RETHINKING BLACK-BOX ON-POLICY DISTILLATION: A RUBRIC-BASED ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) is a powerful paradigm for LLM post-training, yet its reliance on teacher logits restricts it to white-box teachers. In this paper, we study whether rubric-based supervision can provide a text-only feedback interface for black-box OPD, and what makes it effective. Our instantiation, Rubric-based On-Policy Distillation (ROPD), induces rubrics by contrasting teacher responses with the student's current rollouts and uses them to score these rollouts for on-policy optimization. We find that, under matched settings, text-only ROPD outperforms logit-based OPD methods on four competition-math benchmarks, reaching the best AIME24 accuracy of standard logit-based OPD with 9.6× fewer training samples. Our analysis yields three findings: rubric rewards track correctness far better than teacher log-likelihood (AUC 0.90 vs. 0.35); training with them repairs more failed criteria and regresses fewer passed ones than logit-based supervision; and contrasting teacher responses with student rollouts is central to rubric quality. These results position rubric-based OPD as a flexible, black-box-compatible alternative to logit-based OPD. Our implementation is available at `https://anonymous.4open.science/r/ROPD-8257`.

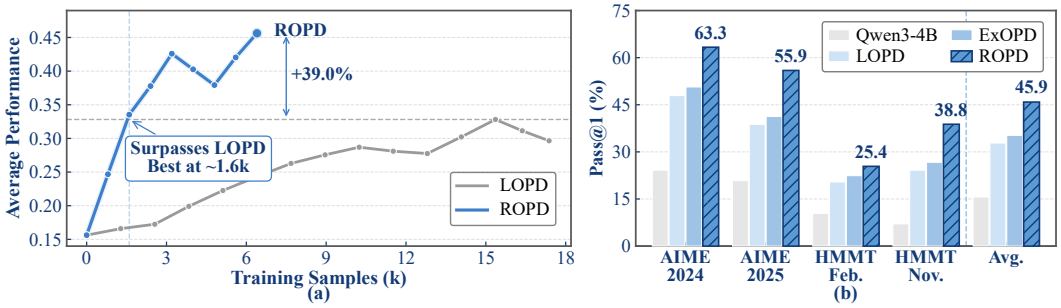


Figure 1: **ROPD efficiency and reasoning performance.** (a) Training dynamics averaged over four math benchmarks (AIME 24/25 and HMMT 25 Feb./Nov. (MAA, 2024; 2025; HMMT, 2025)). ROPD surpasses LOPD's best accuracy early in training (Section 3.3). (b) Comparative results under matched settings (DAPO-Math-17K training set, Qwen3-4B student, Qwen3-30B-A3B teacher (Yu et al., 2025; Yang et al., 2025)). See Section 3.1 for details.

1 INTRODUCTION

On-policy distillation (OPD) trains a student on feedback computed over its own rollouts, keeping the training signal aligned with the current policy (Agarwal et al., 2024; Lu & Lab, 2025; Song & Zheng, 2026). By using teacher logits as a dense signal (Gu et al., 2024), OPD has been widely adopted in recent open-source reasoning models (Yang et al., 2025; Xiao et al., 2026; DeepSeek-AI, 2026). However, this requires white-box teacher access; as frontier models increasingly expose only text through APIs, the set of usable OPD teachers is restricted.

Several lines of work explore text-only supervision for black-box distillation, including preference feedback (Singh et al., 2025), adversarial discriminators (Ye et al., 2026), LLM-as-a-Judge scoring (Shen et al., 2026), and fine-tuning on teacher demonstrations (Kim & Rush, 2016). Some of these

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.

methods, such as GAD (Ye et al., 2026) and OVD (Xiong et al., 2026), update the student on its own current rollouts, showing that on-policy distillation does not depend on access to teacher logits. The remaining limitation lies in the feedback: these methods compress each response into a single preference or score, which indicates how good a rollout is but not which properties it lacks. This raises our central question: *can text-only teacher feedback make black-box on-policy distillation competitive with logit-based OPD, and what makes it effective?*

Answering this question requires turning teacher responses into effective feedback for the current student. A black-box teacher's responses carry knowledge about problem-solving strategies and answer quality; useful feedback should be teacher-grounded, drawing on this knowledge, and student-conditioned, targeting gaps in the student's own rollouts. Rubrics offer a natural interface: they decompose quality into explicit, self-contained criteria that are evaluated separately and aggregated into a weighted reward. Because each criterion can encode a teacher-demonstrated property while being checked against the student's own response, rubrics connect teacher knowledge to current student gaps.

We investigate this question through **ROPD (Rubric-based On-Policy Distillation)**, a concrete instantiation of rubric-based supervision for black-box on-policy distillation. Applying rubrics directly in this setting faces three difficulties: (i) criteria derived only from teacher answers describe desirable properties without revealing what the current student lacks; (ii) criteria from a single teacher answer inherit path-specific choices and may penalize other valid solutions; (iii) rubrics constructed separately for each rollout introduce score differences unrelated to response quality, so rollouts of the same prompt should be judged on a common basis.

ROPD addresses these difficulties through *teacher–student contrastive rubric induction*. For each prompt, a *Rubricator* jointly examines multiple teacher responses and the current student rollout group, and induces one set of weighted, self-contained criteria shared by all rollouts. A *Verifier* evaluates each rollout against these criteria without response provenance, and GRPO (Shao et al., 2024) uses the aggregated verdicts as rewards to update the policy that generated the rollouts, preserving the on-policy loop under text-only teacher access. Unlike rubric-based RL, whose rubrics define the target quality of a task, ROPD derives criteria from the teacher–student contrast to transfer a teacher's competence to the current student (Section 5).

Under matched teacher, student, data, and training configurations, text-only ROPD outperforms logit-based OPD methods on all four math benchmarks and reaches the best AIME24 accuracy of standard logit-based OPD with $9.6\times$ fewer training samples. Our analysis further explains this result through three findings on reward quality, training dynamics, and rubric construction, which together indicate that rubric-based supervision can serve as an effective feedback interface for black-box OPD. To our knowledge, this is the first systematic study of rubric-based supervision for black-box on-policy distillation.

