozen Teacher Protocol and Tool Interface

The Deep-Research Teacher used throughout trajectory generation follows a frozen two-call verification protocol separated by a deterministic runtime. The first model call constructs a source-grounded semantic research representation, the runtime selectively compiles and executes mathematical checks, and the second model call integrates the resulting evidence into the final causal verification judgment. No additional model call is introduced for planning, criticism, selection, or revision.

Source-grounded semantic research. The first call receives the complete problem, the complete candidate reasoning trace, and deterministic aliases for mathematical atoms extracted from the immutable source. It identifies step-local claims, causal dependencies, and candidate verification obligations. For substantive reasoning steps, at least one primary semantic claim is requested, while dependency edges must preserve the causal direction of the original reasoning.

This stage is explicitly prohibited from predicting, ranking, or discussing the first error step. Its role is to specify *what* should be verified, not to decide the final FE location. Each candidate obligation identifies a target claim, explicit premise claims, a semantic operation, and compact toolability and decision-value metadata. The research stage may propose at most eight candidate obligations.

Executable details are not freely authored by the model. Instead, the runtime constructs the verification program from the selected source-bound claims and mathematical atoms, implementing the separation

LLM: WHAT to verify vs. runtime: HOW to execute it.

The exposed verification operations include relation and equivalence checking, implication verification, counterexample search, constraint solving, and expression evaluation. Exact source-bound checks are preferred when available. If the intended proposition cannot be faithfully compiled from the bound source information, the obligation remains non-toolable or unresolved rather than being replaced by an easier surrogate.

Selective causal execution and evidence. The teacher does not execute every toolable candidate. Verification budget is allocated according to causal decision relevance. In particular, high-value, premise-ready programs that can clarify an earlier causal frontier are preferred over checks that are merely easy to execute or concern only downstream consequences.

The execution budget contains four base logical checks and an adaptive extension of at most two additional checks, yielding a maximum of six executed logical/tool checks per task. The extension is used only while a high-value unresolved causal program remains.

Each attempted or executed verification is recorded in an evidence ledger together with its source binding, target claim, executed proposition, tool result, evidence strength, claim-state transition, and provenance information. Decisive evidence is accepted only when the executed proposition remains semantically aligned with the intended source-bound target. Evidence that is unbound, mismatched, or otherwise semantically unaudited is quarantined rather than silently treated as mathematical authority.

Evidence-conditioned causal judgment. The second model call receives the complete immutable source together with the research representation, accumulated evidence, and current claim states. The original reasoning trace remains authoritative: structured claims and tool results constrain the final judgment but do not replace direct causal inspection of the source.

The final protocol enforces *first-error prefix minimality*. A prediction of `FIRST_ERROR` at step k is valid only when the earlier source prefix has been audited for a substantive error. A later verified-false claim therefore acts as an error anchor rather than automatically becoming the first error; the verifier must trace such evidence toward its causal origin and reconsider earlier transitions.

Two causal bases are distinguished for a first-error judgment. `EVIDENCE_GROUNDED` is used when trusted decisive evidence directly supports the causal claim at the predicted first error step. `SOURCE_ONLY_CAUSAL` is used when the original reasoning establishes the root cause but the tool

interface cannot faithfully reach that proposition. Thus, lack of toolability does not by itself force an unresolved verdict, while tool evidence cannot be promoted into an FE label without causal source analysis.

Trusted `VERIFIED_TRUE` information before a proposed FE serves as prefix protection, whereas trusted non-quarantined `VERIFIED_FALSE` information acts as a causal constraint. A deterministic post-final normalizer may repair structural or provenance inconsistencies, but it is not permitted to change the task verdict, first-error location, forced submission, or causal decision basis, nor may it reinterpret tool evidence.

Table B.2: Authority separation in the frozen Deep-Research Teacher.

