BEYOND IMITATION: REFLECTIVE ON-POLICY SELF-DISTILLATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) improves the reasoning capabilities of large language models (LLMs) by providing dense token-level supervision for on-policy rollouts. However, existing OPSD methods often yield limited gains on complex reasoning tasks and suffer from severe training instability. We identify two key causes: conditioning the self-teacher on a complete verified solution encourages imitation of complete reference trajectories rather than extraction of transferable reasoning insights, while indiscriminate full-response distillation imposes superfluous supervision on already-valid reasoning prefixes. Together, these issues suppress reasoning diversity and contribute to late-stage mode collapse.

We propose Reflective On-policy Self-Distillation (ROSD), which distills transferable reasoning insights rather than complete reference trajectories. For each erroneous rollout, a self-reflector contrasts it with a correct rollout from the same group to derive a corrective idea and identify the sentence containing the first reasoning error. The corrective idea provides the self-teacher with targeted guidance, while the diagnosed error boundary allows ROSD to mask out the distillation loss over the valid prefix and apply token-level distillation only from the first erroneous sentence onward. Experiments across multiple reasoning benchmarks and model backbones show that ROSD consistently outperforms standard OPSD and reinforcement learning baselines, better preserves reasoning diversity, stabilizes training, and mitigates late-stage mode collapse. Code is available at `https://anonymous.4open.science/r/ROSD`.

1 INTRODUCTION

Post-training plays a key role in improving the reasoning ability of large language models (Achiam et al., 2023; Kumar et al., 2025). Among post-training approaches, reinforcement learning with verifiable rewards (RLVR) has emerged as an effective paradigm for optimizing models in verifiable domains (Shao et al., 2024; Yu et al., 2026; Team et al., 2025). However, typical RLVR algorithms such as GRPO (Guo et al., 2025; Shao et al., 2024) rely on outcome rewards to compute response-level advantages for model optimization. As a result, they provide no token-level supervision over intermediate reasoning steps, which limits their ability to further improve reasoning quality. OPSD addresses this limitation by using the model itself as a teacher, additionally conditioned on a verified correct solution (Hübotter et al., 2026; Zhao et al., 2026a; Shenfeld et al., 2026; He et al., 2026; Li et al., 2026a). It provides dense supervision by minimizing the token-wise KL divergence between the student and self-teacher distributions along on-policy rollouts.

Despite the promise of dense supervision, standard OPSD often exhibits training instability and suboptimal reasoning performance. As shown in Figure 1(a), a representative OPSD method, SDPO (Hübotter et al., 2026), undergoes pronounced performance degradation and mode collapse during the later stages of training. We hypothesize that this behavior stems from indiscriminate trajectory-wide imitation. By conditioning the self-teacher on a complete verified solution and applying distillation throughout the response, standard OPSD aligns the student with the solution-conditioned teacher even when the student's intermediate reasoning is already valid. Since many problems admit multiple valid reasoning paths, such alignment may penalize an alternative yet sound strategy simply because it differs from the reference solution. Over time, this can suppress reasoning diversity and bias the model toward a narrow set of solution patterns.

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.

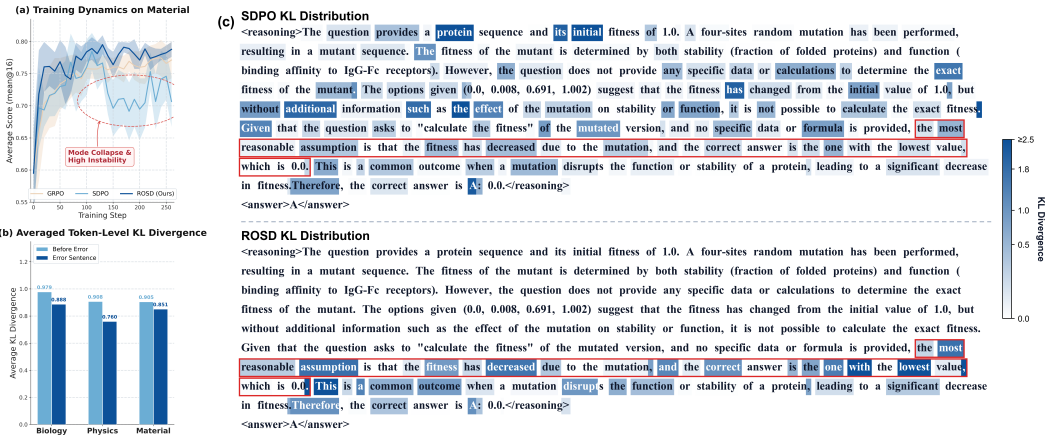


Figure 1: **(a)** Evaluation score dynamics during training on the Material benchmark. **(b)** Average per-token KL divergence induced by SDPO across three scientific reasoning benchmarks. The valid reasoning prefix (*Before Error*) consistently receives greater distillation pressure than the sentence containing the first reasoning error (*Error Sentence*). **(c)** Token-level KL divergence on a representative Biology rollout under SDPO (top) and ROSD (bottom). Darker colors indicate larger KL divergence, and the red rectangle marks the first diagnosed reasoning error. SDPO imposes substantial distillation pressure throughout the valid prefix while applying comparatively weak supervision to the error. In contrast, ROSD masks out the KL loss over the valid prefix and applies token-level distillation only from the diagnosed error onward. This example provides a token-level illustration of the aggregate pattern observed in **(b)** and demonstrates how ROSD preserves valid reasoning while focusing its corrective signal on the erroneous suffix.

To substantiate this hypothesis, we analyze the token-level KL divergence induced by SDPO on erroneous rollouts. For each rollout, we identify the valid prefix preceding the first logical error and the sentence containing that error. As shown in Figure 1(b), we compare the average per-token KL divergence over these two regions across three scientific reasoning benchmarks. Strikingly, the distillation pressure on the valid prefix is consistently greater than that on the first erroneous sentence. Since the prefix is already logically valid, such strong pressure does not directly contribute to correcting the identified error. Figure 1(c) illustrates this aggregate finding with a representative Biology rollout: SDPO imposes substantial KL penalties throughout the valid prefix while applying comparatively weak supervision to the first erroneous sentence. This case study illustrates how the solution-conditioned self-teacher can induce superfluous imitation of the reference trajectory, including its specific phrasing, rather than provide targeted supervision for the actual reasoning failure. By penalizing valid alternative derivations, such redundant supervision may suppress reasoning diversity and concentrate the policy on a narrow set of brittle solution patterns, thereby contributing to mode collapse. We provide the complete analysis protocol and case study, including the problem, rollout, and error diagnosis, in Appendices D and E.

Taken together, these findings suggest that self-distillation should focus on *transferable reasoning insights* rather than enforce full-trajectory imitation. Motivated by this principle, we propose Reflective On-policy Self-Distillation (ROSD), a framework that replaces direct imitation of complete correct solutions with reflection-guided, error-localized distillation. As illustrated in Figure 2(c), for each erroneous rollout, a self-reflector contrasts it with the shortest correct rollout from the same group to derive a corrective idea and identify the sentence containing the first reasoning error. The corrective idea conditions the self-teacher on what went wrong and how to repair it, providing targeted guidance without exposing the teacher directly to the complete correct trajectory. The diagnosed error boundary further allows ROSD to preserve the valid reasoning prefix by masking out its distillation loss and applying token-level distillation only from the first erroneous sentence onward. For a correct rollout, the self-reflector instead summarizes the key idea underlying its successful reasoning, which serves as the auxiliary context for self-teaching. In this way, ROSD distills concise, rollout-specific reasoning guidance while preserving valid alternative derivations, thereby stabilizing training and mitigating late-stage mode collapse.