Our main contributions are as follows:

- **Rubrics as a feedback interface for black-box on-policy distillation.** We study whether rubric-based supervision can stand in for teacher logits under text-only access; under matched settings, ROPD outperforms logit-based OPD on all four math benchmarks.
- **Insight 1: rubric rewards track correctness, whereas teacher likelihood does not.** On a fixed rollout pool, rubric rewards separate correct from incorrect rollouts far better than teacher log-likelihood (AUC 0.90 vs. 0.35), consistent with teacher likelihood favoring fluent but flawed reasoning over correct but stylistically divergent solutions.
- **Insight 2: under matched training, rubric rewards stabilize criterion-level progress relative to logit-based supervision.** During training, ROPD repairs more failed criteria and regresses fewer previously passed criteria than LOPD, indicating that decomposed rewards support targeted progress without eroding acquired capabilities.
- **Insight 3: teacher–student contrast is central to rubric quality.** Controlled source and component ablations indicate that constructing rubrics from both teacher responses and current student rollouts is central to reward quality: multi-teacher coverage is the dominant factor, while shared rubrics and blind verification provide complementary gains in the evaluated setting.

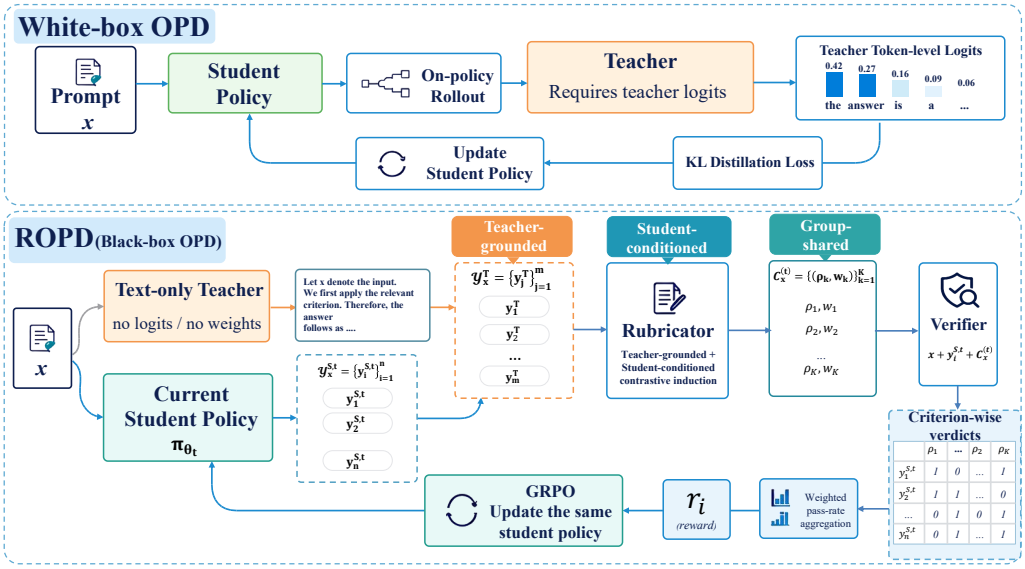


Figure 2: **White-box OPD vs. ROPD.** Top: White-box OPD uses teacher logits for token-level KL supervision. Bottom: ROPD constructs teacher-student contrastive rubrics from text-only access, converting criterion-wise verification into rewards. Both methods use on-policy student rollouts.

2 Rubric-based On-policy Distillation

ROPD instantiates rubric-based supervision for black-box on-policy distillation by making **rubric construction part of the feedback loop**. Teacher responses establish which solution properties are desirable, while the current student's rollouts reveal where those properties are missing. ROPD brings these two sources of evidence together to construct a shared evaluation standard, then converts criterion-wise judgments into rewards for updating the student. The resulting supervision is grounded in teacher competence and conditioned on the current policy's behavior.

Conditioning rubric induction on student rollouts makes current learning gaps visible; using multiple teacher responses provides a broader basis for identifying valid solution properties; and sharing the resulting rubric establishes a consistent basis for reward comparison. As illustrated in Figure 2, ROPD implements these requirements through a *Rubricator* for teacher–student contrastive rubric induction and a *Verifier* for evaluating individual responses against the induced criteria.

2.1 Black-box On-policy Distillation

On-policy distillation trains a student model using teacher-derived feedback on the student's own rollouts (Agarwal et al., 2024; Song & Zheng, 2026). We consider a black-box teacher π_T accessible only through text outputs. For a prompt x, the teacher provides a response set $\mathcal{Y}_x^T = \{y_j^T\}_{j=1}^m$, and the current student policy π_{θ_t} samples a rollout group $\mathcal{Y}_x^{S,t} = \{y_i^{S,t}\}_{i=1}^n$. Teacher token probabilities, hidden states, and model weights are unavailable.

This setting imposes two requirements on the supervision signal. It must recover useful guidance from teacher text, and it must evaluate responses generated by the *current student policy*. The latter distinguishes black-box on-policy distillation from supervised fine-tuning on teacher demonstrations (Kim & Rush, 2016) and preference optimization on fixed teacher-generated pairs (Singh et al., 2025), which do not necessarily involve the current student in feedback construction. Here, the student generates rollouts, receives teacher-grounded feedback on them, and is updated from it.

2.2 Teacher-grounded Contrastive Rubric Induction

The teacher response set provides evidence about desirable solution behavior, but does not by itself specify how to score the student's answers. Useful feedback therefore requires extracting criteria from teacher demonstrations and applying them to the student's own responses.

Recent rubric-based RL methods decompose response quality into reference-grounded criteria and aggregate criterion-level evaluations into rewards (Gunjal et al., 2025; Xu et al., 2026a). ROPD retains this reward structure but **conditions the criteria themselves on the teacher–student gap**: the current student's responses help construct the standard used to evaluate them.

A rubric derived only from teacher answers can describe a high-quality solution, yet remain insensitive to the student's current error distribution. This matters because group-relative optimization depends on reward differences among student rollouts: a criterion that every rollout passes, or every rollout fails, yields no learning signal.

