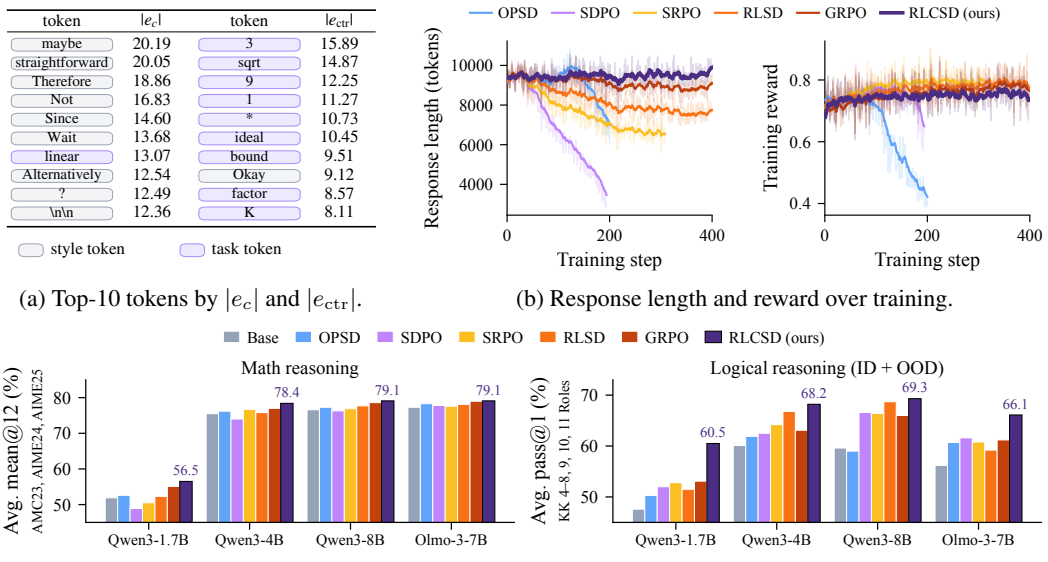
RLCSD: REINFORCEMENT LEARNING WITH CONTRASTIVE ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) provides dense, token-level supervision for reasoning models by aligning a model's own distribution with that under privileged context, typically a verified solution. However, we show that the resulting distributional gap concentrates on style tokens rather than task-bearing ones, as the hinted model tends to produce shorter, more direct outputs. We term this pathology *privilege-induced style drift*, which can destabilize training and shorten responses. To address this, we propose **RLCSD** (**R**einforcement **L**earning with **C**ontrastive on-policy **S**elf-**D**istillation), which mitigates this drift by contrasting the teacher-student gap under a correct hint against that under a wrong hint, suppressing style shifts induced by hints regardless of correctness and yielding a signal more concentrated on task-bearing tokens. Experiments on Qwen3 (1.7B/4B/8B) and Olmo-3-7B-Think across mathematical and logical reasoning show that RLCSD consistently outperforms GRPO and prior OPSD methods, with additional results on agentic tasks supporting broader applicability. We further show that the contrastive principle is general: it plugs into existing OPSD methods to improve them, and its underlying insight extends to broader cross-model on-policy distillation. Code and data are available in anonymized repositories.



(a) Top-10 tokens by $|e_c|$ and $|e_{\text{ctr}}|$.

(b) Response length and reward over training.

(c) Performance comparison across models and benchmarks.

Figure 1: **RLCSD overview. (a) Motivation.** Existing OPSD methods use $|e_c|$ as the optimization signal, which concentrates on style tokens, whereas RLCSD's contrastive signal $|e_{\text{ctr}}|$ shifts the signal toward task-bearing tokens. **(b) Response length and reward over training.** OPSD, SDPO, SRPO, and RLSD exhibit pronounced response length shrinkage during training, with OPSD and SDPO also showing training instability and sharp drops in reward. In contrast, RLCSD remains stable and maintains response length throughout training. **(c) Performance across benchmarks.** RLCSD consistently outperforms GRPO and prior OPSD methods on mathematical and logical reasoning.

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.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR), exemplified by GRPO (Shao et al., 2024), has become a dominant paradigm for training large reasoning models (Guo et al., 2025; Yang et al., 2025; Zeng et al., 2026; Hong et al., 2025; Chen et al., 2026; Team et al., 2026). However, GRPO derives its learning signal from a single outcome-level reward for each rollout. Such supervision becomes increasingly sparse in long-horizon reasoning tasks that require credit assignment across thousands of intermediate tokens. On-policy distillation (OPD) (Agarwal et al., 2024; Song & Zheng, 2026; Gu et al., 2024) addresses this limitation by obtaining dense token-level supervision from a stronger teacher along student-sampled trajectories. Its on-policy formulation mitigates the exposure bias of supervised fine-tuning, while fine-grained supervision accelerates convergence. Recent studies show that OPD can match or outperform RLVR, and it has been incorporated into the post-training pipelines of Qwen3 (Yang et al., 2025), GLM-5 (Zeng et al., 2026), the MiMo series (Xiao et al., 2026), and DeepSeek-V4 (DeepSeek-AI, 2026).

The applicability of OPD is nevertheless limited by its reliance on white-box teacher logits, a shared teacher–student vocabulary, and the continuous deployment of a typically larger teacher during training. On-policy self-distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026; Yang et al., 2026; He et al., 2026; Li et al., 2026a; Jin et al., 2026; Kim et al., 2026; Shen et al., 2026a;b; Ke et al., 2026; Wang et al., 2026) circumvents these requirements by using the same model as both teacher and student, while granting the teacher privileged context unavailable to the student, such as a verified reference solution.

Existing OPSD methods (Zhao et al., 2026; Hübotter et al., 2026; Yang et al., 2026) derive token-level supervision from $e_{c,t} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t})$, where x is the query, y is a student-sampled rollout, T and S denote the same model under different conditioning, and y_c^* is a privileged solution. Although conditioning on a verified solution sharpens the teacher's predictions on task-bearing tokens, it also induces systematic, correctness-orthogonal changes in generative behavior: the privileged teacher favors shorter and more assertive responses, suppresses exploratory expressions, and reallocates probability mass toward formatting and discourse tokens. We call this parasitic component ***privilege-induced style drift***. As shown in Figure 1a, stylistic tokens dominate the resulting learning signal and dilute supervision on task-bearing tokens. This leads to the pathologies illustrated in Figure 1b: **premature response-length shrinkage that limits later-stage reasoning performance, sometimes accompanied by training instability**.

To address this issue, we propose **RLCSD**: **R**einforcement **L**earning with **C**ontrastive on-policy **S**elf-**D**istillation. Rather than relying only on a correct privileged solution, RLCSD conditions the teacher on correct and incorrect reference solutions in parallel:

$$e_{c,t} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t}), \tag{1}$$

$$e_{w,t} = \log \pi_T(y_t \mid x, y_w^*, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t}), \tag{2}$$

$$e_{\text{ctr},t} = e_{c,t} - e_{w,t}, \tag{3}$$

where y_c^* and y_w^* denote correct and incorrect reference solutions, respectively. Since datasets rarely provide incorrect references, we obtain y_w^* from an incorrect student rollout for the same query. When the two references are presented using the same prompt template, their shared stylistic effects can cancel, leaving a signal more closely associated with task correctness. As illustrated in Figure 1a, tokens ranked by $|e_c|$ are dominated by stylistic markers such as "Wait" and "Therefore," whereas those ranked by $|e_{\text{ctr}}|$ more often contain task-relevant mathematical content.

RLCSD uses $e_{\text{ctr},t}$ to modulate the query-level GRPO advantage A_{ORM} at each token rather than replacing it. The verifier therefore determines the direction of the update, while the contrastive signal refines token-level credit assignment. Additional designs detailed in Section 3 prevent this modulation from reversing the direction of A_{ORM} or being diluted by loss aggregation.

We evaluate RLCSD on Qwen3 (Yang et al., 2025) at three scales (1.7B, 4B, and 8B) and Olmo-3-7B-Think (Olmo et al., 2025). Our main evaluation covers mathematical reasoning (AMC23 (MAA, 2023), AIME24 (Zhang & Math-AI, 2024), and AIME25 (Zhang & Math-AI, 2025)) and logical reasoning (Knights & Knaves (Xie et al., 2025) at both in-distribution and out-of-distribution difficulty levels). As shown in Figure 1c, RLCSD consistently outperforms GRPO and prior self-distillation methods, including OPSD (Zhao et al., 2026), SDPO (Hübotter et al., 2026), SRPO (Li et al., 2026a), and RLSD (Yang et al., 2026). It also maintains stable training and preserves response length, whereas

competing methods either destabilize or exhibit premature length shrinkage (Figure 1b). Results on agentic tasks in Appendix I further demonstrate its broader applicability.

Our contributions are summarized as follows:

- We identify *privilege-induced style drift*, a systematic confound that shifts OPSD supervision from task-bearing tokens toward stylistic tokens, causing premature response-length shrinkage and training instability.
- We propose **RLCSD**, which suppresses this drift through a symmetric contrast between correct and incorrect privileged contexts and uses the resulting signal to modulate the verifier-anchored GRPO advantage.
- Across Qwen3 at three scales and Olmo-3-7B-Think, RLCSD consistently outperforms GRPO and prior OPSD methods on mathematical and logical reasoning while maintaining stable training dynamics. Ablations further show that the contrastive principle generalizes to existing self-distillation methods and the broader OPD setting.

2 Related Work

2.1 RLVR and On-Policy Distillation

Reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Yu et al., 2026a; Liu et al., 2025; Zheng et al., 2025; Pan et al., 2025) has become a central post-training paradigm for improving the reasoning capabilities of large language models. Representative algorithms such as GRPO (Shao et al., 2024) eliminate the critic network and estimate advantages by normalizing verifier rewards across multiple rollouts for the same query. However, the resulting supervision remains trajectory-level: a single scalar advantage is assigned to every token in a response, providing no fine-grained credit assignment over intermediate reasoning steps.

On-policy distillation (OPD) (Gu et al., 2024; Agarwal et al., 2024) complements RLVR by computing token-level distribution gaps between a teacher and the student along the student's own trajectories. A canonical formulation minimizes the reverse KL divergence:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{y\sim\pi_S}\left[\frac{1}{|y|}\sum_{t=1}^{|y|}\mathrm{KL}\left(\pi_S(\cdot \mid x, y_{<t}) \,\|\, \mathrm{sg}[\pi_T(\cdot \mid x, y_{<t})]\right)\right], \tag{4}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient. In the policy-gradient view, this objective corresponds to REINFORCE (Williams, 1992) with the per-token reward $r_t = \log \pi_T(y_t \mid x, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t})$ (Gu et al., 2024; Lu & Thinking Machines Lab, 2025).