Our primary contributions are summarized as follows:

- We empirically identify that standard OPSD's reliance on full-trajectory imitation and indiscriminate response-level distillation imposes superfluous supervision on valid reasoning prefixes, suppressing reasoning diversity and inducing training instability and late-stage mode collapse.
- We propose ROSD, a reflection-guided OPSD framework that distills transferable reasoning insights rather than complete reference trajectories, enabling targeted reasoning correction while preserving valid prefixes.
- Extensive experiments across multiple challenging reasoning benchmarks and model backbones show that ROSD consistently outperforms standard OPSD and RLVR baselines, better preserves reasoning diversity, stabilizes training, and avoids the observed late-stage mode collapse.

2 Preliminaries and Related Work

2.1 RL-Based Post-Training for LLMs

Post-training has been widely used to align large language models (LLMs) with human preferences. Representative approaches include reinforcement learning from human feedback (RLHF) (Lee et al., 2024; Ouyang et al., 2022) and direct preference optimization (DPO) (Rafailov et al., 2023), which optimizes a preference objective without explicit reinforcement learning. More recently, reinforcement learning with verifiable rewards (RLVR) has become an effective paradigm for improving LLM reasoning (Guo et al., 2025; Shao et al., 2024; Team et al., 2025; Dai et al., 2025; Meng et al., 2025). Group Relative Policy Optimization (GRPO) is a representative RLVR algorithm that estimates the advantage of each rollout relative to other responses sampled for the same problem, avoiding a separately trained critic (Shao et al., 2024; Guo et al., 2025). Subsequent studies improve GRPO-style training through modifications to the objective, advantage normalization, clipping, and sampling (Yu et al., 2025; Liu et al., 2025; Yue et al., 2025; Zheng et al., 2025; Chu et al., 2025; Zhao et al., 2026b).

Given a problem x, the policy samples a group of on-policy rollouts $\{y_i\}_{i=1}^{G}$ with rewards $\{r_i\}_{i=1}^{G}$. GRPO assigns each rollout the group-relative advantage

$$A_i = \frac{r_i - \mu_G}{\sigma_G}, \tag{1}$$

where μ_G and σ_G are the mean and standard deviation of group rewards. Although the policy objective is optimized over individual tokens, every token in rollout y_i shares the same response-level advantage A_i. GRPO therefore provides no explicit token-level credit assignment for distinguishing valid reasoning steps from erroneous ones. In particular, an incorrect rollout may receive a negative advantage over its entire trajectory even when its prefix contains valid reasoning.

2.2 On-Policy Self-Distillation

Classical knowledge distillation trains a student model to imitate outputs generated by a stronger teacher (Hinton et al., 2015; Gu et al., 2024; Kim & Rush, 2016). For LLM reasoning, such supervision is typically off-policy because the targets are generated by the teacher rather than sampled from the student's current policy. In contrast, on-policy self-distillation samples rollouts from the current student and uses a self-teacher conditioned on additional information to provide token-level supervision along these on-policy trajectories (Agarwal et al., 2024; Lu & Lab, 2025). OPSD (Zhao et al., 2026a) and SDFT (Shenfeld et al., 2026) condition the self-teacher on reference solutions, while SDPO (Hübotter et al., 2026) uses environmental feedback or successful rollouts to guide unsuccessful trajectories.

At token position t, let

$$p^{\mathrm{S}}_{i,t} = \pi_\theta(\cdot \mid x, y_{i,<t}), \qquad p^{\mathrm{T}}_{i,t} = \pi_\theta(\cdot \mid x, c_i, y_{i,<t}), \tag{2}$$

denote the student and self-teacher distributions, respectively, where c_i is the additional context. SDPO minimizes a token-level distillation objective over the entire rollout:

$$\mathcal{L}_{\mathrm{SDPO}}(\theta) = \sum_{i=1}^{G} \sum_{t=1}^{|y_i|} D\left(p^{\mathrm{S}}_{i,t}, \mathrm{stopgrad}\left[p^{\mathrm{T}}_{i,t}\right]\right), \tag{3}$$

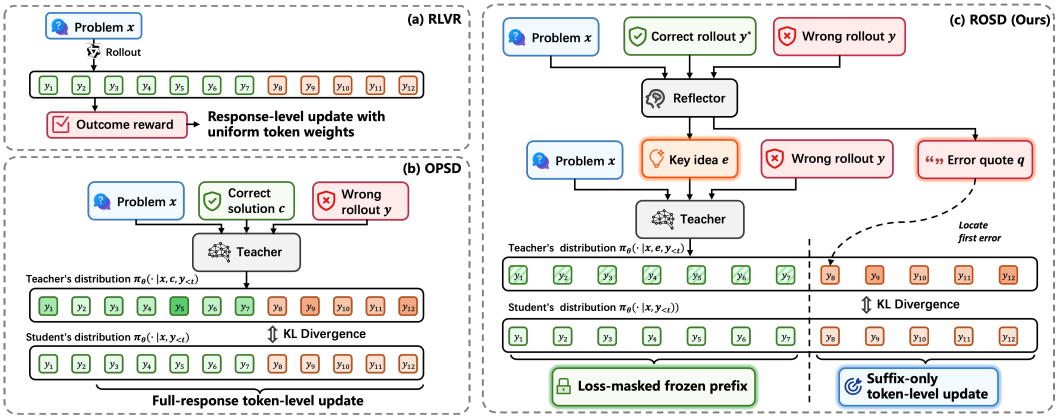


Figure 2: Conceptual comparison of the learning signals in RLVR, standard OPSD, and ROSD, illustrated using an erroneous rollout. **(a)** RLVR provides only an outcome-level reward for the complete rollout. **(b)** Standard OPSD conditions the self-teacher on a complete correct solution and distills the entire response, including its already-valid prefix. **(c)** ROSD uses a self-reflector to derive a corrective idea and identify the sentence containing the first reasoning error. The corrective idea guides the self-teacher, while the diagnosed error boundary masks out the valid prefix and restricts token-level distillation to the suffix beginning with the first erroneous sentence. Green and orange tokens denote the valid prefix and error-affected suffix, respectively; hatched tokens indicate positions masked out from the distillation loss.

where D may be KL divergence, reverse KL divergence, or Jensen–Shannon divergence. Across existing methods, the self-teacher may share parameters with the student, remain fixed, or be updated as an exponential moving average (Li et al., 2026b).

Compared with GRPO, SDPO provides a teacher-derived distribution at every token position rather than sharing a response-level advantage across the rollout. However, standard OPSD applies this dense supervision to the entire response. Conditioning the self-teacher on a complete correct solution may encourage direct trajectory imitation, while full-response distillation may impose unnecessary gradients on already-valid prefixes, suppressing reasoning diversity and destabilizing training.