For each prompt x at step t, the Rubricator jointly processes the teacher responses and student rollouts:

$$\mathcal{C}_x^{(t)} = \mathrm{Rubricator}\big(x, \mathcal{Y}_x^T, \mathcal{Y}_x^{S,t}\big) = \{(\rho_k, w_k)\}_{k=1}^K, \tag{1}$$

where each ρ_k is a self-contained textual criterion, and $w_k > 0$ reflects its importance in the current teacher–student gap. Each criterion must be evaluable given only the prompt and a single response.

This construction assigns complementary roles to the two response sets: *teacher answers ground the quality standards; student rollouts inform which shortcomings those standards should address*. The resulting rubric is therefore both *teacher-grounded* and *student-conditioned*. Its dependence on $\mathcal{Y}_x^{S,t}$ lets the criteria track the student's behavior as training proceeds.

Multiple teacher responses and a shared rubric. The Rubricator receives $m > 1$ teacher responses per prompt, so that criteria can capture solution properties shared across valid reasoning paths rather than choices specific to a single demonstration. All student rollouts for the same prompt are evaluated using the same $\mathcal{C}_x^{(t)}$, including both its criteria and weights. Sharing the rubric holds the evaluation standard fixed within the group. Section 4.3 ablates these design choices.

2.3 Criterion-wise Verification and On-policy Reward

The induced rubric specifies what should be evaluated; verification checks whether each response meets it. ROPD separates these roles to preserve a clear information boundary: **teacher–student comparison shapes the criteria, while each response is judged against the shared standard**.

Blind verification of explicit criteria. Because each criterion is self-contained, the Verifier requires only the prompt, the response, and the criterion. The Verifier sees neither teacher responses nor the source of a rollout; this *blind, criterion-wise evaluation* reduces source-dependent bias:

$$v_{i,k} = \mathrm{Verifier}\big(x,\ y_i^{S,t},\ \rho_k\big), \qquad v_{i,k} \in \{0, 1\}, \tag{2}$$

where $v_{i,k} = 1$ indicates that rollout $y_i^{S,t}$ satisfies criterion ρ_k.

From criterion satisfaction to comparable rewards. The binary verdicts are aggregated into a scalar reward through their weighted pass rate:

$$r_i = \frac{\sum_{k=1}^K w_k\, v_{i,k}}{\sum_{k=1}^K w_k + \epsilon}, \tag{3}$$

where ϵ is a small constant for numerical stability. The shared weights provide a common aggregation rule across rollouts, and the criterion-level verdicts keep each scalar reward interpretable.

Closing the on-policy feedback loop. Finally, GRPO (Shao et al., 2024) uses $\{r_i\}_{i=1}^n$ to update π_{θ_t}, the policy that generated the rollout group. The complete feedback loop is thus *current student rollouts → teacher–student contrastive rubric → criterion-wise rewards → student policy update*. In our experiments, the teacher model serves as both the Rubricator and the Verifier. The complete training algorithm is provided in Appendix I.

3 Main Results

3.1 Setup

Models. We use Qwen3-4B (Yang et al., 2025) as the student. In the black-box setting, GPT-5.2-chat-latest (OpenAI, 2025) is accessed through text-only API responses and is compared with T-Judge,

Table 1: Black-box distillation results (Pass@1%). Bold/underline: best/second-best.

	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	GPQA-D.	HealthBench	IFEval
GPT-5.2-chat (teacher)	80.83	67.08	43.75	57.50	78.66	92.82	94.37
Non-Thinking							
Qwen3-4B (student)	24.17	20.83	10.42	7.08	35.66	83.32	<u>85.21</u>
T-Judge	<u>62.50</u> +38.3	<u>56.64</u> +35.8	28.94 +18.5	<u>38.75</u> +31.7	<u>36.29</u> +0.63	<u>84.52</u> +1.20	84.40 -0.81
OVD (Xiong et al., 2026)	61.56 +37.4	55.71 +34.9	<u>29.11</u> +18.7	37.92 +30.8	35.74 +0.08	83.68 +0.36	84.23 -0.98
GAD (Ye et al., 2026)	27.52 +3.35	23.34 +2.51	12.84 +2.42	14.11 +7.03	36.02 +0.36	83.57 +0.25	85.12 -0.09
ROPD (ours)	**65.02** +40.9	**58.75** +37.9	**31.69** +21.3	**41.67** +34.6	**36.50** +0.84	**84.92** +1.60	**85.28** +0.07

Table 2: Logit-based OPD results (Pass@1%). Bold/underline: best/second-best.

	Access	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg
Qwen3-30B-A3B (teacher)	–	76.25	61.25	33.33	55.00	56.46
Qwen3-4B (student)	–	24.17	20.83	10.42	7.08	15.63
SFT	text	26.69 +2.52	22.50 +1.67	11.62 +1.20	8.33 +1.25	17.29 +1.66
LOPD (Agarwal et al., 2024; Lu & Lab, 2025)	logit	47.92 +23.8	38.75 +17.9	20.42 +10.0	24.17 +17.1	32.82 +17.2
ExOPD (Yang et al., 2026)	logit	<u>50.66</u> +26.5	<u>41.25</u> +20.4	<u>22.42</u> +12.0	<u>26.68</u> +19.6	<u>35.25</u> +19.6
ROPD (ours)	**text**	**63.33** +39.2	**55.93** +35.1	**25.40** +15.0	**38.80** +31.7	**45.87** +30.2

OVD, and GAD. In the matched white-box setting, Qwen3-30B-A3B (Yang et al., 2025) serves as the open-weight teacher and is compared with SFT, the logit-based OPD baseline (hereafter LOPD), and ExOPD; LOPD and ExOPD use teacher logits, while ROPD receives teacher text only. Baseline details are in Appendix E.

Data. We use DAPO-Math-17K (Yu et al., 2025) for mathematics and RaR-Science/Medical-20K (Gunjal et al., 2025) for science and medical tracks. All methods share the same training samples within each domain. SFT uses static teacher responses. **Training.** All RL methods share the same GRPO recipe: learning rate 10^{-6}, batch size 32, $n = 8$ rollouts per prompt, and one epoch. ROPD uses $m = 4$ teacher responses and $K \in [4, 12]$ rubric items. Checkpoints are selected using the validation split of the corresponding training dataset. Full settings are listed in Appendix D.