Stage	Primary authority	Key constraint
Research call	Claims, dependencies, and verification targets	No FE hypothesis or executable-tool rewriting
Runtime	Source binding, compilation, scheduling, execution, and provenance	Fail closed on unfaithful mappings; at most six executed checks
Final call	Causal verification and first-error judgment	Must satisfy prefix minimality and distinguish causal roots from downstream errors
Normalizer	Structural/provenance consistency	Cannot change verdict, FE, or evidence semantics

B.3 Structured Verification Trajectory Schema

The teacher trajectory is a structured record of the observable verification process rather than an unrestricted chain-of-thought transcript. As summarized in Section 4.2, we represent a trajectory abstractly as

$$\tau = (\mathcal{S}, \mathcal{C}, \mathcal{G}, \mathcal{O}, \mathcal{E}, \mathcal{Z}, \mathcal{J}),$$

corresponding to source coverage, semantic claims, causal dependencies, verification obligations, tool evidence, evolving claim states, and the final causal judgment.

Immutable Source $\longrightarrow$ **Claims / Dependencies** $\longrightarrow$ **Verification Obligation**
$\longrightarrow$ **Executed Proposition** $\longrightarrow$ **Evidence** $\longrightarrow$ **Claim State** $\longrightarrow$ **Causal FE Judgment**

Figure 2: Minimal data flow of a structured teacher verification trajectory. Each transition retains source identity and provenance rather than collapsing the process into a single final label.

Table B.3 summarizes how these components are instantiated in the frozen teacher records.

Source and proposition identity. Problem and reasoning-trace hashes bind each trajectory to its immutable input, while individual claims retain source-step and source-span identity. Deterministically extracted mathematical atoms carry canonical formal representations whose hashes are recomputed by the runtime rather than trusted from model output.

Verification obligations similarly retain a frozen identity for the mathematical question being asked. Compilation may attempt different executable representations, but these attempts remain associated with the same frozen verification target. The evidence ledger records both the intended source-bound proposition and the proposition actually executed, together with their provenance and applicability information.

Evidence and state separation. A raw tool result is not itself a claim-state judgment. The trajectory separately records the executed result, its evidence strength, its semantic applicability to the

Table B.3: Principal components of the structured verification trajectory.

Symbol	Record	Retained information
$\mathcal{S}$	Source coverage	Source-step identity and coverage status, source hashes, and the set of claims associated with each reasoning step.
$\mathcal{C}$	Semantic claim	Claim identifier, source step and source span, normalized semantic claim, claim type, primary-claim status, and deterministic source/formal binding information when available.
$\mathcal{G}$	Dependency	Directed links from premise claims to a target claim together with the dependency type, preserving the causal direction of the reasoning trace.
$\mathcal{O}$	Verification obligation	Target claim, explicit premise claims, source step, semantic verification operation, toolability and causal-value metadata, priority, and a frozen verification-target identity.
$\mathcal{E}$	Evidence ledger	Compilation attempts, executed proposition, tool result and witness, evidence strength and applicability, claim effect, state transition, and provenance information.
$\mathcal{Z}$	Claim state	Current epistemic state of a claim, evidence references, and causal flags indicating downstream dependence, unresolved ancestry, or conflicting evidence.
$\mathcal{J}$	Final judgment	Trajectory verdict, first-error location when applicable, causal and downstream-support claims, evidence references, mapping failures, causal explanation, confidence, and decision basis.

intended source proposition, and the resulting state transition. Decisive state updates require sufficiently aligned source and executed representations; otherwise the evidence remains supportive, inconclusive, unexecutable, or quarantined.

This preserves the distinction between a mathematically valid tool result, its applicability to the intended source claim, and the resulting epistemic state of that claim.

Causal rather than purely local structure. Dependencies and claim states are retained because a false statement need not be the causal origin of an error. A verified-false claim can provide an error anchor while its dependency structure points to an earlier error transition. Conversely, verified-true claims in an earlier prefix can protect that prefix from being incorrectly relabeled as erroneous. The final judgment therefore distinguishes causal claims from downstream-support claims rather than treating all false statements as equivalent FE candidates.

Compilation into student supervision. The complete trajectory is richer than any individual student-training view. Following the fail-closed validation in Appendix B.1, only validated information is compiled into supervision. Decision supervision uses the trajectory-level decision signal; First-Error supervision uses validated ERROR examples with authoritative first-error targets; Process supervision retains validated causal links among claims, dependencies, verification obligations, evidence, state updates, and causal localization; and Unified-Deployment supervision expresses the compact behavior required at inference time.

Missing or rejected process information is not reconstructed merely to make the supervision views structurally uniform. Consequently, failure of an evidence or process field can remove that information from Process supervision without necessarily invalidating an otherwise usable decision or localization target.