ROSD addresses both limitations through reflection-guided, error-localized distillation. It extracts concise, rollout-specific guidance—a corrective idea for an incorrect rollout or a key successful insight for a correct one—rather than directly exposing the self-teacher to a complete correct trajectory. For an incorrect rollout, it identifies the sentence containing the first reasoning error, masks the preceding valid prefix, and applies distillation only to the suffix beginning with that sentence. Thus, ROSD retains the dense supervision of OPSD while explicitly localizing where correction should begin.

3 Proposed Method: ROSD

ROSD shifts on-policy self-distillation from rigid trajectory imitation to distilling transferable reasoning insights for targeted reasoning correction. As illustrated in Figure 2, it differs from both the outcome-level supervision of RLVR and the full-response distillation of standard OPSD. Given an input reasoning task, the policy first generates candidate on-policy rollouts. For each erroneous rollout, a self-reflector derives a corrective idea from a verified solution and identifies the first incorrect reasoning step (Section 3.1). The corrective idea guides the self-teacher toward targeted supervision, while the diagnosed error boundary enables ROSD to mask out the distillation loss over the valid prefix and apply token-level distillation only from the first error onward (Section 3.2). ROSD uses the same base model for student model, reflection and self-teaching, requiring no auxiliary models or additional trainable parameters.

3.1 Outcome-Aware Self-Reflection

In standard SDPO, the auxiliary context c can be a complete solution. Although this provides the self-teacher with strong information about a correct solution, it does not reveal where or why the

student's on-policy rollout fails. Consequently, the self-teacher can be overly shaped by the reference trajectory, including its reasoning path, format, and stylistic details. In contrast, ROSD does not directly expose the self-teacher to the complete reference solution. Instead, it introduces an outcome-aware self-reflector that extracts rollout-specific reasoning insights: it diagnoses how an incorrect rollout should be corrected while summarizing the key idea underlying a correct rollout.

For each problem x, the current policy samples G on-policy rollouts:

$$\mathcal{Y}(x) = \{y_1, \ldots, y_G\}. \tag{4}$$

Based on final-answer correctness, we partition them into correct rollouts $\mathcal{Y}^+(x)$ and incorrect rollouts $\mathcal{Y}^-(x)$. Consistent with group-based reinforcement learning, we retain only mixed-outcome groups containing both correct and incorrect rollouts (Shao et al., 2024; Yu et al., 2025). ROSD then constructs different self-reflection prompts for incorrect and correct rollouts.

Contrastive reflection for incorrect rollouts. For each incorrect rollout $y^- \in \mathcal{Y}^-(x)$, we select the shortest correct rollout from the same group:

$$y^* = \arg\min_{y \in \mathcal{Y}^+(x)} |y|, \tag{5}$$

and use it as a contrastive basis for diagnosing where the incorrect reasoning deviates. Selecting the shortest correct rollout encourages the reflection to identify a concise and efficient corrective direction. Given the problem x, the correct rollout y^*, and the incorrect rollout y^-, the self-reflector produces a corrective idea e^- and an error quote q:

$$(e^-, q) \sim \pi_\theta(\cdot \mid x, y^*, y^-). \tag{6}$$

The corrective idea e^- explains why the rollout fails and how its reasoning can be repaired. The error quote q is the exact sentence from y^- that contains the first identified reasoning error. It serves only as a localization anchor and is later aligned with the student rollout to determine where localized distillation begins.

Positive reflection for correct rollouts. For each correct rollout $y^+ \in \mathcal{Y}^+(x)$, no error localization is required. Instead of directly using the complete rollout as teacher context, the self-reflector summarizes the key idea underlying its successful reasoning:

$$e^+ \sim \pi_\theta(\cdot \mid x, y^+). \tag{7}$$

This positive reflection retains the transferable reasoning insight from a successful rollout without encouraging direct imitation of its complete trajectory.

Key insight. The extracted insight, either e^- or e^+, is subsequently used as the auxiliary context for the self-teacher. Self-reflection acts as an information bottleneck that converts complete solutions into concise, rollout-specific reasoning guidance. For incorrect rollouts, it further separates *what to correct*, represented by e^-, from *where to begin correction*, identified by q. This separation enables targeted teacher guidance while preserving reasoning that is already valid.

The self-reflector shares the same model weights as the policy and self-teacher, requiring no auxiliary model or additional trainable parameters. The complete self-reflection prompt templates are provided in Appendix C (Figure 5).

3.2 Error-Localized Self-Distillation

Even with rollout-specific reflection, applying self-distillation to the entire response remains unnecessary. Because the self-teacher is conditioned on additional reasoning guidance, its distribution may differ from the student's natural generation even over valid reasoning steps. Full-response distillation may therefore penalize valid alternative prefixes, suppress reasoning diversity, and increase the risk of mode collapse. For an incorrect rollout, we instead use the error quote q to preserve the valid prefix and restrict distillation to the suffix beginning with the first erroneous sentence.

Specifically, for each incorrect rollout $y^- \in \mathcal{Y}^-(x)$, ROSD matches the error quote q against the original rollout and identifies the token index at which the quoted sentence begins:

$$k = \mathrm{Locate}(q, y^-). \tag{8}$$

If the quote cannot be matched, we fall back to full-response distillation by setting $k = 1$. We then define the token-level mask

$$m_t = \begin{cases} 0, & t < k, \\ 1, & t \geq k. \end{cases} \tag{9}$$

Thus, tokens in the valid prefix do not contribute to the distillation loss, whereas all tokens from the first erroneous sentence onward are retained. For a correct rollout, no error localization is needed, and we set $m_t = 1$ for all response tokens.

Let e denote the reflection associated with the rollout, i.e., $e = e^-$ for an incorrect rollout and $e = e^+$ for a correct rollout. The reflection conditions the self-teacher on targeted reasoning guidance. At each response position t, the student and self-teacher distributions are respectively

$$p_t^{\mathrm{S}} = \pi_\theta(\cdot \mid x, y_{<t}), \qquad p_t^{\mathrm{T}} = \pi_\theta(\cdot \mid x, e, y_{<t}). \tag{10}$$

Both distributions are evaluated over the same rollout prefix $y_{<t}$, while gradients through the self-teacher are stopped. Following prior work (Hübotter et al., 2026; Li et al., 2026a), we instantiate the token-level divergence using Jensen–Shannon divergence (JSD):

$$\mathcal{L}_{\mathrm{ROSD}}(\theta) = \sum_{t=1}^{T} m_t \, \mathrm{JSD}\left(p_t^{\mathrm{S}} \,\middle\|\, \mathrm{stopgrad}\left[p_t^{\mathrm{T}}\right]\right). \tag{11}$$

The mask m_t ensures that, for an incorrect rollout, distillation begins only at the first diagnosed error, thereby preserving the valid reasoning prefix while correcting the subsequent reasoning trajectory. The student and self-teacher prompt templates are provided in Appendix J.