Prior work has explored alternative divergences, including forward KL, Jensen–Shannon divergence (Agarwal et al., 2024), and skew KL (Ko et al., 2024), as well as OPD with black-box teachers (Ye et al., 2025) and more training-efficient implementations (Wu et al., 2026). MiMo-V2-Flash (Xiao et al., 2026), for example, combines the dense token-level OPD advantage with the verifier-derived advantage A_{ORM}. Recent studies further examine when OPD is effective, highlighting teacher–student compatibility, genuine capability gaps, appropriate response lengths, and unbiased approximations such as top-k truncation (Li et al., 2026b; Fu et al., 2026; Zhu et al., 2026).

2.2 On-Policy Self-Distillation

Standard OPD requires token-level teacher logits, a shared vocabulary, and continuous deployment of a separate teacher, limiting its applicability across model families and introducing substantial serving overhead. On-policy self-distillation (OPSD) addresses these constraints by using the same model as both teacher and student while providing the teacher with privileged context, such as a ground-truth solution or terminal environment feedback (Zhao et al., 2026; Han et al., 2026; Yu et al., 2026b; Hao et al., 2026; Heakl et al., 2026). Its standard objective is

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{y\sim\pi_S}\left[\frac{1}{|y|}\sum_{t=1}^{|y|}\mathrm{KL}\left(\pi_S(\cdot \mid x, y_{<t}) \,\|\, \mathrm{sg}[\pi_T(\cdot \mid x, y_c^*, y_{<t})]\right)\right], \tag{5}$$

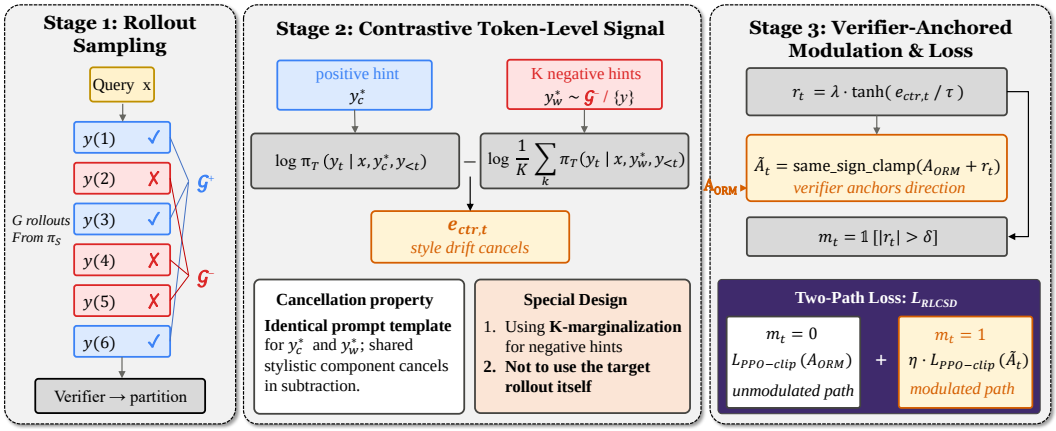


Figure 2: **Overview of RLCSD. Stage 1** samples student rollouts and partitions them into correct ($\mathcal{G}^+$) and incorrect ($\mathcal{G}^-$) subsets using a verifier. **Stage 2** conditions the teacher on a positive hint y_c^* from a ground-truth solution or correct rollout, and up to K negative hints from incorrect rollouts. Rollout-derived hints exclude the target itself. Both branches use the same prompt template; their contrast yields $e_{\mathrm{ctr},t}$, suppressing shared stylistic effects. **Stage 3** converts $e_{\mathrm{ctr},t}$ into a bounded modulation r_t, applies it to A_{ORM} with a sign-preserving clamp, and combines independently normalized unmodulated and modulated paths in a PPO-style clipped objective.

where the teacher and student are the same model and y_c^* denotes the privileged context. This objective induces the per-token advantage $A_t = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t})$.

Existing methods instantiate this framework in different ways. OPSD (Zhao et al., 2026) uses dense forward-KL distillation and applies pointwise KL clipping to control large token-level signals. SDPO (Hübotter et al., 2026) instead adopts the Jensen–Shannon divergence. SRPO (Li et al., 2026a) routes correct rollouts to GRPO and incorrect rollouts to SDPO-style dense distillation. RLSD (Yang et al., 2026) departs from full-distribution matching and uses a sampled-token signal, as does RLCSD. It further constrains A_{ORM} to determine the update direction while using the self-distillation signal only to modulate its magnitude.

Despite these differences, the methods discussed above construct supervision from one-sided positive privileged contexts, leaving correctness-independent stylistic shifts entangled with task-relevant signals. **RLCSD mitigates this issue through contrastive conditioning on correct and incorrect hints, thereby removing the shared drift**. Concurrent work, CEPO (Heakl et al., 2026), also explores positive-negative conditioning for advantage modulation. Appendix B discusses the shared principle and the distinct focus and design of RLCSD.

3 Method

3.1 Problem: Privilege-Induced Style Drift

Existing OPSD methods derive token-level supervision from $e_{c,t} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \pi_S(y_t \mid x, y_{<t})$, where the teacher and student are the same model with and without the privileged context y_c^*. Although conditioning on a verified solution sharpens the teacher's predictions on task-bearing tokens, it also induces correctness-orthogonal changes in generation style, including shorter and more assertive phrasing, fewer exploratory expressions, and increased probability mass on formatting and discourse tokens. We call this parasitic component *privilege-induced style drift*.

To quantify this effect, we partition tokens in mathematical reasoning responses into task, style, and neutral categories, with the complete procedure provided in Ap-

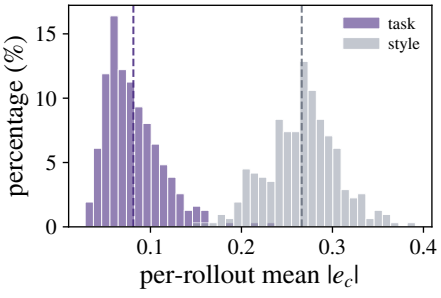


Figure 3: Per-rollout mean $|e_{c,t}|$ on style vs. task tokens. The privilege shifts most of its influence onto style tokens (0.263 vs. 0.083).

pendix G. For each rollout, we compute the mean $|e_{c,t}|$ separately over its task and style tokens. As shown in Figure 3, the mean signal on style tokens is 0.263, approximately three times the task-token mean of 0.083. Thus, much of the one-sided privileged signal reflects changes in expression rather than reasoning correctness. As shown in Section 4, this confound manifests as training instability or premature response-length shrinkage in existing OPSD methods.

3.2 Overview of RLCSD

RLCSD suppresses privilege-induced style drift by contrasting teacher distributions conditioned on correct and incorrect privileged contexts. The resulting token-level signal modulates, rather than replaces, the verifier-derived GRPO advantage. As illustrated in Figure 2, RLCSD consists of three stages: (1) constructing privileged contexts, (2) computing the contrastive signal, and (3) incorporating it into a two-path clipped objective.

3.3 Contrastive Token-Level Signal

For each query x, we sample a group of G rollouts $\{y^{(g)}\}_{g=1}^{G}$ from the student policy. A rule-based verifier partitions them into correct and incorrect subsets, denoted by $\mathcal{G}^+$ and $\mathcal{G}^-$. We discard groups with no correct rollout or fewer than two incorrect rollouts. All-correct and all-incorrect groups have zero group-relative GRPO advantage; groups with only one incorrect rollout are also discarded because that rollout would have no negative hint after self-exclusion.

For each target rollout y, we select a positive privileged context y_c^* from either a dataset-provided ground-truth reasoning trace or a correct sibling rollout. Negative hints are drawn from $\mathcal{G}^- \setminus \{y\}$, using $K_y = \min(K, |\mathcal{G}^- \setminus \{y\}|)$ candidates, where K is the maximum number of negative hints. When fewer than K candidates are available, we use all available candidates. Positive and negative contexts share the prompt template in Appendix D.1, allowing shared privilege-induced stylistic effects to be suppressed.

The resulting contrastive signal is:

$$e_{\mathrm{ctr},t} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \left[\frac{1}{K_y} \sum_{k=1}^{K_y} \pi_T(y_t \mid x, y_{w,k}^*, y_{<t}) \right]. \tag{6}$$

The student term in the corresponding teacher–student gaps cancels under subtraction. Therefore, $e_{\mathrm{ctr},t}$ directly measures how much more the teacher favors token y_t under a correct reference than under the distribution of incorrect references.

We apply two refinements when constructing this signal. First, marginalizing over the available negative contexts, up to K per target rollout, reduces sensitivity to the heterogeneous error modes of individual incorrect solutions. Second, whenever contexts are sampled from the rollout group, we exclude the target rollout y itself: negative contexts are drawn from $\mathcal{G}^- \setminus \{y\}$, while rollout-derived positive contexts are drawn from $\mathcal{G}^+ \setminus \{y\}$. This prevents self-conditioning from making the teacher excessively confident in the target tokens. Detailed motivations and empirical analyses of both refinements are provided in Appendix C. As shown in Figure 4, the style–task separation under $|e_{\mathrm{ctr},t}|$ is substantially smaller than under $|e_{c,t}|$. The contrast therefore suppresses the stylistic component shared by the two conditionings and shifts the signal toward task-bearing tokens.

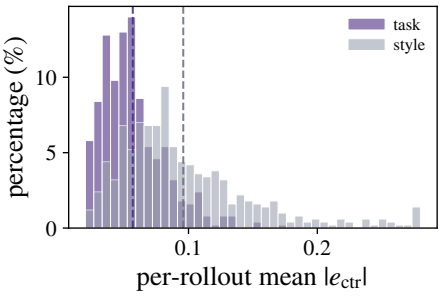


Figure 4: Task/style analysis of $|e_{\mathrm{ctr},t}|$. The reduced style–task gap indicates a cleaner signal than $|e_c|$ (Figure 3).

3.4 Verifier-Anchored Advantage Modulation

Let $A_{\mathrm{ORM}}^{(g)}$ denote the standard group-normalized GRPO advantage of rollout $y^{(g)}$. It provides a verifier-grounded update direction but assigns the same value to every token in the rollout. We

use $e_{\mathrm{ctr},t}$ to refine its magnitude at selected tokens. We first transform the contrastive signal into a bounded modulation and retain only tokens with sufficiently strong signals:

$$r_t = \lambda \tanh\left(\frac{e_{\mathrm{ctr},t}}{\tau}\right), \qquad m_t = \mathbf{1}(|r_t| > \delta), \tag{7}$$

where τ controls the slope, λ controls the modulation scale, and δ is the token-selection threshold. For a selected token, we add r_t to the outcome advantage and clamp the result to preserve its original sign:

$$\tilde{A}_t = \begin{cases} \max(0, A_{\mathrm{ORM}} + r_t), & A_{\mathrm{ORM}} \geq 0, \\ \min(0, A_{\mathrm{ORM}} + r_t), & A_{\mathrm{ORM}} < 0. \end{cases} \tag{8}$$

The verifier thus determines whether an action should be encouraged or suppressed, while the contrastive signal determines how strongly individual tokens are updated.