Evaluation. We evaluate on AIME24/25 (MAA, 2024; 2025), HMMT25 (HMMT, 2025), GPQA-Diamond (Rein et al., 2023), HealthBench (Arora et al., 2025), and IFEval (Zhou et al., 2023). We sample $k = 16$ responses per problem with temperature 1.0, top-p 0.95, and a 32,768-token cap. Teacher evaluation uses the same protocol; full details are provided in Appendix D.

3.2 Performance in Black-Box Scenarios

Table 1 reports Pass@1 across seven benchmark configurations. ROPD ranks first in every configuration and obtains its largest gain on HMMT25 (Nov.), improving the base student from 7.08 to 41.67 (+34.6). The small IFEval gain indicates that this improvement does not reduce the evaluated instruction-following behavior; the corresponding thinking-mode results are reported in Table 8.

3.3 Matched White-Box Comparison

Table 2 reports the matched comparison with Qwen3-30B-A3B. Although ROPD uses teacher text only, it outperforms LOPD and ExOPD on all four mathematics benchmarks, closing 74.1% of the student–teacher gap on average compared with 42.1% for LOPD. Section 4.2 investigates why text-only rubric supervision can outperform dense logit access. ROPD is also more sample-efficient in this matched setting: it reaches LOPD's best AIME24 accuracy with 1.6k training samples versus 15.4k, and in 5.5h versus 34.4h, despite the added Rubricator and Verifier calls per step (Appendix G reports full curves, post-saturation behavior, and API cost).

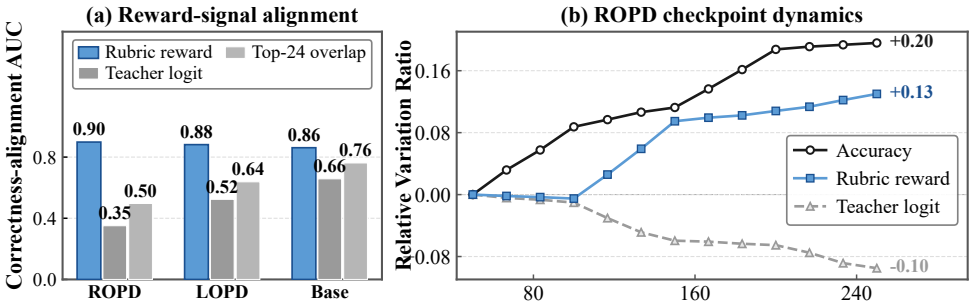


Figure 3: Reward signal alignment with correctness (AIME24). (a) Correctness-alignment AUC for rubric reward, teacher logit, and top-24 overlap. (b) Training trajectories: ROPD accuracy and rubric reward scale synchronously, while teacher logit diverges downward.

4 Analysis

Having established ROPD's empirical effectiveness, we now examine the properties of its reward signal. We begin with a concrete case illustrating how rubric-based rewards differ from scalar judges in distinguishing rollout quality (Section 4.1). We then quantify the alignment between reward signals and ground-truth correctness on a fixed rollout pool (Section 4.2). Finally, we ablate the core design choices to assess the contribution of each reward component (Section 4.3).

4.1 Case Study: Rubric *vs.* Scalar Judge

Before turning to large-scale quantitative analysis, we use a single example to build intuition for how rubric-based and scalar rewards differ in discriminating rollout quality. Table 3 presents a representative case regarding the parity-based contradiction: $n^3 + 3n^2 + 2n + 1 \equiv 0 \pmod{2024}$. Since $n(n+1)(n+2)$ is inherently even, the expression remains odd, precluding any solution for the even modulus 2024.

We compare two student rollouts: Rollout A (partially correct, missing general parity proof) and Rollout C (fabricated derivation guessing 337). The rubric separates them sharply (0.77 vs. 0.23, $\Delta = 0.54$), while the scalar judge barely distinguishes them (0.70 vs. 0.55, $\Delta = 0.15$).

This $3.6\times$ wider margin reflects a structural difference: scalar judges compress quality dimensions into a single value, allowing formatting to dilute logical failure. The rubric decouples evaluation dimensions (factorization, coherence, accuracy), preventing fabricated derivations from hiding behind structured prose.

The generality of this pattern is examined in Section 4.2 and the Appendix.

Key Insight

Criterion-wise rubrics preserve correctness-relevant distinctions that scalar scores can compress. In this case, they separate a partially correct rollout from a fluent but fabricated one by a $3.6\times$ wider margin than scalar judging.

4.2 Text-only Rubric Signals vs. Logit-based Supervision

Table 2 shows that text-only ROPD outperforms logit-based LOPD by +13.1 points on average. A natural question is whether this gap reflects differences in the quality of the underlying reward signal. We investigate this through a controlled analysis on a fixed pool of AIME24 rollouts, comparing three signals: (1) rubric rewards, (2) teacher log-likelihood, and (3) top-24 token overlap, each evaluated against ground-truth correctness. Full protocol details are provided in Appendix G.

Logit Likelihood is Poorly Aligned with Correctness. While LOPD treats teacher likelihood as a quality proxy, Figure 3 (a) reveals that this signal is negatively associated with correctness: rubric rewards achieve an AUC of 0.90, whereas teacher log-likelihood yields an AUC of 0.35, below the 0.50 random baseline. This pattern suggests that high teacher likelihood can favor fluent but logically flawed reasoning paths over correct but stylistically divergent ones. As shown in Figure 4 (b), ROPD produces more discriminative advantage signals across the majority of prompts. These observations

Table 3: Case study: Multi-dimensional rubric evaluation on an AIME-style number theory problem. We present five rubrics alongside blind Verifier verdicts (✓/×) for two representative rollouts (A and C) selected from a group of eight. Weights $w_k \in [1, 5]$ are dynamically assigned by the Rubricator.