4 Experiments

4.1 Experimental Setup

Models. We evaluate our method across two diverse and representative model families: Qwen3-4B, Qwen3-8B (Yang et al., 2025), and Llama-3.2-3B-Instruct (Grattafiori et al., 2024). For ROSD and OPSD baselines, the student, self-teacher, and self-reflector share the same base model. Thus, the performance gains come from training with a self-teacher conditioned on privileged information, rather than from using a larger external teacher.

Datasets. Following prior work (Hübotter et al., 2026), we train models separately on five datasets. The first four datasets are science question-answering benchmarks from the reasoning subset (L3) of SciKnowEval (Feng et al., 2024), covering undergraduate-level scientific reasoning in Chemistry, Physics, Biology, and Materials Science. In addition, we include a tool-use dataset from ToolAlpaca (Tang et al., 2023), where the model is required to map a tool-API specification and a user request to the correct tool call. Each of the five datasets is split into a training set and a test set. Models are trained on each domain and evaluated on its corresponding test set. The statistics of each dataset are shown in Appendix A.

Baselines. We compare ROSD with two baselines in the standard RLVR setting, where no human-labeled golden solutions are available and models are trained only with feedback from a verifier. (1) For the GRPO baseline, we use a strengthened implementation of GRPO following recent best practices (Khatri et al., 2025), representing outcome-level reinforcement learning. This implementation incorporates recent refinements to standard GRPO, including asymmetric clipping (Yu et al., 2026), unbiased normalization (Liu et al., 2025), and off-policy correction for efficient inference-based rollout generation (Yao et al., 2025). (2) For the on-policy self-distillation baseline, we use SDPO (Hübotter et al., 2026), adopting its official implementation and hyperparameter configuration for the verifier-only RLVR setting. SDPO uses successful rollouts sampled by the current policy as additional self-teacher context, thereby providing denser token-level guidance. We extend training to the epoch budgets specified in Appendix B to examine longer-term training dynamics. We do not include other OPSD methods, such as OPSD (Zhao et al., 2026a) and SDFT (Shenfeld et al., 2026), in the main comparison because they rely on high-quality human-labeled golden solutions.

Model	Method	Material	Physics	Biology	Chemistry	ToolUse	Average
Qwen3-4B	Base	62.90	60.63	31.63	42.92	58.55	51.32
	GRPO	**81.45**	74.61	53.75	77.53	61.95	69.86
	SDPO	73.60	68.52	55.38	76.93	60.66	67.02
	ROSD	80.18	**76.56**	**57.50**	**82.47**	**67.46**	**72.83**
Qwen3-8B	Base	59.71	59.06	32.75	41.28	57.72	50.10
	GRPO	77.46	76.09	56.88	80.77	68.75	71.99
	SDPO	78.46	**76.41**	**61.13**	80.71	65.81	72.50
	ROSD	**79.52**	75.16	60.38	**83.07**	**69.12**	**73.45**
Llama-3.2-3B-Instruct	Base	0.01	0.01	0.00	0.01	0.00	0.01
	GRPO	46.14	**48.04**	45.37	65.56	**58.36**	52.69
	SDPO	45.47	46.01	42.25	61.78	55.97	50.30
	ROSD	**46.76**	47.57	**49.12**	**66.36**	57.07	**53.38**

Table 1: Reasoning performance across scientific and tool-use benchmarks on Qwen3-4B, Qwen3-8B, and Llama-3.2-3B-Instruct. "Base" denotes the untuned base model. All methods are trained to convergence, and scores are reported as the mean@16(%). **Boldface** indicates the best result, and underline indicates the second-best result.

Implementation details. All methods are trained with the same data splits, answer verifiers, and evaluation prompts. For each prompt, we sample 8 on-policy rollouts during training. Following prior work (Hübotter et al., 2026; Li et al., 2026b), evaluation uses 16 sampled responses per problem and reports the mean accuracy, denoted as mean@16. Additional training hyperparameters and distillation settings are provided in Appendix B.

4.2 Main Results

Table 1 reports the results across five tasks using Qwen3-4B, Qwen3-8B, and Llama-3.2-3B-Instruct. We highlight two main findings.

ROSD performs consistently across model architectures. ROSD achieves the best average performance on all three backbones and consistently outperforms both GRPO and SDPO across model families and scales. In contrast, the relative performance of SDPO depends more strongly on model capacity: it slightly outperforms GRPO on Qwen3-8B but falls behind GRPO on the smaller Qwen3-4B and Llama-3.2-3B models. These results suggest that dense token-level supervision alone does not guarantee improvements over outcome-level RL, particularly when model capacity is limited. ROSD instead delivers more consistent gains across the evaluated backbones.

ROSD performs well across diverse task domains. The two self-distillation methods also differ substantially across individual tasks. ROSD achieves or approaches the best performance on both scientific question answering and tool use, whereas SDPO exhibits pronounced degradation in several settings, including Physics and Materials Science on Qwen3-4B. This discrepancy is consistent with the limitations of full-response distillation, which may impose unnecessary supervision on already-valid reasoning prefixes. By extracting rollout-specific guidance and applying distillation only to the suffix beginning with the first erroneous sentence, ROSD preserves valid prefixes while correcting error-affected continuations.

4.3 Training Dynamics and Stability

To examine the optimization process, we analyze Qwen3-4B and Qwen3-8B on Chemistry and Materials Science. We report test-set mean@16 in Figure 3 and on-policy rollout accuracy in Figure 4. The results reveal two main patterns.

Dense self-distillation accelerates early learning. During early training, both SDPO and ROSD improve rollout accuracy considerably faster than GRPO. GRPO assigns each rollout a shared advantage derived from its outcome, providing no explicit token-level credit assignment. In contrast, self-distillation provides teacher-derived supervision at every token position, enabling faster initial optimization.



Figure 3: Test-set performance during training on Chemistry and Materials Science using Qwen3-4B and Qwen3-8B. We report mean@16, with shaded regions indicating standard deviation across sampled responses.

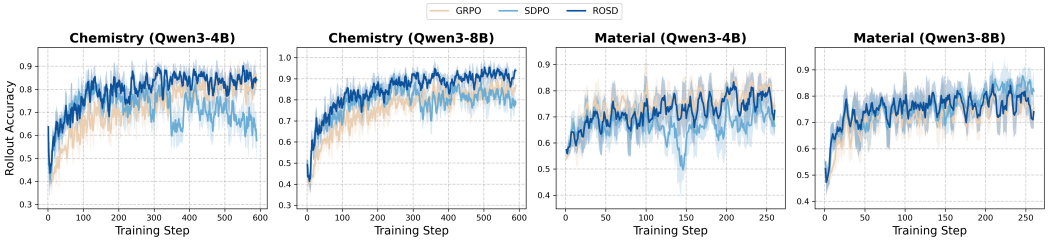


Figure 4: On-policy rollout accuracy during training on Chemistry and Materials Science using Qwen3-4B and Qwen3-8B. Curves show rolling averages over a five-step window, with shaded regions indicating standard deviation.