3.5 Two-Path Training Objective

For rollout $y^{(g)}$, let $\rho_{g,t}(\theta) = \frac{\pi_\theta(y_t^{(g)}|x, y_{<t}^{(g)})}{\pi_{\theta_{\mathrm{old}}}(y_t^{(g)}|x, y_{<t}^{(g)})}$ denote the token-level importance ratio. We define the PPO-style clipped surrogate (Schulman et al., 2017) as:

$$\psi_{g,t}(A;\theta) = \min(\rho_{g,t}(\theta)A,\ \mathrm{clip}\left(\rho_{g,t}(\theta), 1-\epsilon, 1+\epsilon\right) A). \tag{9}$$

Let $\mathcal{U}^{(g)} = \{t : m_t^{(g)} = 0\}$ and $\mathcal{M}^{(g)} = \{t : m_t^{(g)} = 1\}$ denote the unmodulated and modulated token sets. The rollout-level objective is:

$$J_{\mathrm{RLCSD}}^{(g)}(\theta) = \frac{1}{|\mathcal{U}^{(g)}|} \sum_{t \in \mathcal{U}^{(g)}} \psi_{g,t}\left(A_{\mathrm{ORM}}^{(g)}; \theta\right) + \eta \frac{1}{|\mathcal{M}^{(g)}|} \sum_{t \in \mathcal{M}^{(g)}} \psi_{g,t}\left(\tilde{A}_t^{(g)}; \theta\right), \tag{10}$$

where η controls the contribution of the modulated path. If either token set is empty, its corresponding path is omitted. Independently normalizing the two paths prevents the relatively sparse modulated tokens from being diluted by the remaining tokens.

The complete objective averages over rollouts and training queries:

$$J_{\mathrm{RLCSD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}}\left[\frac{1}{G}\sum_{g=1}^{G} J_{\mathrm{RLCSD}}^{(g)}(\theta)\right], \qquad \mathcal{L}_{\mathrm{RLCSD}}(\theta) = -J_{\mathrm{RLCSD}}(\theta). \tag{11}$$

4 Experiments

We conduct comprehensive experiments to answer the following research questions:

- **RQ1:** How does RLCSD compare with GRPO and other OPSD baselines in terms of task performance and training stability? (§4.2)
- **RQ2:** Can the proposed contrastive-hint signal improve not only RLCSD itself but also other OPSD baselines as a general-purpose component? (§4.3)
- **RQ3:** How important are the individual design choices in RLCSD, including reference-hint selection, integration with A_{ORM}, and the two-path loss aggregation? (§4.4)
- **RQ4:** What broader insights does RLCSD offer for on-policy distillation (OPD)? (§4.5)

4.1 Experimental Setup

Models and tasks. We evaluate Qwen3-1.7B/4B/8B and Olmo-3-7B-Think, with thinking mode enabled throughout training and evaluation. For mathematics, models are trained on DeepMath-103K (He et al., 2025) using difficulty levels 5–7 for Qwen3-1.7B, 6–8 for Qwen3-4B, and 7–10 for Qwen3-8B and Olmo-3-7B-Think, and evaluated on AMC23, AIME24, and AIME25 (MAA, 2023; Zhang & Math-AI, 2024; 2025). For logical reasoning, models are trained on 4–8-role Knights-and-Knaves problems and evaluated both in distribution (4–8 roles) and out of distribution (9–11 roles) (Xie et al., 2025). Results on ALFWorld and WebShop are provided in Appendix I.

Table 1: Main results across model scales. Math reasoning (AMC23, AIME24, AIME25) is reported as mean@12; logical reasoning (Knight-and-Knaves) is reported as pass@1, with 4–8 Roles as the in-domain (ID) setting and 9 Roles, 10 Roles, and 11 Roles as out-of-domain (OOD) settings. **Bold** marks the best result in each block, and underline marks the second-highest result; green subscripts denote the gain over the base model.

		Math Reasoning (mean@12)				**Logical Reasoning: Knight-and-Knaves** (pass@1)				
Model	**Method**	AMC23	AIME24	AIME25	Avg.	4–8 Roles	9 Roles	10 Roles	11 Roles	Avg.
Qwen3-1.7B	Base	74.1	48.3	33.3	51.9	63.2	53.0	43.0	31.0	47.6
	GRPO	76.6	51.6	37.2	55.1	67.4	59.0	52.0	34.0	53.1
	OPSD	74.6	47.2	36.1	52.6	62.0	59.0	51.0	29.0	50.3
	SDPO	73.0	42.2	31.4	48.9	63.8	56.0	52.0	36.0	52.0
	SRPO	72.6	45.6	33.3	50.5	63.0	59.0	54.0	35.0	52.8
	RLSD	73.9	46.1	36.9	52.3	68.0	57.0	49.0	32.0	51.5
	RLCSD	**78.2**$_{+4.1}$	**53.1**$_{+4.8}$	**38.3**$_{+5.0}$	**56.5**$_{+4.6}$	**71.8**$_{+8.6}$	**66.0**$_{+13.0}$	**61.0**$_{+18.0}$	**43.0**$_{+12.0}$	**60.5**$_{+12.9}$
Qwen3-4B	Base	88.6	72.5	65.3	75.5	73.2	67.0	58.0	42.0	60.1
	GRPO	89.1	75.8	66.1	77.0	75.4	71.0	61.0	45.0	63.1
	OPSD	89.4	73.3	65.8	76.2	73.4	71.0	61.0	42.0	61.9
	SDPO	88.2	70.3	63.6	74.0	75.8	71.0	60.0	43.0	62.5
	SRPO	89.2	74.2	66.7	76.7	74.6	**76.0**	59.0	47.0	64.2
	RLSD	88.9	72.5	66.1	75.8	76.0	75.0	67.0	49.0	66.8
	RLCSD	**90.8**$_{+2.2}$	**76.4**$_{+3.9}$	**68.1**$_{+2.8}$	**78.4**$_{+2.9}$	**77.6**$_{+4.4}$	**76.0**$_{+9.0}$	**68.0**$_{+10.0}$	**51.0**$_{+9.0}$	**68.2**$_{+8.1}$
Qwen3-8B	Base	88.8	74.2	66.9	76.6	72.4	67.0	55.0	44.0	59.6
	GRPO	90.1	76.1	69.7	78.6	76.8	75.0	63.0	49.0	66.0
	OPSD	90.0	74.7	67.2	77.3	68.8	65.0	56.0	46.0	59.0
	SDPO	90.1	74.2	64.7	76.3	76.4	78.0	66.0	46.0	66.6
	SRPO	89.4	75.6	65.8	76.9	76.6	76.0	63.0	50.0	66.4
	RLSD	89.6	75.6	67.8	77.7	76.8	**79.0**	**67.0**	52.0	68.7
	RLCSD	**91.0**$_{+2.2}$	**76.4**$_{+2.2}$	**70.0**$_{+3.1}$	**79.1**$_{+2.5}$	**78.2**$_{+5.8}$	**79.0**$_{+12.0}$	66.0$_{+11.0}$	**54.0**$_{+10.0}$	**69.3**$_{+9.7}$
Olmo-3-7B	Base	91.2	73.9	66.9	77.3	70.6	64.0	55.0	35.0	56.2
	GRPO	92.4	75.8	**68.9**	79.0	73.8	69.0	63.0	39.0	61.2
	OPSD	92.3	75.4	67.1	78.3	72.6	70.0	61.0	39.0	60.7
	SDPO	91.8	74.3	67.2	77.8	73.4	68.0	60.0	45.0	61.6
	SRPO	91.9	75.2	65.6	77.6	73.0	67.0	62.0	41.0	60.8
	RLSD	92.5	74.9	66.8	78.1	73.6	66.0	60.0	37.0	59.2
	RLCSD	**92.7**$_{+1.5}$	**76.1**$_{+2.2}$	68.6$_{+1.7}$	**79.1**$_{+1.8}$	**75.4**$_{+4.8}$	**76.0**$_{+12.0}$	**65.0**$_{+10.0}$	**48.0**$_{+13.0}$	**66.1**$_{+9.9}$

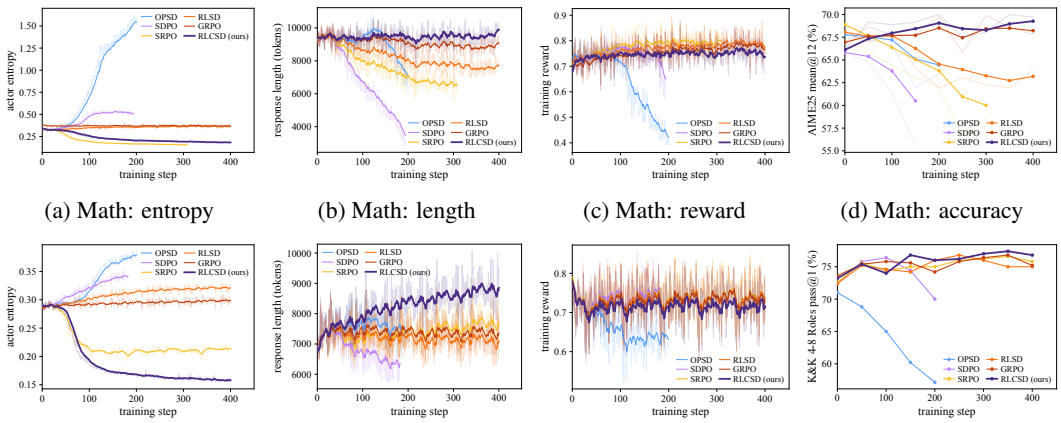
Baselines and implementation. We compare against GRPO (Shao et al., 2024) and four OPSD methods: OPSD (Zhao et al., 2026), SDPO (Hübotter et al., 2026), SRPO (Li et al., 2026a), and RLSD (Yang et al., 2026). All methods use full-parameter training on 8×H20 GPUs with a learning rate of 10^{-6} and identical rollout and evaluation settings. RLCSD uses a fixed teacher and $(K, \tau, \lambda, \delta, \eta) = (4, 1.3, 0.5, 0.02, 0.5)$. See sensitivity analysis in Appendix J. All self-distillation methods use dataset-provided ground-truth solutions as positive contexts; RLCSD draws negative contexts from incorrect same-query student rollouts. For the main results, we report the checkpoint with the best validation performance for each method and model. See full settings in Appendix D.