ID	Category	Rubric	w_k	Rollout A	Rollout C
C1	Task Completion	Produces an explicit final answer.	5	✓	✓
C2	Observable Quality	Identifies the parity obstruction ($P(n)$ odd, 2024 even $\rightarrow$ no solution).	5	×	×
C3	Observable Quality	Correctly factorizes $n^3 + 3n^2 + 2n$ into $n(n+1)(n+2)$.	4	✓	×
C4	General Reasoning	Argument is logically coherent, each step follows from the last.	5	✓	×
C5	Observable Quality	No hallucinated numerical claims or guessed answers.	3	✓	×
Rubric Weighted Pass Rate $\left(\sum_k w_k v_{i,k} / \sum_k w_k\right)$				$17/22$=0.77	$5/22$=0.23
Scalar Score				0.70	0.55

are consistent with the rubric reward providing the optimizer with more correctness-aligned group-relative feedback than teacher log-likelihood.

Early Convergence Followed by Divergence in Token-level Similarity. The training trajectories reveal a two-phase pattern in how ROPD relates to teacher token distributions. Figure 4 (a) shows that in the earliest stages, ROPD's token overlap increases even faster than LOPD's, suggesting that the rubric reward initially encourages alignment with the teacher's formatting and linguistic conventions. However, as Figures 4 (a) and 3 (b) show, a divergence follows: LOPD continues to optimize its logit-based objective throughout training, while ROPD's accuracy and rubric rewards increase as its token-level similarity to the teacher declines. One interpretation is that ROPD leverages teacher knowledge structurally rather than distributionally: once the student acquires the teacher's basic reasoning conventions, the rubric reward directs optimization toward correctness rather than token-level fidelity.

Decoupled Supervision and Per-Criterion Progress. The alignment evidence above concerns rollout-level reward signals. We next examine whether this alignment translates into targeted per-criterion gains during training.

To analyze criterion-level dynamics, we group criteria into three categories: *Task Completion* (producing a required final answer), *Observable Quality* (verifiable correctness evidence such as key steps, constraint identification, valid transformations, and absence of hallucinated claims), and *General Reasoning* (logical consistency, self-checking, step-to-step connectivity). Category definitions and generation instructions are in Appendix G.

Table 4 reports pass rates across these categories. ROPD achieves superior gains (Δ) in every dimension. By decomposing quality into independent milestones, the rubric enables separable feedback along each dimension, unlike LOPD's entangled logit signal. Table 9 shows a 15.9% regression rate for LOPD's previously-passed cells, consistent with interference across dimensions.

Taken together, these results show that rubric rewards are more correctness-aligned than teacher log-likelihood and that ROPD's criterion-level progress is more stable. Which aspect of rubric construction contributes to these gains, however, remains an open question; Section 4.3 examines it through controlled ablation of the reward signal's components.

Key Insight

Under matched training, correctness-aligned rubric rewards stabilize criterion-level progress relative to logit-based supervision. On a fixed rollout pool, rubric rewards achieve an AUC of 0.90, compared with 0.35 for teacher log-likelihood; during training, ROPD improves more failed criteria and regresses fewer previously passed criteria than LOPD.

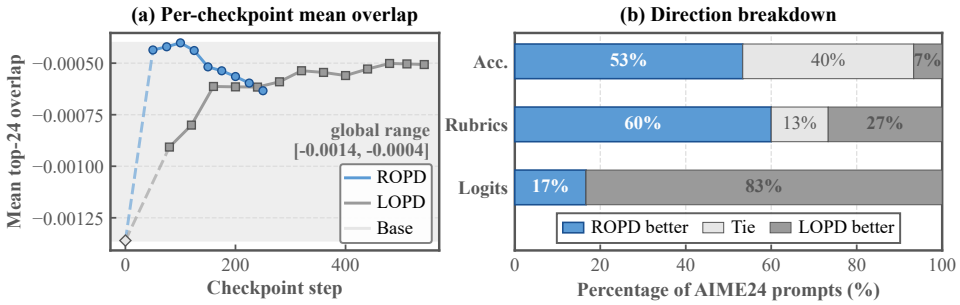


Figure 4: Evolution of stylistic mimicry and comparative performance. (a) Mimicry Trajectories: ROPD rapidly saturates stylistic alignment then pivots toward reasoning, while LOPD exhibits persistent mimicry. (b) Prompt-wise Comparison on AIME24: ROPD outperforms in reasoning accuracy and rubric satisfaction; LOPD's advantage is confined to logit mimicry.

Table 4: Rubric-level pass rates of ROPD *vs.* LOPD at early and final checkpoints on AIME24.

	ROPD			LOPD		
Rubric Category	Early	Final	Δ	Early	Final	Δ
Task Completion	54.0	67.6	+13.6	48.0	53.3	+5.3
Observable Quality	53.5	66.1	+12.6	45.2	54.7	+9.5
General Reasoning	44.6	58.9	+14.3	33.9	45.1	+11.2
Overall	**52.5**	**65.6**	**+13.1**	**44.7**	**53.0**	**+8.3**

4.3 Ablation Study: Deconstructing the Reward Signal

We disentangle these contributions through a two-part ablation (Table 5): *rubric construction source ablation* varies what information the Rubricator receives, and *leave-one-out component ablation* removes internal design choices from the full ROPD configuration.

Rubric construction source ablation. These variants share the same training infrastructure; only the Rubricator's input differs. Starting from the Generic baseline (60.79), adding teacher responses (Teacher-only 61.39) yields +0.60, while adding student rollouts (Student-only 62.25) yields +1.46. The larger effect from student conditioning suggests that anchoring criteria to the student's actual behavior produces more discriminative signals.

Actual ROPD (65.02) exceeds the additive baseline (62.85) by 2.17 points, suggesting synergistic interaction: the Rubricator requires both the teacher's quality principles and the student's error patterns. All partial-information variants (60.79–62.25) substantially underperform full ROPD (65.02).