Method	Chemistry		Material		Physics		Biology		ToolUse		Average	
	SB-4 ↓	Dist-2 ↑	SB-4 ↓	Dist-2 ↑	SB-4 ↓	Dist-2 ↑	SB-4 ↓	Dist-2 ↑	SB-4 ↓	Dist-2 ↑	SB-4 ↓	Dist-2 ↑
Base	80.01	20.24	95.09	8.75	90.19	10.08	80.79	19.99	97.28	9.37	88.67	13.69
GRPO	**87.69**	**15.02**	90.97	12.90	**90.68**	11.89	**93.70**	**8.82**	95.37	11.52	**91.68**	12.03
SDPO	92.99	12.30	**89.55**	12.69	94.13	11.00	99.20	6.13	92.56	12.65	93.69	10.95
ROSD (Ours)	90.87	14.69	91.17	**15.10**	91.82	**13.78**	98.61	6.84	**90.48**	**14.14**	92.59	**12.91**

Table 2: Reasoning diversity analysis on Qwen3-4B at training step 140 across five domains. We report Self-BLEU-4 (SB-4, %, lower indicates higher diversity) and Distinct-2 (Dist-2, %, higher indicates higher lexical richness) on generated rollouts. **Boldface** indicates the best result, and underline indicates the second-best result among post-training methods (excluding Base).

ROSD remains stable during extended training. Despite its rapid initial improvement, SDPO exhibits substantial degradation later in training. On Qwen3-4B, its rollout accuracy and test performance collapse on both Chemistry and Materials Science. On Materials Science with Qwen3-8B, its rollout accuracy remains high while its test score steadily declines, indicating a widening generalization gap consistent with overfitting. In contrast, ROSD maintains stable or improving performance across the evaluated settings. These results suggest that masking valid prefixes and restricting distillation to the error-affected suffix mitigates the instability associated with full-response distillation.

4.4 Response Diversity Analysis

We further examine whether full-response self-distillation reduces response diversity. Specifically, we evaluate rollouts generated by Qwen3-4B at the final training checkpoint across all five tasks. Following prior work, we use two complementary metrics: **Self-BLEU-4** (Zhu et al., 2018), which measures similarity among responses generated for the same prompt (lower is more diverse), and **Distinct-2** (Li et al., 2016), which measures the proportion of unique bigrams among all generated bigrams (higher is more diverse).

ROSD better preserves response diversity. As shown in Table 2, SDPO generally produces higher Self-BLEU-4 and lower Distinct-2 than GRPO, with particularly low diversity on Biology. This pattern indicates that full-response distillation can make generated trajectories more homogeneous. In contrast, ROSD achieves higher Distinct-2 across all five tasks and generally lower Self-BLEU-4

Model	Method	Material	Physics	Biology	Chemistry	ToolUse	Average
Qwen3-4B	SDPO	73.60	68.52	55.38	76.93	60.66	67.02
	ROSD	**80.18**	**76.56**	**57.50**	**82.47**	**67.46**	**72.83**
	⊢ w/o Refl.	75.27	73.13	51.13	78.84	59.47	67.56
	⊢ w/o LocDist.	75.13	75.39	54.50	80.12	63.51	69.73
Qwen3-8B	SDPO	78.46	**76.41**	**61.13**	80.71	65.81	72.50
	ROSD	79.52	75.16	60.38	83.07	**69.12**	**73.45**
	⊢ w/o Refl.	73.80	71.41	59.13	**83.10**	65.07	70.50
	⊢ w/o LocDist.	**79.99**	76.25	56.50	81.76	67.19	72.34
Llama-3.2-3B-Instruct	SDPO	45.47	46.01	42.25	61.78	55.97	50.30
	ROSD	**46.76**	**47.57**	**49.12**	**66.36**	**57.07**	**53.38**
	⊢ w/o Refl.	45.23	46.54	45.65	63.24	54.47	51.03
	⊢ w/o LocDist.	46.02	45.71	47.84	65.92	56.31	52.26

Table 3: "w/o Refl." denotes without reflection guidance; "w/o LocDist." denotes without localized distillation. **Boldface** indicates the best result, and underline indicates second-best.

than SDPO. These results suggest that masking valid prefixes reduces unnecessary imitation and better preserves variation among generated responses.

4.5 Ablation Study

To dissect the contribution of each core component in ROSD, we evaluate two ablation variants across all five reasoning benchmarks on the three backbones (Table 3): (1) **w/o Reflection**, which retains error localization to identify the logical inflection point but conditions the self-teacher directly on the raw reference solution rather than generating targeted self-reflective diagnosis; and (2) **w/o Localized Distillation**, which preserves reflective guidance but applies distillation loss indiscriminately across the entire response trajectory, omitting prefix-preserving selective supervision.

As shown in Table 3, removing reflective guidance leads to a substantial performance drop across all three backbones, with average accuracy falling close to that of standard SDPO. This indicates that raw reference imitation provides diffuse and uninformative supervision, whereas targeted self-reflection explicitly diagnoses logical flaws to deliver sharp, semantically focused corrective signals. Furthermore, even with reflective teacher guidance, omitting localized distillation consistently underperforms full ROSD across all architectures. Forcing the model to distill already valid prefixes introduces redundant imitation penalties that stifle exploration and induce mode collapse, whereas confining supervision strictly to the erroneous suffix leaves sound prefixes unperturbed, stabilizing optimization and maximizing reasoning gains. Together, these findings demonstrate that reflective diagnosis and localized prefix preservation are complementary and jointly drive the superior performance of ROSD.

5 Discussion and Conclusion

Discussion. Our results reveal a central trade-off in reasoning post-training: outcome-level RLVR provides coarse feedback, whereas standard on-policy self-distillation, as represented by SDPO, offers dense supervision but may unnecessarily alter valid reasoning prefixes. The effectiveness of ROSD, however, depends on the quality of self-reflection and error localization, which may be less reliable for ambiguous or highly specialized problems. Moreover, sentence-level localization is a practical approximation to exact token-level error attribution. We provide further analyses of cross-domain generalization (Appendix F), training efficiency (Appendix G), response length dynamics (Appendix H), error localization dynamics (Appendix I), and qualitative examples (Appendix E).

Conclusion. We introduced ROSD, a reflective on-policy self-distillation framework for improving LLM reasoning. ROSD extracts rollout-specific guidance through self-reflection and applies distillation only to the suffix beginning with the first erroneous sentence, thereby correcting error-affected continuations while preserving valid prefixes. Across five tasks, ROSD achieves the best overall average performance across the three backbones, outperforming both GRPO and SDPO. Compared with SDPO, ROSD better preserves response diversity and improves training stability. Additional results demonstrate stronger cross-domain generalization than SDPO. Overall, our findings highlight the importance of combining dense token-level guidance with error-aware localization for reliable and generalizable post-training.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and assisting with translation. We also used an LLM as a judge to evaluate the model rollouts, as detailed in Appendix D. We have not used generative AI tools for proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, or interpreting results. Generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for finding related work. We have reviewed all AI-assisted work: all AI-assisted code implementations were manually inspected, debugged, and verified through unit and integration testing by the authors, and all suggested citations were independently read and confirmed. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies post-training methods for improving the reasoning capabilities of large language models. It does not involve human participants, crowdsourcing, or the collection of personal or sensitive information. All experiments use publicly available benchmark datasets, which are used in accordance with their licenses and intended research purposes. As with other methods that improve reasoning and tool-use capabilities, ROSD may strengthen both beneficial and potentially harmful downstream capabilities. Our experiments are limited to benchmark reasoning and tool-use tasks and should not be interpreted as establishing the safety, fairness, or reliability of the resulting models in real-world deployment. The trained models may also inherit biases and other limitations from their pretrained backbones and benchmark data. Deployment should therefore be accompanied by application-specific safety evaluation, appropriate access controls, and continued monitoring. We adhere to the ICLR Code of Ethics and acknowledge these potential limitations and risks.