4.2 Main Results

Performance. Table 1 shows that RLCSD achieves the best average performance on both task families across all four model scales. Relative to the base models, its gains range from $+1.8$ to $+4.6$ points on mathematics and from $+8.1$ to $+12.9$ points on logical reasoning. The improvements are particularly pronounced on the OOD 9–11-role settings, where RLCSD gains 9.0–18.0 points, indicating strong generalization beyond the training distribution.

Failure Modes of Other On-policy Self-distillation Methods. Figure 5 shows that existing methods exhibit *premature response-length shrinkage, sometimes accompanied by training instability*. (1) **On math reasoning**, all four methods show declining response lengths (Figure 5b) and deteriorating or limited later-stage performance (Figure 5d). OPSD and SDPO also exhibit rising entropy and sharp reward drops (Figures 5a and 5c). (2) **On logical reasoning**, shrinkage is most evident for SDPO, while other baselines plateau at shorter lengths than RLCSD (Figure 5f). OPSD and SDPO also show rising entropy and declining validation accuracy (Figures 5e and 5h), with OPSD exhibiting a sustained reward drop (Figure 5g).

Stable Dynamics of RLCSD. In contrast, RLCSD maintains stable policy entropy and response length throughout training (Figures 5a, 5b, 5e, and 5f), comparable to GRPO. RLCSD augments



(a) Math: entropy (b) Math: length (c) Math: reward (d) Math: accuracy

(e) K&K: entropy (f) K&K: length (g) K&K: reward (h) K&K: accuracy

Figure 5: Training dynamics of Qwen3-8B on math reasoning (top) and Knights-and-Knaves (bottom). Prior OPSD methods exhibit premature response-length shrinkage or unstable entropy, reward, and validation performance, whereas RLCSD remains stable and preserves response length.

Table 2: Contrastive privileged contexts as a plug-in component. For each method, we compare a one-sided construction using a correct reference solution with a contrastive construction that additionally uses an incorrect student rollout for the same query. Math reasoning (AMC23, AIME24, AIME25) is reported as mean@12; Knight-and-Knaves is reported as pass@1, with 4–8 Roles as the in-domain (ID) setting and 9–11 Roles as out-of-domain (OOD) settings. All results use Qwen3-4B; **bold** marks the best result within each method block, including ties. Δ denotes the gain of the contrastive construction over its one-sided counterpart.

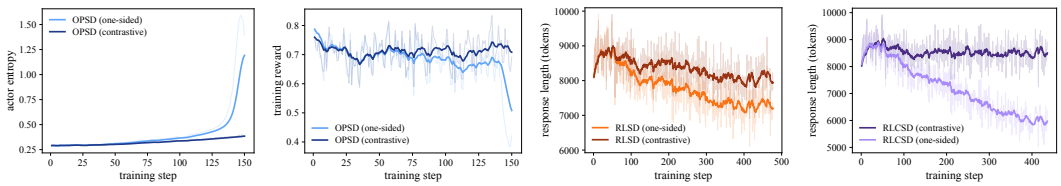
		Math Reasoning (mean@12)				**Logical Reasoning: Knight-and-Knaves** (pass@1)				
Method	**Context construction**	AMC23	AIME24	AIME25	Avg.	4–8 Roles	9 Roles	10 Roles	11 Roles	Avg.
OPSD	One-sided	89.4	73.3	65.8	76.2	73.4	**71.0**	**61.0**	42.0	61.9
	Contrastive	**89.6**	**75.6**	**66.4**	**77.2**	**74.2**	**71.0**	**61.0**	**45.0**	**62.8**
	Δ	+0.2	+2.3	+0.6	+1.0	+0.8	0.0	0.0	+3.0	+0.9
RLSD	One-sided	88.9	72.5	66.1	75.8	76.0	**75.0**	**67.0**	**49.0**	66.8
	Contrastive	**89.8**	**75.3**	**66.4**	**77.2**	**76.4**	**75.0**	**67.0**	**49.0**	**66.9**
	Δ	+0.9	+2.8	+0.3	+1.4	+0.4	0.0	0.0	0.0	+0.1
RLCSD	One-sided	88.9	73.1	62.5	74.8	73.8	71.0	58.0	43.0	61.5
	Contrastive	**90.8**	**76.4**	**68.1**	**78.4**	**77.6**	**76.0**	**68.0**	**51.0**	**68.2**
	Δ	+1.9	+3.3	+5.6	+3.6	+3.8	+5.0	+10.0	+8.0	+6.7

GRPO's verifier-derived outcome signal with token-level contrastive modulation, enabling finer-grained credit assignment without sacrificing stability. It thus preserves GRPO's stable optimization behavior while achieving better validation performance (Figures 5d and 5h).

4.3 Contrastive Privileged Contexts as a General Component

Generalization to OPSD and RLSD. To test whether contrastive conditioning benefits existing methods, we incorporate it into OPSD and RLSD. For RLSD, $e_{\mathrm{ctr},t}$ replaces the one-sided gap in its original advantage modulation. For OPSD, we construct a contrastively guided teacher distribution and a disagreement-based soft token mask (See Appendix E for details). These variants improve math/K&K averages by $1.0/0.9$ points for OPSD and $1.4/0.1$ for RLSD (Table 2), while reducing entropy growth in OPSD and response-length shrinkage in RLSD (Figure 6).

Controlled comparison within RLCSD. We replace $e_{\mathrm{ctr},t}$ with $e_{c,t}$ while keeping the modulation rule, sign-preserving clamp, selection threshold, two-path objective, and training settings unchanged. Token masks are computed from the respective signals. Contrast improves math/K&K averages by $3.6/6.7$ points (Table 2) and mitigates response-length shrinkage (Figure 6). Since both variants use two-path aggregation, these gains establish the benefit of contrastive conditioning within the same



(a) OPSD: entropy (b) OPSD: reward (c) RLSD: length (d) RLCSD: length

Figure 6: Effect of contrastive privileged contexts on Qwen3-4B. Contrast stabilizes OPSD on K&K (a–b) and mitigates premature response-length shrinkage in RLSD and RLCSD on math (c–d).

Table 3: Ablation of key design choices in RLCSD (Qwen3-4B). Each variant removes a single component from the full method. Math reasoning (AMC23, AIME24, AIME25) is reported as mean@12; Knight-and-Knaves (pass@1) uses roles 4–8 as in-domain (ID) and roles 9/10/11 as out-of-domain (OOD). Bold marks the best result in each column.

	Math Reasoning (mean@12)				Logical Reasoning: Knight-and-Knaves (pass@1)				
Method	AMC23	AIME24	AIME25	Avg.	4–8 Roles	9 Roles	10 Roles	11 Roles	Avg.
RLCSD (full)	**90.8**	**76.4**	**68.1**	**78.4**	**77.6**	**76.0**	**68.0**	**51.0**	**68.2**
− K-marginal negative hint	89.3	73.2	67.8	76.8	76.4	71.0	63.0	47.0	64.4
+ target rollout in hint pool	89.1	73.4	68.0	76.8	75.8	70.0	63.0	46.0	63.7
− A_{ORM} anchoring	88.6	72.1	66.3	75.7	74.6	69.0	61.0	45.0	62.4
− two-path loss aggregation	89.0	73.1	67.2	76.4	75.6	70.0	62.0	46.0	63.4

framework. Together with the task/style analysis, they support the interpretation that suppressing shared stylistic effects contributes to the improvements.

4.4 Ablation of Key Design Choices

Table 3 validates the four main design choices of RLCSD. Replacing K-marginalization with one negative hint reduces math/K&K by $1.6/3.8$ points, while allowing the target rollout into its own hint pool causes drops of $1.6/4.5$ points. These results confirm the importance of reducing error-type mismatch and avoiding self-conditioning. Removing the A_{ORM} anchor produces the largest degradation ($-2.7/-5.8$), showing that the contrastive signal should refine rather than determine the update direction. Finally, replacing independent two-path normalization with a global token average reduces performance by $2.0/4.8$ points because the relatively sparse modulated tokens are diluted by the unmodulated majority.

4.5 Broader Insights for On-Policy Distillation

The task-versus-style decomposition also appears in cross-model OPD. Using Qwen3-1.7B-Base as the student, we compare a stylistically distant Qwen3-4B-Instruct teacher with a more aligned Qwen3-4B-Base-GRPO teacher. The former exhibits its largest per-token KL values at discourse and formatting positions, while the latter emphasizes mathematical content. Despite stronger standalone performance, the instruct teacher yields a weaker student (15.7 vs. 20.7 average math accuracy). This suggests that effective OPD depends on both teacher capability and whether its signal reflects task knowledge rather than stylistic mismatch. Settings, token-level statistics, and full results appear in Appendix F.

5 Conclusion

We identified *privilege-induced style drift* in on-policy self-distillation, where privileged conditioning biases supervision toward stylistic tokens. RLCSD mitigates this drift by contrasting correct and incorrect hints to modulate A_{ORM}. Across Qwen3 and Olmo models, RLCSD outperforms GRPO and prior OPSD methods on mathematical and logical reasoning while maintaining stable training, with further gains on agentic tasks. Our findings also support contrastive conditioning in existing OPSD methods and distinguishing task knowledge from stylistic mismatch in broader OPD.

Ethics Statement

This work studies training methods for improving language-model reasoning using publicly available benchmarks. It does not involve human subjects or private data. While improved reasoning capabilities may have broader societal implications, we do not identify ethical risks specific to the proposed method beyond those generally associated with capable language models.

Reproducibility Statement

We provide detailed descriptions of the method, training configuration, hyperparameters, datasets, and evaluation protocols in the main text and appendices. Code and data are available in anonymized repositories to facilitate reproduction of our results.

References

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.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changqing Yu, Chao Wang, Cheng Ma, Cheng Zhong, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

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

Zihao Han, Tiangang Zhang, Huaibin Wang, and Yilun Sun. Adaptive teacher exposure for self-distillation in llm reasoning. *arXiv preprint arXiv:2605.11458*, 2026.

Guangya Hao, Yitong Shang, Yunbo Long, Zhuokai Zhao, and Hanxue Liang. Self-policy distillation via capability-selective subspace projection. *arXiv preprint arXiv:2605.22675*, 2026.

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.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Ahmed Heakl, Abdelrahman M Shaker, Youssef Mohamed, Rania Elbadry, Omar Fetouh, Fahad Shahbaz Khan, and Salman Khan. Cepo: Rlvr self-distillation using contrastive evidence policy optimization. *arXiv preprint arXiv:2605.19436*, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

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.

Yiqiao Jin, Yiyang Wang, Lucheng Fu, Yijia Xiao, Yinyi Luo, Haoxin Liu, B Aditya Prakash, Josiah Hester, Jindong Wang, and Srijan Kumar. Unisd: Towards a unified self-distillation framework for large language models. *arXiv preprint arXiv:2605.06597*, 2026.

Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting self-uncertainty in on-policy self-distillation for efficient llm reasoning. *arXiv preprint arXiv:2605.13255*, 2026.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled rlvr. *arXiv preprint arXiv:2605.10781*, 2026.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 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.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huang 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. https://thinkingmachines.ai/blog/on-policy-distillation/, 2025.

MAA. Amc 2023. https://huggingface.co/datasets/math-ai/amc23, 2023. Dataset compiled from the 2023 American Mathematics Competitions.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Zheyu Fu, Liancheng Fang, Minghua He, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, et al. d-treerpo: Towards more reliable policy optimization for diffusion language models. *arXiv preprint arXiv:2512.09675*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

Guobin Shen, Xiang Cheng, Chenxiao Zhao, Lei Huang, Jindong Li, Dongcheng Zhao, and Xing Yu. Anti-self-distillation for reasoning rl via pointwise mutual information. *arXiv preprint arXiv:2605.11609*, 2026a.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation. *arXiv preprint arXiv:2605.11613*, 2026b.

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. Trace: Distilling where it matters via token-routed self on-policy alignment. *arXiv preprint arXiv:2605.10194*, 2026.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Yecheng Wu, Song Han, and Hai Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

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.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 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, 2026a.

Xin Yu, Liuchen Liao, Yiwen Zhang, Yingchen Yu, Lingzhou Xue, and Qinzhen Guo. Preference-based self-distillation: Beyond kl matching via reward regularization. *arXiv preprint arXiv:2605.05040*, 2026b.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 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. *arXiv preprint arXiv:2601.18734*, 2026.

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.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026.

Appendices

Table of Contents

A AI Use Statement **15**

B Comparison with Concurrent Work **15**

C Additional Analysis of Reference-Hint Selection **16**

C.1 Negative-Hint Selection: Case Analysis and K-Marginalization 16

C.2 Self-Referential Collapse and Excluding the Target Rollout 17

D Implementation Details **18**

D.1 Privileged Prompt Template 18

D.2 Shared training setup 18

D.3 Method-specific hyperparameters 19

E Additional Details of the Contrastive Plug-In **19**

F Broader Insights for Cross-Model On-Policy Distillation **20**

F.1 Experimental Setup 20

F.2 Which Tokens Dominate the OPD Signal? 20

F.3 Effect on Distillation Performance 20

F.4 Implications 21

G Vocabulary Partitioning for Task / Style Statistics **21**

H Computational Cost Analysis **22**

I Additional Results on Agentic Tasks **23**

J Sensitivity Analysis of Hyperparameters in RLCSD **23**

J.1 How the Hyperparameters Affect the Learning Signal 23

J.2 Training Dynamics under the Default Configuration 25

J.3 Controlled Hyperparameter Comparisons 25

A AI USE STATEMENT

Generative AI tools were used solely to improve the clarity, grammar, and presentation of the manuscript. All technical content, experiments, analyses, and conclusions were developed and verified by the authors.

B COMPARISON WITH CONCURRENT WORK

RLCSD and the independently developed concurrent work CEPO (Heakl et al., 2026) both use correct–incorrect privileged conditioning to refine verifier-anchored advantages. Their single-negative contrastive signals share the same log-ratio form under identical teacher distributions and conditioning. RLCSD contributes a distinct empirical diagnosis of privilege-induced style drift, a reference-construction procedure addressing negative-hint variability and self-conditioning, and a selective two-path optimization objective. We detail these differences below.

Diagnosing privilege-induced style drift. CEPO motivates contrastive evidence through limitations of one-sided credit assignment, including the difficulty of distinguishing decisive tokens from filler. RLCSD specifically identifies and quantifies how privileged conditioning shifts supervision toward stylistic tokens. Under our token partition, the mean absolute one-sided signal on style tokens is approximately three times that on task tokens (Figure 3). Contrastive conditioning substantially reduces this imbalance (Figure 4). We further examine response length, entropy, reward, and validation accuracy throughout training, and show that introducing contrast alleviates observed instability or response-length shrinkage in multiple methods (Figure 5, Figure 6). These analyses connect the token-level signal to training behavior and provide an empirical basis for addressing stylistic mismatch in self-distillation.

Robust reference construction. CEPO constructs its negative branch from a single rejected reference. RLCSD addresses two additional sources of signal distortion. First, it marginalizes teacher probabilities over K incorrect same-query rollouts, reducing dependence on an individual negative hint and its particular error mode. Second, it excludes the target rollout from rollout-derived hint pools, preventing direct self-conditioning from producing excessively confident teacher predictions. These choices are supported by controlled ablations: replacing multiple negative hints with one reduces math/K&K averages by 1.6/3.8 points, while allowing the target rollout into its own hint pool reduces them by 1.6/4.5 points (Table 3). The reference-construction procedure therefore contributes materially to the effectiveness of RLCSD.

Selective modulation with independent loss normalization. CEPO uses a clipped exponential evidence weight to multiplicatively modulate the outcome advantage. RLCSD instead applies bounded additive modulation, $r_t = \lambda \tanh(e_{\mathrm{ctr},t}/\tau)$, followed by a sign-preserving clamp. It selects tokens with sufficiently strong contrastive signals and separately normalizes the selected and remaining token sets. This two-path objective controls the contribution of selected tokens independently of their prevalence, while retaining outcome-based supervision on the remaining positions. Replacing it with a global token average reduces math/K&K performance by 2.0/4.8 points (Table 3). Thus, RLCSD differs not only in reference construction but also in how contrastive evidence is allocated within the optimization objective.

Evidence beyond the complete training recipe. RLCSD also evaluates contrastive conditioning as a component of existing methods. Applying it to OPSD and RLSD improves their mathematical reasoning averages by 1.0 and 1.4 points, respectively, with smaller gains on K&K (Table 2). The accompanying training curves show reduced entropy growth in OPSD and mitigated response-length shrinkage in RLSD (Figure 6). These experiments extend the evidence beyond RLCSD's particular objective and support the usefulness of contrastive conditioning across different self-distillation formulations. Additional cross-model OPD experiments examine the broader relevance of task-versus-style discrepancies when privileged hints are absent.

Taken together, RLCSD contributes an empirically grounded account of privilege-induced style drift and a training recipe that addresses both reference-dependent signal distortion and the allocation

of token-level supervision. Its contribution encompasses the diagnosis, construction, and use of contrastive signals, beyond the shared positive–negative log-ratio principle.

C Additional Analysis of Reference-Hint Selection

This appendix expands on the two refinements introduced in Section 3.3: K-marginalization of negative hints (Appendix C.1) and excluding the target rollout from the hint pool (Appendix C.2). Together they detail why the vanilla contrastive signal is unreliable and how each refinement restores a clean, verifier-aligned signal.

Vanilla contrastive signal. Given a correct privileged context y_c^* and a single incorrect context y_w^*, the vanilla contrastive signal is

$$e_{\text{ctr},t}^{(\text{vanilla})} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \pi_T(y_t \mid x, y_w^*, y_{<t}). \tag{12}$$

Although symmetric conditioning suppresses stylistic effects shared between the two branches, this one-to-one contrast remains sensitive to the sampled negative hint and to whether the target rollout is selected as its own privileged context. The following analyses characterize these two problems.

C.1 Negative-Hint Selection: Case Analysis and K-Marginalization

Why a single negative hint is unreliable. Correct solutions tend to converge to a small set of canonical strategies, whereas incorrect solutions diverge across many distinct error modes. Whether the optimization direction induced by the vanilla contrastive signal aligns with the verifier therefore depends on whether the target rollout y and the sampled negative hint y_w^* share a comparable error type. Table 4 enumerates the three resulting cases:

- **Correct y.** The teacher endorses each token in y under y_c^* (driving $e_{c,t}$ up) and conflicts under y_w^* (driving $e_{w,t}$ down), giving a strongly positive $e_{\text{ctr},t}$ that matches the verifier.
- **Incorrect y with similar-error y_w^*.** Symmetrically, $e_{c,t}$ drops and $e_{w,t}$ rises, giving a strongly negative $e_{\text{ctr},t}$ that again matches the verifier.
- **Incorrect y with dissimilar-error y_w^*.** $e_{c,t}$ still drops, but because y_w^* commits a different mistake from y, the teacher does not endorse y's tokens under it either; $e_{w,t}$ loses its expected positive sign, and $e_{\text{ctr},t}$ can carry either sign.

The third case is the failure mode: a substantial fraction of incorrect rollouts end up with their tokens overall encouraged rather than penalized, injecting substantial noise into the optimization. Actively selecting an error-matched negative hint via online LLM-as-judge scoring is prohibitively expensive at RL-training scale, so we instead marginalize over up to K available negative hints, taking their probability mean as the negative branch. This spreads the teacher's mass across the error modes of $\mathcal{G}^-$ rather than betting on a single one.

Table 4: Behavior of the naive contrastive signal $e_{\text{ctr},t}^{(\text{Naive})}$ across three sampling cases. Arrows indicate the sign of each quantity: ↑ positive, ↓ negative. The first two cases yield a $e_{\text{ctr},t}$ that aligns with the verifier (✓), but the third case is the failure mode (✗): when y_w^* does not share y's error type, the negative branch $e_{w,t}$ no longer reliably endorses y's tokens, and $e_{\text{ctr},t}$ loses its directional anchor.

y	y_w^* relative to y	$e_{c,t}$	$e_{w,t}$	$e_{\text{ctr},t}$	
correct	—	↑	↓	↑	✓
incorrect	similar error	↓	↑	↓	✓
incorrect	dissimilar error	↓	?	?	✗

Empirical validation. Figure 7 confirms the analysis through rollout-level means of $e_{\text{ctr},t}$. Correct rollouts produce a cleanly positive signal (left, frac > 0 = 0.989), validating Case 1. To isolate Case 3, the middle panel uses the least-similar negative hint per incorrect y (selected offline by an LLM judge): the signal is centered at zero, with roughly half the rollouts bearing the wrong sign. K-marginalization (right) restores a cleanly negative-skewed signal, reducing the wrong-sign rate from 51% to under 8%.