C Additional Results and Ablations

C.1 Earlier Teacher Evaluations

Before the final 200-example confirmatory evaluation, we conducted two smaller teacher-side evaluations during development of the verification procedure. They are reported for completeness but are not used as the primary evidence for the teacher-side claim in the main text.

FAST40 was used during development of the Deep-Research verification pipeline and therefore is not an independent confirmatory set. BLIND80 was subsequently used as a blind held-out check. The final 200-example evaluation reported in Section 5.2 was conducted after the teacher procedure had been fixed and is used as the primary confirmatory result.

Table C.1: Earlier evaluations of the teacher-verification procedure.

Evaluation	N	Native	Deep-Research	Δ	Corr./Reg.	Role
FAST40	40	55.00	65.00	+10.00	5 / 1	Development
BLIND80	80	68.75	76.25	+7.50	10 / 4	Blind held-out

On FAST40, native verification solves 22 of 40 examples (55.00%), whereas the Deep-Research procedure solves 26 of 40 (65.00%). The agent corrects five examples that are incorrect under native verification while regressing on one previously correct example. Because FAST40 was used during development, this result is treated only as early development evidence.

On BLIND80, native verification achieves 68.75% accuracy and the Deep-Research Teacher achieves 76.25%, corresponding to a 7.50 percentage point improvement. The paired comparison contains 10 corrections and 4 regressions. This result is directionally consistent with FAST40, but the larger held-out 200-example evaluation remains the sole confirmatory teacher-side result used for our primary conclusions.

C.2 Multi-View Supervision Ablation

Table C.2 reports the contribution of the complementary SFT supervision views on DEV151.

Table C.2: Multi-view supervision ablation on DEV151.

Supervision	CLEAN	Detection	Exact FE	Strict Ver. Acc.
U only	54.17	86.61	20.47	25.83
$D + F + U$ (No-P)	58.33	88.98	26.77	31.79
$D + F + P + U$	58.33	91.34	28.35	33.11

C.3 Natural Deployment Alignment

Table C.3 reports the pre/post alignment comparison on DEV151 under the natural inference interface.

Table C.3: Effect of natural deployment alignment on DEV151.

Configuration	CLEAN	Detection	Exact FE	Strict Ver. Acc.
Pre-alignment	50.00	82.68	16.54	21.85
Post-alignment	54.17	92.13	25.98	30.46

Deployment alignment improves both trajectory-level behavior and exact first-error localization under the natural inference interface without introducing new teacher trajectories or mathematical supervision.

C.4 Process-Supervision Scaling

The Process-view scaling study separates the effect of increasing supervision diversity from the amount of training exposure assigned to that supervision. Although 466 trajectories are Process-eligible after constrained re-checking (Table A.2), the controlled scaling experiment uses a frozen 458-trajectory Process training pool as its reference condition. Table C.4 reports the corresponding PRM300 comparison.

Table C.4: Process-supervision scaling on PRM300.

Configuration	Unique P	P exposure	Strict Ver. Acc.
Initial frozen Process pool	458	1,099	33.67
Expanded Process pool	790	1,099	33.67
Exposure-matched Process pool	790	1,895	35.33

Increasing the number of unique Process trajectories from 458 to 790 while holding total Process exposure fixed at 1,099 does not improve PRM300 Strict Verification Accuracy. Matching Process exposure to the expanded pool instead raises Strict Verification Accuracy from 33.67% to 35.33%. The final 1,895-presentation configuration approximately restores the per-trajectory Process exposure density of the 458-trajectory reference pool while retaining the expanded set of 790 unique Process trajectories.

C.5 RL Localization and Component Ablation

We evaluate the contribution of pairwise first-error optimization and frozen-reference KL regularization on DEV151. Because all variants use the same frozen SFT decision gate, CLEAN Accuracy and Error Detection are identical across configurations; differences therefore arise entirely from conditional first-error localization.

Table C.5: RL localization and component ablation on DEV151. Strict Ver. Acc. denotes Strict Verification Accuracy.