REPRODUCIBILITY STATEMENT

We provide the complete formulation and training procedure of ROSD in Section 3. Section 4.1 describes the model backbones, datasets, baselines, evaluation metrics, and the shared training and evaluation protocol used for all compared methods. Appendix A reports the dataset composition and train–test splits, while Appendix B specifies the computing infrastructure, training budgets, batch size, sequence-length limits, evaluation frequency, and distillation hyperparameters. The self-reflection and teacher prompt templates are provided in Appendix C. An anonymous implementation is available at `https://anonymous.4open.science/r/ROSD`.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

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

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for llms. *arXiv preprint arXiv:2510.13786*, 2025.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning*, pp. 26874–26901. PMLR, 2024.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 110–119, 2016.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning, 2025. URL https://arxiv.org/abs/2503.07365.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

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. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, august 2025. *URL https://fengyao. notion. site/off-policy-rl*, 2025.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026a. URL https://arxiv.org/abs/2601.18734.

Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, et al. Reinforced efficient reasoning via semantically diverse exploration. *arXiv preprint arXiv:2601.05053*, 2026b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Dataset Statistics

Table 4 reports the number of training and held-out examples used in our experiments. The science question-answering and ToolUse domains are used for training and evaluation. The AIME2024 dataset is used only as an additional held-out reasoning benchmark and is not used for training.

Datasets	Train	Test
Biology	450	50
Chemistry	1,890	210
Material	841	94
Physics	720	80
ToolUse	4,046	68
AIME2024	-	30

Table 4: Dataset statistics used in our experiments.

B Implementation Details

Infrastructure. All experiments are implemented on top of the `verl` library (Sheng et al., 2025). We use the distributed actor-rollout training pipeline provided by `verl`, with FSDP actor training and vLLM-based rollout generation. The experiments are run on 8 NVIDIA A800 80G GPUs.

Training. Unlike prior methods (Hübotter et al., 2026) that train under a fixed time budget, we use an epoch budget large enough for the methods to converge in our setting. We train for 10 epochs on the science question-answering tasks and 5 epochs on the ToolUse task, using a training batch size of 32. The maximum length of the reflection prompt is 8k tokens, and the maximum length of the generated reflection is 4k tokens. All algorithms are evaluated every 10 training steps. Both SDPO and ROSD use Jensen–Shannon divergence with $\alpha = 0.5$ and top-k distillation with $k = 100$. Following the original implementation and hyperparameter configuration, SDPO updates its self-teacher via an exponential moving average (EMA) with an update rate of 0.05 at every training step. In ROSD, the self-teacher and self-reflector remain frozen throughout each training run.

C Self-Reflection Prompt Templates

Figure 5 presents the self-reflection prompt templates used by ROSD. The system prompt provides task-level instructions to the reflector. For wrong rollouts, the reflector receives the problem, the shortest correct rollout, and the wrong rollout to identify the first erroneous span and provide a corrective explanation. For correct rollouts, the reflector summarizes why the reasoning path is valid and what key idea leads to the correct answer.

D Token-Level KL Divergence: Valid Prefixes vs. Error Sentences

In Section 1 and Figure 1(b), we report that standard On-Policy Self-Distillation (SDPO) assigns unexpectedly large KL divergence supervision to valid reasoning prefixes—frequently exceeding the supervision exerted on the actual error sentence itself. Here, we present the comprehensive empirical methodology and benchmark-wide statistics.

To investigate how self-distillation distributes supervision across sound and erroneous reasoning steps, we sample 50 questions from each of three scientific reasoning domains in SciKnowEval (Feng et al., 2024) (*Biology*, *Physics*, and *Materials*) and generate reasoning rollouts using the base Qwen3-4B model at training Step 0. For each incorrect rollout paired with a verified reference solution, we employ `claude-opus-5` as an impartial adjudicator to identify the exact first erroneous sentence (`<error_quote>`) and explain the underlying reasoning flaw (see Figure 6 for the prompt template). Using the extracted error span, we partition each trajectory into the valid prefix prior to the error and the subsequent error sentence, and compute the token-level KL divergence along the rollout.

System prompt for both rollouts.
You are a careful tutor. Respond strictly in the required format and nothing else.

User prompt for wrong rollouts.
[Problem]
{problem}
[Correct Solution]
{correct_rollout}
[Incorrect Solution]
{wrong_rollout}
The correct final answer is {answer}. Diagnose the [Incorrect Solution] by responding strictly in the following format:
<error_quote>
(Extract the EXACT substring from the [Incorrect Solution] where the reasoning first goes wrong)
</error_quote>
<explanation>
(Explain the exact mistake in the quote, how to fix it, and what the correct logic is)
</explanation>

User prompt for correct rollouts.
[Problem]
{problem}
[Correct Solution]
{correct_rollout}
Explain why this reasoning is valid and why the final answer should be {answer}. Respond strictly in the following format:
<explanation>
(Your explanation of why the logic is correct)
</explanation>

Figure 5: Self-reflection prompt templates used by ROSD. The system prompt provides task-level instructions to the reflector. In the prompt templates, {problem} is replaced with the problem x; {correct_rollout} is replaced with the selected correct rollout, i.e., y^* in Equation 6 and y^+ in Equation 7; {wrong_rollout} is replaced with the wrong rollout y^-; and {answer} is replaced with the correct final answer. The outputs enclosed by <error_quote> and <explanation> correspond to the error quote q and the key corrective idea e, respectively.

As summarized in Figure 1(b), across all three domains, standard SDPO exhibits counter-intuitive supervisory behavior: the average token KL on valid prefixes is 0.9786 on Biology, 0.9080 on Physics, and 0.9054 on Materials, which consistently exceeds the average KL on the actual error sentence (0.8881 on Biology, 0.7605 on Physics, and 0.8513 on Materials). Because tokens preceding the error are already scientifically sound, this strong pull toward the reference teacher merely penalizes harmless stylistic deviations and legitimate alternative derivations rather than rectifying flaws, directly evidencing that standard OPSD causes premature mode collapse by over-supervising valid reasoning prefixes.

E Case Study: Protein Mutation Fitness Prediction

To supplement the qualitative case study presented in Figure 1(c), this section provides the complete verbatim problem statement, student rollout text, and self-reflector diagnostic output for the representative diagnostic example from the Biology domain.

Problem Description. The student policy is queried to predict the functional fitness score of an experimentally mutated protein sequence (a four-site random mutant of the streptococcal protein G B1 domain):

> **Problem:** *"Given the wide-type protein sequence 'MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGVDGEWTYDDATKTFTVTE' with an initial fitness of 1.0, we have performed a four-sites random mutation resulting in the mutant 'MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGMRKEWTYDDATKTFTGTE'. Here the fitness is determined by both stability (i.e. the fraction of folded proteins) and function (i.e. binding affinity to IgG-Fc receptors). Can you calculate the fitness of this mutated version?"*

[System]
You are an expert academic evaluator analyzing reasoning chains of language models. Your task is to pinpoint the exact location where an incorrect reasoning rollout first makes an error. You must respond strictly in the requested XML format and nothing else.