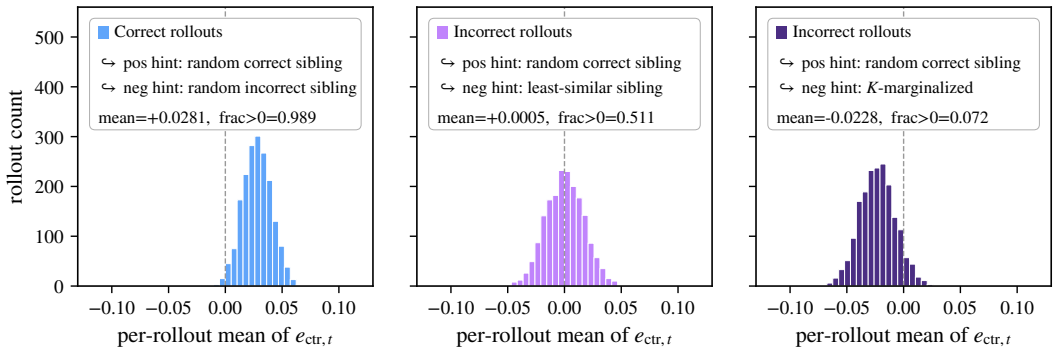


Figure 7: Rollout-level mean of $e_{\mathrm{ctr},t}$, computed with one positive hint and one or more negative hints. The positive hint is a randomly sampled correct sibling in all three panels; the panels differ in the rollout type and the negative-hint strategy. **Left:** correct rollouts with a single randomly sampled negative hint, illustrating the expected positive signal of Case 1. **Middle:** incorrect rollouts with a single LLM-judged least-similar negative hint (the worst case, Case 3). **Right:** incorrect rollouts with K-marginalized negative hints.

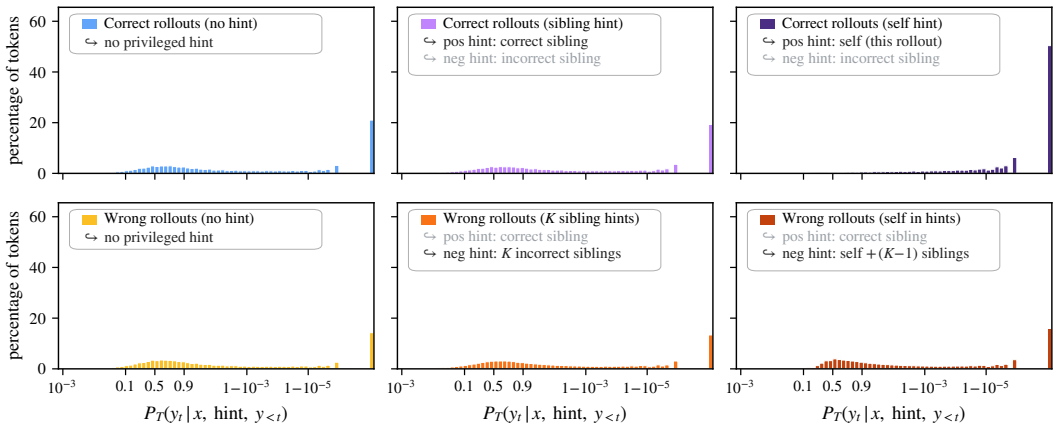


Figure 8: Distribution of teacher token probabilities $P_T(y_t \mid x, \text{hint}, y_{<t})$ under different hint sources, for correct (top) and incorrect (bottom) rollouts. Using a sibling hint leaves the distribution close to the no-hint case, whereas including the rollout itself as the hint shifts mass toward over-confident, high-probability tokens, motivating its exclusion from the hint pool.

C.2 Self-Referential Collapse and Excluding the Target Rollout

Self-conditioning induces over-confidence. Whenever a privileged hint is sampled from student rollouts, the target rollout y itself may be selected as its own hint. Conditioning the teacher on the exact response being evaluated induces extreme over-confidence, making the resulting signal overly sharp and obscuring the token-level distinctions that $e_{\mathrm{ctr},t}$ is intended to capture.

To characterize this effect, we compare conditioning on the target rollout itself with conditioning on another rollout for the same query, considering both correct and incorrect targets. As shown in Figure 8, for correct rollouts, a sibling hint leaves the overall distribution largely unchanged, whereas using the target itself as the hint raises the fraction of high-probability tokens ($P_T > 1 - 10^{-5}$) by roughly 30%. For incorrect rollouts, K-marginalization mitigates this effect, but including the target among the K candidates still increases the high-probability fraction by about 10%.

We therefore exclude the target rollout y from all rollout-derived hint pools. In our setting, this means drawing negative hints from $\mathcal{G}^- \setminus \{y\}$. More generally, if correct student rollouts are used as positive hints, the same exclusion should apply: positive hints should be drawn from $\mathcal{G}^+ \setminus \{y\}$.

D Implementation Details

This section presents the full implementation details that were not fully described in Section 4.1 of the main text.

D.1 Privileged Prompt Template

Teacher Privileged Prompt Template

```
Problem: {problem}
Here is a reference solution to this problem:
=== Reference Solution Begin ===
{reference solution text}
Correct final answer: {extracted answer}
=== Reference Solution End ===
After reading the reference solution above, make sure you
understand the reasoning behind each step. Please reason step
by step, and put your final answer within \boxed{}.
```

The same template is used for both branches. In the positive branch, `{extracted answer}` is the answer associated with the correct reference. In the negative branch, it is the answer extracted from the incorrect rollout, rather than the ground-truth answer. The wording `Correct final answer:` is retained in both branches to keep the template identical.

D.2 Shared training setup

All methods are implemented on top of the veRL (Sheng et al., 2024) PPO trainer with FSDP2 actor sharding and vLLM rollouts. All runs use full-parameter training (no LoRA) on 8×H20 GPUs. The settings in Table 5 are identical across GRPO, OPSD, SDPO, SRPO, RLSD, and RLCSD within a given (task, backbone) configuration. Only the per-step loss differs across methods.

Table 5: Hyperparameters shared by all six methods.

Group	Hyperparameter	Value
Optimization	Optimizer	AdamW
	Weight decay	0.01
	LR warmup (linear)	50 steps
	LR	1×10^{-6}
	GPUs	8×H20
Batching	Per-device batch	8
	Group size G	8
	PPO mini-batch	16
	Rollout IS mode	token-level
	Rollout IS threshold	2.0
Rollout sampling	Temperature	1.0
	Top-p / Top-k	0.95 / 20
	Max prompt length	2,048
	Max completion length	16,384
Evaluation sampling	Temperature	0.6
	Top-p / Top-k	0.95 / 20
	Max completion length	38,912
	Eval batch size	16
	Samples per query	12 (math, mean@12) / 1 (K&K, pass@1)

Table 6: Method-specific hyperparameters.

Method	Hyperparameter	Value	Description
GRPO	KL-to-ref coef.	10^{-3}	coef. on $\mathrm{KL}(\pi_\theta \| \pi_{\text{ref}})$.
	PPO clip ϵ	0.2	PPO-clip range.
OPSD	KL-to-ref coef.	0	disabled.
	JSD mixing β	0.0	0 = forward KL, 1 = reverse KL.
	Vocab clip	0.05	per-(token, vocab) divergence cap.
	Full-logit distill	yes	dense top-k + tail-bucket target.
	Top-k	100	vocab indices per token.
	Tail bucket	yes	residual mass outside top-k.
	Teacher mode	fixed	frozen pretrained checkpoint.
SDPO / SRPO	KL-to-ref coef.	0	disabled.
	Mixing coef. α	0.5	student/teacher mix in JSD target.
	Full-logit distill	yes	dense top-k + tail-bucket.
	Top-k	100	vocab indices per token.
	PPO clip ratio	2.0	IS-ratio clip on distill term.
	EMA decay	0.95	EMA self-teacher decay.
	Teacher mode	EMA	EMA of policy weights.
RLSD	KL-to-ref coef.	0	disabled.
	PPO clip ϵ	0.2	outer PPO-clip on $\pi_\theta/\pi_{\text{old}}$.
	Evidence clamp ϵ_w	0.2	clamp on $w_t = \exp(\mathrm{sign}(A)\,(\log \pi_T - \log \pi_S))$.
	Modulator λ	0.5	λ=0: GRPO; λ=1: full modulation.
	λ schedule	$0.5 \to 0$ in 50 steps	linear decay.
	Teacher mode	snapshot (τ_{sync}=10)	hard copy every 10 steps.
RLCSD (ours)	KL-to-ref coef.	0	disabled.
	PPO clip ϵ	0.2	both paths of Eq. 10.
	Slope τ	1.3	$\tanh$ temperature, Eq. 7.
	Scale λ	0.5	overall scale of r_t, Eq. 7.
	Mask threshold δ	0.02	$m_t = \mathbb{1}[\lvert r_t \rvert > \delta]$, Eq. **??**.
	Two-path weight η	0.5	modulated-path weight, Eq. 10.
	Max. neg. hints K	4	up to K negative siblings per target, Eq. **??**.
	Teacher mode	fixed	

D.3 Method-specific hyperparameters

Table 6 lists the hyperparameters that vary across methods. We follow the *original papers* wherever values are specified there.

E Additional Details of the Contrastive Plug-In

We apply the contrastive construction to two families of self-distillation methods while leaving their remaining optimization procedures unchanged.

Advantage-modulation methods. For RLSD, the one-sided sampled-token signal is replaced by

$$e_{\text{ctr},t} = \log \pi_T(y_t \mid x, y_c^*, y_{<t}) - \log \left[\frac{1}{K} \sum_{k=1}^{K} \pi_T(y_t \mid x, y_{w,k}^*, y_{<t}) \right]. \tag{13}$$

The resulting signal is then passed to the original method-specific advantage-modulation procedure.

Dense-distribution OPSD. For OPSD, we apply the contrast at the distribution level using a classifier-free-guidance formulation (Ho & Salimans, 2022). Let ℓ_c and ℓ_w denote the teacher logits under the correct and incorrect hints. We replace the original teacher target with

$$q_{\text{target}} = \mathrm{softmax}\left((1+\alpha)\ell_c - \alpha \ell_w\right), \tag{14}$$

where $\alpha \geq 0$ controls the guidance strength and $\alpha = 0$ recovers vanilla OPSD.

We additionally use the disagreement between the two teacher distributions to construct a soft token mask:

$$D_t = \mathrm{KL}\left(\pi_T(\cdot \mid x, y_w^*, y_{<t}) \,\middle\|\, \pi_T(\cdot \mid x, y_c^*, y_{<t})\right), \qquad w_t = \frac{D_t}{D_t + \tau}. \tag{15}$$

The resulting objective is

$$\mathcal{L}_{\mathrm{OPSD+Ctr}} = \sum_t w_t \; \mathrm{KL}\left(q_{\mathrm{target},t} \,\middle\|\, \pi_\theta(\cdot \mid x, y_{<t})\right). \tag{16}$$

Setting $\tau = 0$ disables the mask.

F Broader Insights for Cross-Model On-Policy Distillation

The task-versus-style decomposition identified in on-policy self-distillation may also apply to cross-model OPD. In standard OPD, the teacher–student distribution gap can reflect both genuine capability differences and mismatches in discourse, formatting, and reasoning style. If the stylistic component dominates, the optimization may transfer the teacher's register rather than its task knowledge. This perspective is consistent with the observation of Li et al. (2026b) that a stronger teacher can be less effective than a weaker but stylistically aligned one.