Variant	CLEAN	Detection	Exact FE	FE|Det.	Strict Ver. Acc.
Pre-RL SFT localizer	58.33	91.34	28.35	31.03	33.11
RL w/o KL ($\beta = 0$)	58.33	91.34	33.86	37.07	37.75
Full RL ($\beta = 0.02$)	58.33	91.34	**35.43**	**38.79**	**39.07**

The pairwise RL update without KL substantially improves localization over the pre-RL localizer, increasing Exact FE from 28.35% to 33.86% and Strict Verification Accuracy from 33.11% to 37.75%. Adding reference KL further raises Exact FE to 35.43% and Strict Verification Accuracy to 39.07%, while leaving CLEAN Accuracy and Error Detection unchanged.

A paired analysis confirms that the improvement is primarily driven by the RL localization update. Relative to the pre-RL localizer, RL without KL produces 8 gains and 1 regression (net $+7$; McNemar $p = 0.0391$), while the full RL configuration produces 10 gains and 1 regression (net $+9$; McNemar $p = 0.0117$). The additional effect of KL is smaller: moving from RL without KL to full RL produces 2 gains and no regressions (net $+2$; McNemar $p = 0.5000$). Thus, pairwise hard-negative optimization provides the main localization improvement, while KL regularization contributes a smaller additional gain.

C.6 RL Optimizer Comparison on PRM300

We compare our frozen-gate pairwise localization objective with PPO and Bandit-OPD on three representative backbones. All methods preserve the same frozen SFT CLEAN/ERROR decision gate for each backbone; therefore, CLEAN Accuracy and Error Detection are invariant within each backbone and are omitted from the table. All values are percentages.

Table C.6: Comparison of RL optimization methods on PRM300 under the same frozen-gate setting. Ours denotes the pairwise contextual-bandit objective with frozen-reference KL regularization. CLEAN Accuracy and Error Detection are fixed by construction and therefore omitted.

Backbone	RL method	Exact FE	FE|Det.	Strict Ver. Acc.
Qwen2.5-Math-7B	Pairwise PG + KL (Ours)	**26.51**	**31.15**	**37.00**
	PPO	25.58	30.05	36.33
	Bandit-OPD	22.79	26.78	34.33
Qwen2.5-7B-Instruct	Pairwise PG + KL (Ours)	**31.63**	**37.16**	**37.67**
	PPO	30.70	36.07	37.00
	Bandit-OPD	26.98	31.69	34.33
Mathstral-7B	Pairwise PG + KL (Ours)	**32.09**	**36.51**	**35.67**
	PPO	31.16	35.45	35.00
	Bandit-OPD	27.44	31.22	32.33

D Qualitative Analysis and Failure Modes

The following representative cases illustrate three typical behaviors: Deep-Research teacher verification can correct erroneous attributions to earlier valid steps through source-grounded evidence; frozen-gate RL can shift localization from a downstream consequence to the authoritative first errory step while keeping the reasoning-trace-level CLEAN/ERROR decision unchanged; however, when a downstream error is easier to identify than its causal origin, first-error localization may still fail. These cases respectively illustrate evidence grounding, conditional localization optimization, and the remaining localization bottlenecks of the current method.

D.1 Teacher Correction through Source-Grounded Verification

Problem. Compute the sum of all digits used to write the integers from 1 through 110, inclusive.

The authoritative first error step is Step 14. The Native Teacher instead predicts Step 4, whereas the Deep-Research Teacher correctly predicts Step 14.

The earlier step identified by the Native Teacher is in fact valid:

$$10\text{–}19: \qquad 10 \times 1 + 45 = 55.$$

The substantive error occurs later when the candidate reasoning states:

$$11 \times 1 + (0 + 1 + \cdots + 10) = 11 + 55 = 66.$$

The tens and ones digits of 00 through 10 instead sum to

$$(0 + 1 + \cdots + 9) + (1 + 0) = 45 + 1 = 46,$$

so the digit sum for 100 through 110 is

$$11 + 46 = 57,$$

and the final total is 957 rather than 966.

A tool call associated with Step 14 formally checks the relation

$$66 = 66,$$

which is true but does not verify the substantive claim that the digit sum from 100 through 110 equals 66. The final causal audit records this as a source-proposition mismatch and does not use the tool result as supporting evidence for the final judgment. The Deep-Research Teacher nevertheless identifies Step 14 from the source-grounded causal analysis. This example illustrates why successful tool execution alone is insufficient: evidence must also correspond to the mathematical proposition actually under verification.