[User]
We have a scientific reasoning problem, the ground-truth answer, a verified correct reference solution, and an incorrect student rollout.

Please analyze the incorrect rollout and identify the EXACT first sentence or clause where the reasoning goes wrong.
Respond strictly in the following format:
<error_quote>
(Extract the EXACT substring from the [Incorrect Solution] where the reasoning first commits a factual, conceptual, or deductive error. Do not quote correct setup sentences preceding the mistake.)
</error_quote>
<explanation>
(Explain concisely what the error is and why it deviates from the correct scientific logic.)
</explanation>

[Problem]
{problem_text}

[Ground-Truth Answer]
{ground_truth}

[Correct Reference Solution]
{reference_correct_solution}

[Incorrect Student Solution]
{student_wrong_response}

Figure 6: Prompt template provided to `claude-opus-5` for extracting the exact first error quote and diagnostic explanation.

Options: *A: 0.0* *B: 0.008* *C: 0.691* *D: 1.002*
Ground Truth: B (*0.008*).

Student Reasoning Rollout. The sampled on-policy rollout from Qwen3-4B unfolds across three segments:

1. *Valid Prefix:* The model correctly recognizes the problem setup and biological background:

   "The question provides a protein sequence and its initial fitness of 1.0. A four-sites random mutation has been performed, resulting in a mutant sequence. The fitness of the mutant is determined by both stability (fraction of folded proteins) and function (binding affinity to IgG-Fc receptors). However, the question does not provide any specific data or calculations to determine the exact fitness of the mutant. The options given (0.0, 0.008, 0.691, 1.002) suggest that the fitness has changed from the initial value of 1.0, but without additional information such as the effect of the mutation on stability or function, it is not possible to calculate the exact fitness. Given that the question asks to "calculate the fitness" of the mutated version, and no specific data or formula is provided,"

2. *Error Sentence (highlighted by the red rectangle in Figure 1(c)):* The model introduces an unfounded heuristic jump, falsely equating a fitness reduction with total loss of function:

   "the most reasonable assumption is that the fitness has decreased due to the mutation, and the correct answer is the one with the lowest value, which is 0.0."

3. *Post-Error Suffix:* The model rationalizes this arbitrary assumption and concludes with the incorrect option **A**:

   "This is a common outcome when a mutation disrupts the function or stability of a protein, leading to a significant decrease in fitness. Therefore, the correct answer is A: 0.0."

Self-Reflector Diagnosis. Conditioned on the problem and the verified correct answer, our self-reflector successfully pinpoints the reasoning failure and produces the following structured reflection:

<error_quote>
the most reasonable assumption is that the fitness has decreased due to the mutation, and the correct answer is the one with the lowest value, which is 0.0.
</error_quote>
<explanation>
The mistake is in arbitrarily assuming that the correct answer must be the lowest possible value (0.0) representing complete loss of function. While random mutations disrupt stability and function, a quadruple mutant can retain low non-zero fitness rather than complete functional collapse. Therefore, the correct fitness score corresponding to this mutant is 0.008 (Option B).
</explanation>

As discussed in Section 1, Figure 1(c) visualizes the token-by-token KL divergence under both frameworks across this exact trajectory, directly demonstrating that standard SDPO penalizes sound prefixes while failing to penalize errors, whereas ROSD shields valid prefixes and focuses corrective pressure squarely on the diagnosed logical flaw.

F Cross-Domain Generalization Results

To evaluate whether the models acquire robust, generalizable reasoning capabilities beyond their specific training domains, we conduct cross-domain evaluation where models trained on each source domain are evaluated across all other held-out domains as well as the out-of-domain mathematical reasoning benchmark AIME2024 (Feng et al., 2024). Table 5 reports the cross-domain results across all five training datasets on Qwen3-4B and Qwen3-8B.

GRPO demonstrates strong cross-domain transfer. Across training datasets and model scales, the reinforcement learning baseline (GRPO) shows strong cross-domain transfer. This is because GRPO relies solely on outcome-level reward verification without imposing strict token-level distribution constraints, thereby encouraging policy exploration and mitigating domain-specific overfitting. In contrast, on-policy self-distillation introduces dense token-level KL divergence constraints against teacher trajectories, which can make the model more prone to overfitting to domain-specific phrasing.

ROSD substantially alleviates distillation collapse and outperforms SDPO. Despite the exploration benefits of outcome-level RL, ROSD consistently outperforms standard SDPO on cross-domain average accuracy across all training datasets and both model scales. Standard SDPO suffers catastrophic collapse when transferring across substantially different domains: for instance, when trained on ToolUse and evaluated on scientific question-answering and mathematical reasoning, SDPO collapses to an average score of only 2.88% on Qwen3-4B (compared to 43.57% for GRPO). In sharp contrast, ROSD preserves strong reasoning transfer, maintaining 41.31% under the same setting. This demonstrates that by diagnosing error spans and restricting self-distillation exclusively to erroneous suffixes while safeguarding valid reasoning prefixes, ROSD prevents the memorization of brittle domain-specific syntactic templates, retaining strong cross-domain generalization while benefiting from dense on-policy supervision.

G Training Efficiency

We evaluate the training efficiency dynamics on Chemistry with the representative Qwen3-8B backbone. As shown in Figure 7, although ROSD introduces an additional self-reflection step, its overall per-step wall-clock training time is comparable to standard SDPO and consistently lower than GRPO. In later training stages, ROSD becomes even more computationally efficient: because the response lengths of the baseline methods (GRPO and SDPO) expand significantly, they incur progressively higher attention and decoding overhead. In contrast, because ROSD maintains concise and stable response lengths, its training time per step remains steady throughout optimization.

H Response Length

We evaluate the response length dynamics on Chemistry with the representative Qwen3-8B backbone. As illustrated in Figure 8, ROSD produces relatively shorter responses and maintains a stable