F.1 Experimental Setup

We follow the teacher–student configuration of Li et al. (2026b). The student is Qwen3-1.7B-Base, and we compare two teachers with similar mathematical performance but substantially different stylistic distances from the student:

- **Qwen3-4B-Instruct**, a heavily post-trained instruction model used in non-thinking mode, which is stylistically distant from the base-model student;
- **Qwen3-4B-Base-GRPO**[*], a mathematical reasoning model obtained by applying zero-RL GRPO to Qwen3-4B-Base, which remains closer to the student in model family and generation style.

We sample 200 prompts from DeepMath (He et al., 2025) and generate two student rollouts for each prompt. Along each student trajectory, we compute the per-token reverse KL divergence

$$\mathrm{KL}_t = \mathrm{KL}\big(\pi_S(\cdot \mid x, y_{<t}) \,\big\|\, \pi_T(\cdot \mid x, y_{<t})\big), \tag{17}$$

where π_S and π_T denote the student and teacher distributions, respectively. We then inspect the response positions with the largest KL_t under each teacher.

F.2 Which Tokens Dominate the OPD Signal?

Table 7 reports the 15 response tokens receiving the largest per-token KL values for each teacher. The two teachers produce qualitatively different signals. For Qwen3-4B-Instruct, the largest KL values are concentrated on discourse and structural tokens, including connectives such as "The," "If," "Thus," and "Therefore," as well as paragraph breaks and LaTeX delimiters. These tokens primarily reflect differences in how the teacher expresses and structures its reasoning.

In contrast, the signal from Qwen3-4B-Base-GRPO is more concentrated on digits, operators, mathematical delimiters, and content-bearing tokens such as "subsets" and "ex." Its distribution gap therefore provides more direct supervision about mathematical content rather than surface style.

F.3 Effect on Distillation Performance

We further train Qwen3-1.7B-Base through OPD using each teacher. Both runs use DeepMath as the training set and share the same training configuration and budget of 300 steps. Table 8 compares the standalone teacher performance with the resulting distilled students.

[*] https://huggingface.co/lllyx/Qwen3-4B-Base-GRPO

Table 7: Response tokens with the largest per-token reverse KL values on Qwen3-1.7B-Base rollouts from DeepMath. The instruct teacher's signal is dominated by discourse and structural tokens, whereas the aligned GRPO teacher places more weight on mathematical content.

	Qwen3-4B-Instruct			**Qwen3-4B-Base-GRPO**	
#	Token	KL	#	Token	KL
1	`<|endoftext|>`	29.2	1	`3`	11.6
2	` The`	25.3	2	` subsets`	10.4
3	` $\`	23.5	3	` \\`	9.6
4	`\n\n`	23.0	4	`,`	9.4
5	` five`	21.9	5	`}}`	8.7
6	` If`	21.2	6	`!)`	8.7
7	` This`	21.0	7	`M`	8.0
8	`Done`	18.6	8	`ex`	8.0
9	` Thus`	17.4	9	`(\\)`	7.5
10	`qx`	15.6	10	` x`	7.4
11	` ¨`	15.6	11	`=(`	7.4
12	` \(`	15.1	12	`1`	7.3
13	`Name`	14.7	13	`}`	7.3
14	` Therefore`	14.2	14	`c`	7.0
15	`Reach`	14.1	15	`{`	6.9

Table 8: Cross-model OPD with teachers at different stylistic distances from Qwen3-1.7B-Base. Both runs use DeepMath and the same 300-step training configuration. Despite its lower standalone accuracy, the stylistically aligned GRPO teacher produces the stronger distilled student. All results are mean@12 (%).

	Teacher standalone				**Distilled student**			
Teacher	AMC23	AIME24	AIME25	Avg.	AMC23	AIME24	AIME25	Avg.
Qwen3-4B-Instruct	**60.8**	**24.6**	**19.2**	**34.9**	32.9	9.0	5.2	15.7
Qwen3-4B-Base-GRPO	56.1	22.3	19.0	32.5	**39.7**	**13.3**	**9.2**	**20.7**
Base student (Qwen3-1.7B-Base)		–			12.0	1.5	1.7	5.1

Although Qwen3-4B-Instruct is the stronger standalone teacher, achieving an average math score of 34.9 compared with 32.5 for Qwen3-4B-Base-GRPO, it produces a substantially weaker distilled student. The student distilled from the instruct teacher reaches only 15.7 average accuracy, whereas the student distilled from the stylistically aligned GRPO teacher reaches 20.7 and performs better on all three benchmarks.

F.4 Implications

These results suggest that teacher capability alone does not determine the effectiveness of cross-model OPD. The instruct teacher is stronger in standalone evaluation, but much of its teacher–student divergence is concentrated on discourse and formatting tokens, and it produces the weaker distilled student. The stylistically aligned GRPO teacher instead provides more task-bearing supervision and yields better downstream performance.

Because standard cross-model OPD does not condition the teacher on a privileged reference, RLCSD's contrastive cancellation cannot be applied directly. Nevertheless, the distribution of per-token KL offers a simple, training-free diagnostic for teacher selection: an effective teacher should not only be capable, but should also concentrate its supervision on tokens associated with the target capability rather than on stylistic mismatch.

G Vocabulary Partitioning for Task / Style Statistics

This section specifies the token partition of math reasoning tasks used in Figures 3 and 4.

For each decoded token, we first strip the leading-space marker used by the Qwen3 tokenizer and lowercase the result. The token is then sent through the following rules in order, returning at the first match:

(1) empty or whitespace-only → **style**;

(2) matches any of the math regexes (a digit `\d`; an arithmetic operator in `+ − = * / < > × ÷ ≤ ≥ ≠`; a LaTeX command `\[A-Za-z]+`; a double backslash; or one of `$ ^ _`) → **task**;

(3) the normalized form is in the math wordlist {`mod, prime, factor, gcd, lcm, log, ln, sin, cos, tan, exp, integral, sqrt, boxed, frac, sum, prod, pi, alpha, beta, gamma, theta, delta, lambda, mu, sigma, infty, leq, geq, neq, cdot, times, div`} → **task**;

(4) pure punctuation or a literal newline token (`\n`, `\\n`) → **style**;

(5) the normalized form is in the discourse wordlist (connectives `therefore, so, thus, hence, then, because, since`; hedges `wait, maybe, perhaps, seems, okay, ok, well, now, first, next, finally, actually, alternatively, however`; scaffolding `step, answer, let, lets`; closed-class function words `is, are, us, we, the, a, an, of, to, for, in, on, by, at, as, and, or, but, if, yes, no, this, that, these, those, it, its, be, been, being, have, has, had, do, does, did, will, would, should, could, can, may`) → **style**;

(6) otherwise → **neutral**.

H Computational Cost Analysis

RLCSD introduces additional teacher-side computation compared with one-sided self-distillation methods, since it evaluates the teacher on both a correct hint and multiple wrong hints to form the contrastive signal. This extra cost is reflected in the `teacher log prob` time in Table 9: RLCSD spends 14.210s per step on teacher scoring, compared with about 9s for one-sided methods such as RLSD and OPSD.

However, this overhead is small compared with the dominant cost of autoregressive rollout generation. RLCSD spends 473.89s per step on `gen`, so the extra teacher scoring is only a minor fraction of the generation cost. In other words, although RLCSD requires more teacher forwards, these forward-only evaluations add relatively little wall-clock overhead compared with sampling long reasoning trajectories.

More importantly, the end-to-end step time remains favorable. As shown in Table 9, RLCSD is the *third fastest* method overall, behind only GRPO and RLSD. In particular, it is faster than dense-distillation baselines such as OPSD, SDPO, and SRPO. A likely reason is that these methods perform dense token-level distillation over the vocabulary during training, which makes the optimization stage itself much more expensive than the sampled-token modulation used in RLCSD and RLSD.

Table 9: Average wall-clock time per training step (seconds) on math reasoning task, using Qwen3-4B as the base model. RLCSD spends more time on `teacher_log_prob` because it uses additional wrong-hint teacher forwards, but this overhead is small relative to rollout generation (`gen`). In total step time, RLCSD remains the third fastest method overall.

Method	Gen (s/step)	Teacher log prob (s/step)	Total step (s/step)
GRPO	503.78	–	759.01
OPSD	413.92	9.552	1035.54
SDPO	495.64	9.701	961.14
SRPO	411.01	9.566	1011.51
RLSD	412.27	9.485	812.84
RLCSD	473.89	14.210	891.55

Table 10: Results on agentic tasks using Qwen3-4B. ALFWorld metrics are success rates (%). WebShop Score is the average reward on a scale of $[0, 1]$, and Success is the percentage of episodes receiving a reward of 1. Higher is better for all metrics. **Bold** marks the best result in each column, and underline marks the second-best result.

	ALFWorld			WebShop	
Method	Overall (%)	Seen (%)	Unseen (%)	Score	Success (%)
Base	43.06	45.00	41.04	0.5081	18.4
GRPO	47.44	48.57	46.26	0.5170	18.8
OPSD	46.35	42.85	**50.00**	0.5164	18.5
SDPO	46.35	**50.71**	41.79	0.5225	18.6
SRPO	45.98	46.42	45.52	0.5194	18.7
RLSD	45.62	43.57	47.76	0.5218	19.1
RLCSD	**48.17**	49.28	47.01	**0.5300**	**19.4**

I Additional Results on Agentic Tasks

Settings. We evaluate Qwen3-4B on ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022). We allow at most 30 interaction steps per episode on ALFWorld and 20 on WebShop, where each step consists of an agent action and the corresponding environment response. For both benchmarks, we enable thinking mode and impose a trajectory budget of 16,384 tokens, counting both model outputs and subsequent environment observations. At evaluation time, we sample one trajectory per task with a temperature of 0.6; we use top-p of 1.0 without top-k filtering for ALFWorld, and top-p of 0.95 with top-k of 20 for WebShop. For ALFWorld, we report overall success rate and success rates on the seen and unseen splits. For WebShop, we report the average reward, which includes partial credit and ranges from 0 to 1, and the success rate, defined as the fraction of episodes receiving a reward of 1. All success rates are reported as percentages.

Results. As shown in Table 10, RLCSD achieves the highest overall success rate on ALFWorld and the best average reward and success rate on WebShop. On ALFWorld, RLCSD reaches 48.17% overall success, outperforming the base model by 5.11 percentage points and the strongest baseline, GRPO, by 0.73 percentage points. Although SDPO and OPSD achieve the highest success rates on the seen and unseen splits, respectively, RLCSD performs consistently across both splits, yielding the best overall result. On WebShop, RLCSD achieves an average reward of 0.5300 and a success rate of 19.4%, exceeding the strongest baselines for these metrics by 0.0075 and 0.3 percentage points, respectively. These results show consistent, albeit modest, improvements over the competing methods across both environments.