Leave-one-out component ablation. Given the rubric construction source, ROPD's reward signal further depends on three internal design choices: multi-teacher seeding, cross-rollout rubric sharing, and blind verification. Table 5b presents a leave-one-out ablation. **Multi-teacher coverage is the primary driver.** Transitioning from $m = 4$ to $m = 1$ results in a 17.9-point drop; diverse strategies help induce path-independent criteria. **Sharing aggregates cross-rollout contrast.** A single shared rubric per prompt yields +3.77 points, surfacing systematic gaps that individual rubrics miss. **Blind scoring prevents identity bias.** Revealing response identities costs 3.27 points; the Verifier evaluates rollouts without knowing their source.

Key Insight

Teacher–student contrastive rubric construction is central to reward quality. Using both teacher responses and current student rollouts to construct rubrics outperforms all partial-information variants; multi-teacher coverage is the dominant component, while shared rubrics and blind verification provide complementary gains.

Table 5: Reward signal ablation on AIME24 (non-thinking, GPT-5.2-chat teacher). Pass@1 (%) under (a) rubric construction sources and (b) leave-one-out design choices. T-Judge uses the same teacher as a holistic judge, controlling for judge strength; Teacher-only corresponds to RaR-style reference-grounded rubrics (Gunjal et al., 2025); Generic and Teacher-only rubrics are generated once before training.

Variant	Configuration	AIME24
(a) Rubric construction source		
T-Judge	teacher holistic score (no rubric)	$62.50_{+38.33}$
Generic	prompt only	$60.79_{+36.62}$
Teacher-only	prompt + teacher responses	$61.39_{+37.22}$
Student-only	prompt + student rollouts	$62.25_{+38.08}$
(b) Leave-one-out internal design		
Qwen3-4B (base)	(no RL training)	24.17
w/o multi-teacher	single teacher answer ($m = 1$)	$47.08_{+22.91}$
w/o sharing	per-student rubrics	$61.25_{+37.08}$
w/o blind scoring	verifier sees response source	$61.75_{+37.58}$
ROPD (full)	**prompt + teacher + student,** $m = 4$	$\mathbf{65.02}_{+40.85}$

5 Related Work

On-policy Distillation. On-policy distillation (OPD) trains a student on feedback computed over its own rollouts, mitigating exposure bias and improving sample efficiency (Gu et al., 2024; Agarwal et al., 2024; Lu & Lab, 2025; Song & Zheng, 2026). Recent work strengthens OPD through objective design and reward extrapolation (Jin et al., 2026; Yang et al., 2026), training efficiency and signal calibration (Zhang et al., 2026; Wu et al., 2026; Xu et al., 2026c;d; Zheng et al., 2026), cross-tokenizer distillation (Zhang et al., 2025), and empirical analyses of failure modes (Li et al., 2026; Fu et al., 2026). These methods require dense teacher logits, which proprietary teachers do not expose.

Black-box Distillation. When teachers expose only text, black-box distillation substitutes response-level signals for logits. SeqKD on teacher demonstrations (Kim & Rush, 2016) and preference optimization on fixed pairs (Singh et al., 2025) remain offline. A narrower set preserves on-policy updates on current student rollouts: GAD (a co-evolving discriminator) (Ye et al., 2026), OVD (verbal trajectory scores) (Xiong et al., 2026), and RL-based KD (a judge's continuous reward) (Shen et al., 2026). These methods show that text-only teachers can supervise current rollouts, but collapse each response into a single preference or score.

Rubric-based Reinforcement Learning. Rubric-based reinforcement learning decomposes response quality into explicit criteria, extending RLVR beyond binary-correctness domains (Shao et al., 2024). RaR (Gunjal et al., 2025), OpenRubrics (Liu et al., 2025), Rubicon (Huang et al., 2025), RLER (Shao et al., 2025), and SibylSense (Xu et al., 2026b) formalize, scale, and evolve these criteria, building rubrics from reference answers, task descriptions, or accumulated evidence. In these methods, rubrics define the target quality of a task. ROPD shares their criterion-level rewards and uses rubrics for a different purpose, transferring a teacher's competence: criteria are induced from the contrast between multiple teacher responses and the current student rollout group, making rubrics a text-only, teacher-grounded, student-conditioned distillation interface.

6 Conclusion

We studied whether rubric-based supervision can serve as a text-only feedback interface for black-box on-policy distillation, using ROPD as a concrete instantiation. Under matched settings, ROPD outperforms logit-based OPD on all four math benchmarks and reaches the best AIME24 accuracy of standard logit-based OPD with 9.6× fewer training samples. We trace this result to correctness-aligned rubric rewards, stable criterion-level progress, and teacher–student contrastive rubric construction. These findings establish the interface in the evaluated settings; extending it to open-ended settings, where criteria are harder to verify, and to broader student families remains future work.

AI USE STATEMENT

We used LLM-based tools to aid and polish the writing of this paper, including editing draft text for clarity and flow, and proofreading. All technical content, experiments, and conclusions are the work of the authors. We did not use LLMs for data generation, methodology or experiment design, method implementation, or analysis of results.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide a complete specification of the ROPD training procedure and evaluation protocol. Appendix I presents the full pseudocode of the training loop, including the rubric induction, verification, and GRPO update stages. Appendix D lists all training hyperparameters, and Appendix F provides the exact prompt templates used to elicit rubrics and verification judgements, so the pipeline can be reimplemented with any text-only teacher. Section 3.1 describes the datasets, benchmarks, and baselines; Appendix D documents the decoding and evaluation settings, and Appendix C includes worked rubric examples and case studies that illustrate the reward construction. All models and datasets used in this paper are publicly available.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, and Joaquin Quiñonero-Candela. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

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.

Gemma Team, Google DeepMind. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025. URL https://arxiv.org/abs/2507.17746.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

HMMT. Hmmt 2025: Harvard-mit mathematics tournament, 2025.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025. URL https://arxiv.org/abs/2508.12790.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

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. URL https://aclanthology.org/D16-1139/.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. *arXiv preprint arXiv:2510.07743*, 2025.

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

MAA. Aime 2024: American invitational mathematics examination, 2024. URL https://artofproblemsolving.com/wiki/index.php/2024_AIME_I.

MAA. Aime 2025: American invitational mathematics examination, 2025. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_I.

OpenAI. Introducing gpt-5.2. https://openai.com/index/introducing-gpt-5-2/, 2025. Accessed: 2026-05-06.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen-tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025. URL https://arxiv.org/abs/2511.19399.