Train on Biology	**Method**	AIME2024	Material	Physics	Chemistry	ToolUse	Average
Qwen3-4B	GRPO	31.25	**56.16**	**56.35**	**45.23**	**58.85**	**49.57**
	SDPO	**33.33**	38.16	45.68	39.35	57.05	42.71
	ROSD	31.94	51.37	49.58	40.93	56.46	46.06
Qwen3-8B	GRPO	**38.89**	**59.04**	**60.89**	**41.58**	59.16	**51.91**
	SDPO	31.25	38.10	40.36	34.78	**59.44**	40.79
	ROSD	32.64	56.01	56.98	38.21	59.38	48.64
Train on Chemistry	**Method**	AIME2024	Biology	Material	Physics	ToolUse	Average
Qwen3-4B	GRPO	**35.42**	28.88	**49.78**	**49.71**	**58.92**	**44.54**
	SDPO	0.00	**31.54**	20.41	23.05	39.64	22.93
	ROSD	30.56	30.50	34.64	39.95	57.05	38.54
Qwen3-8B	GRPO	**43.06**	36.17	**64.18**	**61.80**	**60.23**	**53.09**
	SDPO	9.03	28.67	22.89	29.92	0.83	18.27
	ROSD	34.03	**37.08**	50.22	49.09	58.27	45.74
Train on Material	**Method**	AIME2024	Biology	Physics	Chemistry	ToolUse	Average
Qwen3-4B	GRPO	**38.89**	**33.63**	**60.78**	42.88	**60.85**	**47.41**
	SDPO	4.86	28.46	30.29	40.03	54.87	31.70
	ROSD	29.86	31.29	45.05	**45.48**	55.15	41.37
Qwen3-8B	GRPO	**39.58**	27.79	**63.39**	**46.87**	**61.31**	**47.79**
	SDPO	25.00	**35.83**	42.76	44.07	0.86	29.70
	ROSD	34.03	31.75	55.57	42.12	59.38	44.57
Train on Physics	**Method**	AIME2024	Biology	Material	Chemistry	ToolUse	Average
Qwen3-4B	GRPO	**35.42**	28.71	**71.48**	43.60	**58.98**	**47.64**
	SDPO	8.33	32.96	43.62	**43.63**	54.01	36.51
	ROSD	28.47	**33.88**	62.21	43.43	57.17	45.03
Qwen3-8B	GRPO	**44.44**	25.96	**63.76**	**46.12**	**59.93**	**48.04**
	SDPO	29.17	**34.46**	49.38	44.00	2.27	31.86
	ROSD	33.33	34.17	60.33	41.62	59.19	45.73
Train on ToolUse	**Method**	AIME2024	Biology	Material	Physics	Chemistry	Average
Qwen3-4B	GRPO	**33.33**	**30.54**	52.70	**57.29**	43.97	**43.57**
	SDPO	11.11	0.88	1.09	0.49	0.83	2.88
	ROSD	31.94	29.63	**53.30**	46.25	**45.43**	41.31
Qwen3-8B	GRPO	**36.11**	36.13	**58.82**	**57.76**	42.47	**46.26**
	SDPO	31.94	35.21	44.22	40.34	**45.18**	39.38
	ROSD	33.33	**37.04**	55.87	54.30	45.03	45.11

Table 5: Cross-domain generalization results. All methods are trained to convergence on the training dataset and scores are reported as mean@16 (%) averaged over the last three evaluation steps. **Boldface** indicates the best result, and underline indicates the second-best result within each model scale.

response-length trajectory throughout training. In contrast, the response lengths of both GRPO and SDPO increase substantially as training proceeds. By focusing distillation strictly on the diagnosed error span without forcing full-trajectory imitation, ROSD avoids superfluous verbose phrasing and prevents unnecessary reasoning dilation, demonstrating superior optimization stability.

I Error Localization Dynamics

Figure 9 provides additional evidence for the localization behavior of ROSD. The left figure shows that the error quote match rate maintains relatively high values on both Material and Chemistry, indicating that a substantial fraction of reflector outputs can be aligned back to the original rollout. The Qwen3-8B model achieves a higher match rate than Qwen3-4B on both datasets, which is consistent with its stronger instruction-following ability. The right figure shows that the normalized

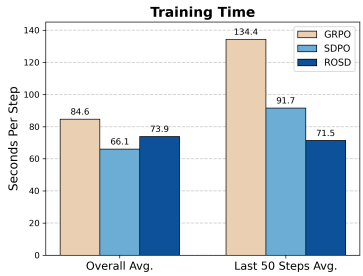


Figure 7: Training efficiency dynamics on Chemistry with Qwen3-8B across training steps (time in seconds per step for total steps and the last 50 steps).

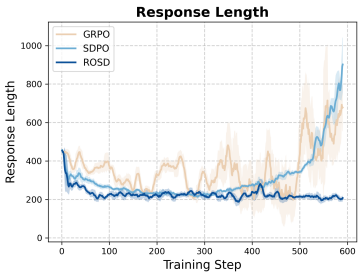


Figure 8: Response length dynamics on Chemistry with Qwen3-8B (rolling average of response length over 10 steps).

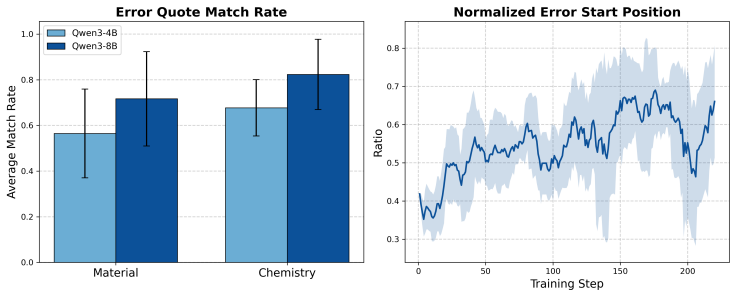


Figure 9: Error localization dynamics during training. The left figure shows the average match rate of error quotes produced by ROSD on Material and Chemistry with Qwen3-4B and Qwen3-8B. The right figure shows the average normalized error position over 10 steps on Physics with Qwen3-8B.

error position gradually moves toward the later part of the response, suggesting that errors become less frequent and occur later as training progresses. This trend also explains why full-response distillation can become harmful in the late stage: when most early reasoning steps are already valid, continuing to distill the whole response may unnecessarily perturb correct prefixes. These observations further support the necessity of localizing self-distillation to the diagnosed error span.

J Prompt Templates

Following SDPO (Hübotter et al., 2026), we use the same dataset prompts for student rollout and evaluation. For Qwen models, we disable the thinking-mode chat-template option during prompt construction. The concrete ScienceQA, ToolUse, and mathematics prompt templates are shown in Figures 10–13. The prompt used as input to the self-teacher is shown in Figure 14.

Given a question and four options, please select the right answer. Respond in the following format:
<reasoning>
...
</reasoning>
<answer>
...
</answer>
For the answer, only output the letter corresponding to the correct option (A, B, C, or D), and nothing else. Do not restate the answer text. For example, if the answer is "A", just output:
<answer>
A
</answer>

Figure 10: System prompt for ScienceQA.

[User] {question}

Please reason step by step.

Figure 11: User prompt for ScienceQA.

[User] Your task is to answer the user's question using available tools.
You have access to the following tools:
Name: {tool_name}
Description: {tool_description}
Documentation:
{function_name}: {function_description}
Parameters: {parameter_schema}
Output: {output_description}
- Format: {output_format}
- Structure: {output_structure}

Use the following format:
Thought: think about what to do
Action: the action to take; use one tool name.
Action Input: the input in JSON format.
Begin!
Question: {question}

Figure 12: User prompt for ToolUse.

[User] {question}

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

Figure 13: User prompt for AIME2024.

[System] {original system prompt}
[User] {original user prompt}
The following is the key idea to solve the question:
{key_idea}
Correctly solve the original question.
[Assistant] {original rollout}

Figure 14: Teacher input template with reflection. The self-teacher preserves the original system and user prompts used for rollout generation. The placeholder {key_idea} is replaced with the key idea generated by the self-reflector.