D.2 RL Correction of a Downstream Localization Error

Problem. Find the minimum value of

$$2\sqrt{x} + \frac{1}{x}, \qquad x > 0.$$

The authoritative first error step is Step 5. Before RL, the frozen SFT localizer predicts Step 6; after RL, the prediction moves to the correct Step 5.

The relevant reasoning is:

$$\text{Step 5:} \qquad \frac{2\sqrt{x} + 1/x}{2} \geq \sqrt{2\sqrt{x} \cdot \frac{1}{x}} = 1,$$

followed by

$$\text{Step 6:} \qquad 2\sqrt{x} + \frac{1}{x} \geq 2.$$

The first error already occurs in Step 5, because

$$\sqrt{2\sqrt{x} \cdot \frac{1}{x}} = \sqrt{\frac{2}{\sqrt{x}}},$$

which is not identically equal to 1. Step 6 is therefore a downstream consequence of the earlier incorrect equality.

Thus the RL update changes

$$\text{Step 6} \quad \longrightarrow \quad \text{Step 5},$$

without altering the trajectory-level ERROR decision. This case directly illustrates the intended role of frozen-gate RL: correcting the ordering of competing error locations toward the causal first error rather than changing whether the trajectory is classified as erroneous.

D.3 Remaining Failure: Downstream Error Preferred over Its Cause

Problem. Classify the graph of

$$x^2 + 2y^2 - 6x - 8y + 21 = 0.$$

The authoritative first error step is Step 3, while both the pre-RL and post-RL localizers predict Step 4.

The relevant reasoning begins by factoring the y terms as

$$2(y^2 - 4y).$$

Step 3 then states that completing the square requires adding and subtracting

$$(8/2)^2 = 16$$

inside the parentheses. This is already incorrect: after factoring out the coefficient 2, the expression inside the parentheses is $y^2 - 4y$, so the required square-completion term is

$$(-4/2)^2 = 4.$$

Step 4 subsequently writes

$$(x^2 - 6x + 9) + 2(y^2 - 4y + 16) + 21 - 9 - 32 = 0,$$

making the earlier mistake explicit in the algebraic expression. The localizer selects this downstream manifestation rather than Step 3, where the incorrect operation was first introduced.

This failure illustrates a remaining distinction between detecting an incorrect mathematical state and identifying its causal origin. Even after RL, explicit downstream inconsistencies can remain more salient than the earlier reasoning operation that generated them.

E LIMITATIONS

Limitations. Several limitations define the scope of the current study. First, the Deep-Research Teacher shows a directional improvement on the 200-example confirmatory evaluation, but the paired difference is not statistically significant under McNemar's test. We therefore do not treat this result as standalone evidence that the Teacher procedure universally improves verification; instead, the teacher-side experiment supports the intended verification behavior and complements the student-side results. This does not affect the main empirical conclusion that Deep-Research-grounded trajectory distillation improves Strict Verification Accuracy and Exact FE across 10 open-weight backbones.

Second, the current external evaluation is centered on PRM300 and the tested backbones occupy a limited model-scale range. The present experiments therefore do not establish cross-distribution or larger-scale generalization. However, the consistent Base→SFT improvement across 10 heterogeneous open-weight backbones still supports the narrower conclusion that the learned verification behavior is not specific to a single backbone or model family under the evaluated setting.

Third, frozen-gate RL does not improve every backbone. This limits the strength of any claim that RL provides universal cross-backbone gains, but it does not undermine the methodological role of the frozen-gate design: the CLEAN/ERROR decision remains fixed, so post-RL changes are attributable to conditional localization rather than altered upstream routing. Moreover, on the three representative backbones used for controlled optimizer comparison, Pairwise PG + KL achieves higher Exact FE, FE|Detected, and Strict Verification Accuracy than PPO and Bandit-OPD.

Finally, our efficiency analysis reports trajectory counts, supervision exposure, and optimization budgets rather than complete API and GPU costs. This restricts the scope of our efficiency claims, but does not affect the reported verification-performance comparisons, which are evaluated under the same frozen inference, parsing, and scoring protocols. Overall, these limitations constrain the breadth of the conclusions rather than the validity of the central findings supported by the current experiments.