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

Yiyang Shen, Lifu Tu, and Weiran Wang. Reinforcement learning-based knowledge distillation with llm-as-a-judge. *arXiv preprint arXiv:2604.02621*, 2026. URL https://arxiv.org/abs/2604.02621.

Aasheesh Singh, Vishal Vaddina, and Dagnachew Birru. Orpo-distill: Mixed-policy preference optimization for cross-architecture llm distillation. *arXiv preprint arXiv:2509.25100*, 2025. URL https://arxiv.org/abs/2509.25100.

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

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.

Jing Xiong, Hui Shen, Shansan Gong, Yuxin Cheng, Jianghan Shen, Chaofan Tao, Haochen Tan, Haoli Bai, Lifeng Shang, and Ngai Wong. Ovd: On-policy verbal distillation. *arXiv preprint arXiv:2601.21968*, 2026. URL https://arxiv.org/abs/2601.21968.

Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tao Zhao, and Haoyu Wang. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable llm post-training. *arXiv preprint arXiv:2602.01511*, 2026a. URL https://arxiv.org/abs/2602.01511.

Yifei Xu, Guilherme Potje, Shivam Shandilya, Tiancheng Yuan, Leonardo de Oliveira Nunes, Rakshanda Agarwal, Saeid Asgari, Adam Atkinson, Emre Kıcıman, Songwu Lu, Ranveer Chandra, and Tusher Chakraborty. Sibylsense: Adaptive rubric learning via memory tuning and adversarial probing. *arXiv preprint arXiv:2602.20751*, 2026b. URL https://arxiv.org/abs/2602.20751.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. Paced: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026c.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026d.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *CoRR*, abs/2602.12125, 2026.

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*, 2026. URL https://arxiv.org/abs/2511.10643.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E Tillman. Fast and effective on-policy distillation from reasoning prefixes. *arXiv preprint arXiv:2602.15260*, 2026.

Xue Zhang, Songming Zhang, Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. A dual-space framework for general knowledge distillation of large language models. *arXiv preprint arXiv:2504.11426*, 2025.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

APPENDIX

Appendix Overview

§A Broader Impacts 14
§B Related Work (Complete Version) 15
§C Qualitative Analysis and Case Studies 16
§D Hyperparameters and Training Configuration 17
§F Prompt Templates 19
§G Additional Figures and Analysis 23
§H Cross-Architecture Generalization 25
§I Algorithm Pseudocode and Method Details 26

A Broader Impacts

ROPD studies a potentially more information-efficient interface for LLM alignment by substituting opaque token-level mimicry with structured, human-readable semantic rubrics. In the matched comparison reported in Section 3.3, this interface reaches the LOPD reference performance with approximately 9.6$\times$ fewer training samples, while API and serving costs remain practical considerations. By grounding the distillation process in explicit criteria rather than implicit probability distributions, ROPD offers enhanced transparency; developers can audit and intervene in the specific principles being transferred. However, the reliance on automated rubricators may inadvertently propagate or amplify biases inherent in the teacher's evaluative logic. If a teacher's rubrics contain flawed or prejudiced heuristics, the student may systematically internalize these errors under the guise of logical rules. Therefore, the deployment of rubric-based distillation should be accompanied by rigorous fairness audits of the induced criteria and the development of safety-aligned verifiers, especially in high-stakes sectors like medicine or law. We view ROPD as a starting point for studying text-only distillation, while emphasizing that the transition from blind imitation to principled alignment must be guided by ethical accountability and robust safety monitoring.

B Related Work (Complete Version)

This section provides the complete Related Work discussion with full context and citations. A condensed overview appears in Section 5 of the main text.

Knowledge distillation and on-policy distillation. Knowledge distillation (KD) transfers the behavior of a large teacher model into a smaller student, and is widely used to adapt or compress language models. Classical KD matches teacher soft targets on a fixed data distribution (Hinton et al., 2015), and Sequence-Level Knowledge Distillation (SeqKD) extends this to generation by substituting teacher-decoded sequences for label-level targets (Kim & Rush, 2016). Both are offline and suffer from exposure bias: training follows teacher-forced trajectories, while inference exposes the student to its own prefixes and errors, creating a mismatch between the distributions seen at training and test time. On-policy distillation (OPD) addresses this by training on student-generated sequences: MiniLLM optimizes reverse-KL on sampled responses (Gu et al., 2024), Generalized Knowledge Distillation (GKD) learns from self-generated mistakes with teacher feedback (Agarwal et al., 2024), and recent work scales this recipe to reasoning post-training (Yang et al., 2025; Li et al., 2026). Despite this progress, these methods share a common assumption: they require token-level teacher information such as logits, which is unavailable for proprietary teachers and difficult to align across different architectures or vocabularies. ROPD studies the complementary black-box regime where the teacher exposes only text responses, enabling on-policy distillation when token-level supervision is infeasible.

Black-box On-policy Distillation. Recent black-box distillation methods answer this question with different forms of response-level supervision: ORPO-Distill constructs mixed-policy preference pairs from teacher and student reasoning traces (Singh et al., 2025); GAD trains a discriminator to distinguish teacher from student responses and uses its score as a co-evolving reward (Ye et al., 2026); On-policy Verbal Distillation (OVD) asks the teacher for discrete verbal trajectory scores, avoiding token alignment and reducing memory cost (Xiong et al., 2026); and RL-based KD with LLM-as-a-Judge trains from scalar evaluator rewards over unlabeled data (Shen et al., 2026). These methods demonstrate that output-only teachers can supervise student rollouts, but their signals remain largely implicit or holistic: preferences compare whole traces, discriminators hide the criteria behind a learned score, and verbal or judge rewards summarize a response into a single value. ROPD instead makes the distillation interface explicit by deriving shared rubrics from multiple teacher answers and current student rollouts, verifying each rollout against these criteria, and using the resulting weighted pass rates as on-policy rewards.