J Sensitivity Analysis of Hyperparameters in RLCSD

We investigate how RLCSD's hyperparameters affect token selection and training behavior using Qwen3-8B on DeepMath. The five hyperparameters are the number of negative hints K, the soft-normalization temperature τ, the modulation scale λ, the selection threshold δ, and the modulated-path weight η. Our default configuration is

$$(K, \tau, \lambda, \delta, \eta) = (4, 1.3, 0.5, 0.02, 0.5). \tag{18}$$

We first analyze their direct effects on fixed contrastive signals, then examine the default training dynamics and controlled comparisons in which one hyperparameter is varied at a time.

J.1 How the Hyperparameters Affect the Learning Signal

Token selection depends on a joint effective threshold. RLCSD transforms the raw contrastive signal into a bounded modulation and selects tokens according to

$$r_t = \lambda \tanh\left(\frac{e_{\mathrm{ctr},t}}{\tau}\right), \qquad m_t = \mathbf{1}(|r_t| > \delta). \tag{19}$$

Table 11: Analytical effects of changing one hyperparameter from the default configuration. The effective threshold h_{eff} and local gain λ/τ are calculated from the modulation formula. Selection trends assume fixed raw contrastive signals and describe non-strict changes relative to the default configuration.

Configuration	h_{eff}	Local gain λ/τ	Bound λ	Selection coverage
Default	0.0520	0.3846	0.50	—
$\tau = 0.65$	0.0260	0.7692	0.50	Increases
$\tau = 2.6$	0.1041	0.1923	0.50	Decreases
$\lambda = 0.25$	0.1042	0.1923	0.25	Decreases
$\lambda = 1.0$	0.0260	0.7692	1.00	Increases
$\delta = 0.01$	0.0260	0.3846	0.50	Increases
$\delta = 0.04$	0.1042	0.3846	0.50	Decreases
$\eta = 0.25$	0.0520	0.3846	0.50	Unchanged
$\eta = 1.0$	0.0520	0.3846	0.50	Unchanged

For $\tau > 0$ and $0 < \delta < \lambda$, the selection criterion is equivalent to

$$m_t = \mathbf{1}\left(|e_{\text{ctr},t}| > h_{\text{eff}}\right), \qquad h_{\text{eff}} = \tau \operatorname{arctanh}\left(\frac{\delta}{\lambda}\right). \tag{20}$$

Thus, τ, λ, and δ jointly determine selection coverage. On fixed raw signals, increasing τ or δ raises the effective threshold and selects fewer tokens, whereas increasing λ lowers it and selects more tokens. The default configuration corresponds to $h_{\text{eff}} \approx 0.0520$.

Table 11 summarizes these analytical relationships. Different configurations can produce identical selection masks while assigning different modulation magnitudes. For example, doubling λ and halving δ, considered separately from the same default configuration, produce the same effective threshold at fixed τ. However, only the former scales the modulation values.

Modulation strength differs from selection coverage. For small $|e_{\text{ctr},t}|/\tau$, the modulation is approximately linear:

$$r_t \approx \frac{\lambda}{\tau} e_{\text{ctr},t}. \tag{21}$$

The ratio λ/τ controls the local gain, while λ bounds the modulation magnitude. Increasing λ scales every modulation value proportionally. Decreasing τ also strengthens the modulation, but pushes more tokens toward saturation and compresses differences among large raw signals.

In contrast, changing δ leaves every r_t unchanged and affects only selection. A larger δ therefore retains fewer, stronger signals without changing the average modulation magnitude over all tokens. For λ and τ, changes in modulation magnitude and selected-set membership occur together, so the average magnitude among selected tokens need not follow the same trend as the all-token average. The sign-preserving clamp further limits the effective advantage change when a modulation would otherwise reverse the outcome-based update direction.

Negative-hint marginalization controls estimation variability. The parameter K sets the maximum number of negative hints. For a target rollout y, the effective number is $K_y = \min(K, |\mathcal{G}^- \setminus \{y\}|)$, and the contrastive signal is

$$e_{\text{ctr},t} = \log p_{c,t} - \log\left(\frac{1}{K_y}\sum_{k=1}^{K_y} p_{w,k,t}\right), \tag{22}$$

where $p_{c,t}$ and $p_{w,k,t}$ denote teacher probabilities under the correct and incorrect references, respectively. Using multiple negative hints can reduce sensitivity to individual references and improve coverage of the error modes represented in the hint pool. Increasing K changes the estimate only when additional candidates are available. It does not imply a monotonic change in the absolute contrastive signal or the selected-token ratio.

Table 12: Training dynamics of Qwen3-8B on DeepMath under the default RLCSD configuration. Each entry is the mean of the logged per-step metric over the indicated training window.

Metric	Steps 1–50	Steps 201–250	Steps 351–400
Selected tokens (%)	30.01	28.62	28.24
Response length	9423	9346	9617
Training reward	0.736	0.753	0.749
Actor entropy	0.321	0.201	0.183

Path weighting interacts with selection sparsity. For a rollout with nonempty unmodulated and modulated token sets, the objective is

$$J = \frac{1}{|\mathcal{U}|} \sum_{t \in \mathcal{U}} \psi_t(A_{\mathrm{ORM}}) + \frac{\eta}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \psi_t(\widetilde{A}_t). \tag{23}$$

Writing $q = |\mathcal{M}|/(|\mathcal{U}| + |\mathcal{M}|)$, the ratio of per-token normalization weights is

$$\frac{\eta/|\mathcal{M}|}{1/|\mathcal{U}|} = \eta \frac{1-q}{q}. \tag{24}$$

Selecting fewer tokens therefore concentrates the modulated-path weight on a smaller subset rather than proportionally reducing its nominal weight. At $\eta = 0.5$, this ratio is 2.0 for $q = 0.2$ and approximately 1.17 for $q = 0.3$, before accounting for advantage magnitudes and PPO clipping.

Unlike λ, η does not directly change the signal or the selection mask. It scales the entire modulated path, including the outcome-level anchor within $\widetilde{A}_t$. During training, however, changing η can indirectly alter subsequent selection statistics by changing the policy and its rollout distribution.

J.2 Training Dynamics under the Default Configuration

Table 12 reports the default configuration's training statistics across three windows. The selected-token ratio remains close to 28%–30%, decreasing from 30.01% over steps 1–50 to 28.24% over steps 351–400. Response length is maintained, changing from approximately $9{,}423$ to $9{,}617$ tokens, while training reward remains comparable across the middle and late stages. Actor entropy decreases from 0.321 to 0.183 without an accompanying collapse in response length. These observations characterize the default configuration as maintaining selective modulation throughout training while preserving extended responses.

J.3 Controlled Hyperparameter Comparisons

Evaluation protocol. We vary one hyperparameter at a time while keeping the remaining settings and training budget fixed. We report the selected-token ratio, response length, training reward, and actor entropy averaged over steps 351–400, together with AIME25 mean@12 evaluated at the fixed step-400 checkpoint for every configuration. Unlike the main results in Table 1, which use checkpoints selected by best validation performance, this comparison fixes the training step across configurations. Accordingly, the default configuration scores 69.72% here, whereas its validation-selected checkpoint scores 70.0% in Table 1.

Selection coverage follows the effective-threshold analysis. The changes in selected-token ratio agree with the predicted directions. Decreasing τ to 0.65, increasing λ to 1.0, or decreasing δ to 0.01 expands coverage to approximately 39%–40%. The opposite changes reduce coverage to approximately 18%–19%. Although these relationships are derived for fixed raw signals, their directions remain evident after training. Across these configurations, AIME25 ranges from 67.78% to 70.00%, indicating that substantial changes in selection coverage do not translate into proportional changes in validation performance.

Multiple negative hints improve performance. Increasing K from 1 to 2 and 4 improves AIME25 from 66.94% to 68.89% and 69.72%, respectively. Training reward also increases from 0.731 to 0.743 and 0.749. These results are consistent with the benefit of a more reliable negative-branch

Table 13: Hyperparameter sensitivity of RLCSD on Qwen3-8B and DeepMath. Each configuration changes one parameter from the default setting. Training statistics are averaged over steps 351–400; AIME25 mean@12 is evaluated at the fixed step-400 checkpoint for every configuration, rather than at the validation-selected checkpoints used in Table 1.

Configuration	Selected (%)	Length (tokens)	Reward	Entropy	AIME25 (%)
Default	28.24	9617	0.749	0.183	69.72
$K = 1$	32.67	9148	0.731	0.207	66.94
$K = 2$	29.83	9486	0.743	0.192	68.89
$\tau = 0.65$	39.56	10083	0.744	0.216	68.61
$\tau = 2.6$	18.72	9324	0.738	0.174	68.06
$\lambda = 0.25$	19.15	9408	0.741	0.178	68.33
$\lambda = 1.0$	40.38	10426	0.736	0.231	67.78
$\delta = 0.01$	38.91	9702	0.747	0.189	69.17
$\delta = 0.04$	18.46	9538	0.750	0.181	70.00
$\eta = 0.25$	27.63	9361	0.740	0.176	68.61
$\eta = 1.0$	29.47	10054	0.745	0.204	69.44

estimate. The incremental accuracy gain decreases from 1.95 points for $K = 1$ to $K = 2$ to 0.83 points for $K = 2$ to $K = 4$, supporting the use of multiple hints while suggesting diminishing gains within the tested range.

Stronger modulation does not necessarily improve accuracy. Increasing λ to 1.0 or decreasing τ to 0.65 produces longer responses than the default configuration, reaching $10{,}426$ and $10{,}083$ tokens on average, respectively. However, their AIME25 scores decrease to 67.78% and 68.61%. Weaker modulation also yields lower scores: $\lambda = 0.25$ and $\tau = 2.6$ achieve 68.33% and 68.06%. These comparisons support balancing modulation strength rather than maximizing it, and show that longer responses alone do not guarantee better reasoning performance.

Sparser selection can preserve performance. Increasing δ to 0.04 reduces the selected-token ratio from 28.24% to 18.46% while maintaining comparable AIME25 performance (70.00% versus 69.72%). Decreasing δ to 0.01 expands coverage to 38.91% but yields 69.17%. These results suggest that broader coverage is not always necessary: independent normalization allows a smaller selected set to retain substantial influence on optimization. The small accuracy difference between $\delta = 0.04$ and the default does not establish a clear preference between them.

Modulated-path weighting has a moderate effect in the tested range. Changing η to 0.25 or 1.0 yields AIME25 scores of 68.61% and 69.44%, respectively, compared with 69.72% for the default. The selected-token ratio remains relatively close to the default, ranging from 27.63% to 29.47%, consistent with η affecting selection only indirectly through training. These comparisons support $\eta = 0.5$ as a reasonable operating point, while the performance differences do not indicate that it is uniquely optimal.