Rubric-based Reinforcement Learning. Reinforcement learning with verifiable rewards (RLVR) has driven strong gains in math and code (Shao et al., 2024), but its reliance on binary correctness limits it to domains with deterministic ground truth. Rubrics address this by decomposing response quality into structured, multi-dimensional criteria, extending RL to open-ended tasks. Rubrics-as-Rewards (RaR) formalized instance-specific rubrics as on-policy RL rewards, showing RLVR to be a special case of rubric-based RL (Gunjal et al., 2025). On rubric generation, OpenRubrics scales synthesis via contrastive prompting (Liu et al., 2025). On training dynamics, Rubicon identifies a seesaw effect between conflicting rubric types—improving one dimension can degrade another—and proposes multi-stage training to stabilize learning (Huang et al., 2025). Recognizing that static rubrics fail to capture emergent behaviors, Reinforcement Learning with Evolving Rubrics (RLER) and SibylSense introduce evolving rubrics that co-adapt with the policy: RLER grounds them on retrieved search evidence (Shao et al., 2025), while SibylSense pursues adversarial memory tuning (Xu et al., 2026b). A common assumption underlies these methods: rubrics function as evaluation instruments—they measure response quality against criteria sourced from benchmarks, reference answers, or self-generated preferences—but they are not designed to transfer knowledge from a stronger model to a weaker one. ROPD instead induces rubrics from the contrast between multi-teacher answers and on-policy student rollouts, converting them via a verifier into weighted pass-rate rewards for Group Relative Policy Optimization (GRPO). This repositions rubrics as a distillation interface—the resulting reward is simultaneously teacher-grounded and rollout-conditioned.

C Qualitative Analysis and Case Studies

Case study: Rubric disagreement reveals teacher bias. When multiple teacher answers disagree on a rubric criterion, the Rubricator surfaces this ambiguity explicitly (*e.g.*, "Criterion 7: Uses proof by induction – 2/4 teachers support, 2/4 use direct computation"). This prevents the student from overfitting to one teacher's style.

Case study: Failure mode – rubric exploitation. In rare cases ($< 2\%$ of rollouts), the student learns to produce responses that score highly on rubrics without being substantively correct (*e.g.*, formatting tricks, keyword stuffing). We observe this mainly in early training, and it diminishes as training proceeds: each rubric contains at least one high-weight criterion that checks final-answer correctness (Appendix F), which bounds the reward attainable by responses that satisfy only surface-level criteria.

Rubric item examples. Table 6 shows representative rubric items generated by the Rubricator for different prompt types.

Table 6: **Representative rubric items generated by ROPD's Rubricator.** $K = 12$ items are generated per instance; we show 4 examples per domain.

Domain	Example Rubric Items
Math (AIME)	"The solution defines all variables before computation" "Intermediate steps are explicitly justified with theorems or algebraic rules" "The final answer is boxed and matches the required format" "No arithmetic errors in the numerical computation chain"
Science (GPQA)	"The answer identifies the relevant physical/chemical principle" "Quantitative reasoning includes correct unit conversions" "Alternative hypotheses are considered and ruled out" "The conclusion explicitly addresses the question asked"
Medicine (HealthBench)	"Diagnosis is supported by specific findings from the case description" "Differential diagnosis lists at least 2 alternative conditions" "Treatment recommendation follows guideline-concordant reasoning" "Referral or follow-up plan is specified when appropriate"

D HYPERPARAMETERS AND TRAINING CONFIGURATION

Complete hyperparameter specification. Table 7 lists all hyperparameters used in ROPD experiments.

Table 7: **Complete hyperparameter configuration.** For the matched white-box comparison (Table 2), Qwen3-30B-A3B replaces GPT-5.2-chat-latest as the teacher, Rubricator, and Verifier on the math track; all other settings are unchanged.

Hyperparameter	Math Track	Science Track	Medical Track
	Model		
Student model	Qwen3-4B	Qwen3-4B	Qwen3-4B
Teacher model	GPT-5.2-chat-latest	GPT-5.2-chat-latest	GPT-5.2-chat-latest
Rubricator model	GPT-5.2-chat-latest	GPT-5.2-chat-latest	GPT-5.2-chat-latest
Verifier model	GPT-5.2-chat-latest	GPT-5.2-chat-latest	GPT-5.2-chat-latest
	Training		
Training dataset	DAPO-Math-17K	RaR-Science-20k	RaR-Medical-20k
Learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
LR scheduler	Cosine	Cosine	Cosine
Warmup steps	100	100	100
Batch size	32	32	32
GRPO group size n	8	8	8
Max training steps	531	625	625
Precision	bf16	bf16	bf16
Optimizer	AdamW	AdamW	AdamW
AdamW (β_1, β_2)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Weight decay	0.1	0.1	0.1
Gradient clipping	1.0	1.0	1.0
	ROPD Specific		
Teacher answers m	4	4	4
Rubric items K	4–12	4–12	4–12
Rubricator temperature	0.7	0.7	0.7
Verifier temperature	0.0	0.0	0.0
	Training Rollout Decoding		
Max rollout tokens	8192	8192	8192
Teacher temperature	1.0	1.0	1.0
Student rollout temp	1.0	1.0	1.0
	Hardware		
GPUs	8×A100-80GB	8×A100-80GB	8×A100-80GB

Validation and checkpoint selection. We evaluate every 50 steps on the validation split of the corresponding training dataset and select the checkpoint with the highest validation accuracy; benchmark results are not used for checkpoint selection. For OOD evaluation on IFEval, we use the math-track checkpoint without any instruction-following fine-tuning.

Evaluation Details. We use temperature $= 1.0$ and top-$p = 0.95$ for all sampling, with a maximum output length of 32,768 tokens. For each problem, we sample $k = 16$ responses and report pass@1. For *think* mode, we prepend a standard chain-of-thought prompt; for *no-think*, answers are generated directly.

E BASELINE IMPLEMENTATION DETAILS

All baselines follow the procedures of their original papers and use the same student, training data, and evaluation protocol as ROPD within each setting (Section 3.1); they differ only in the source of the training signal. **SFT** fine-tunes the student on teacher responses generated offline for the training prompts (Kim & Rush, 2016). **T-Judge** asks the teacher to assign a holistic scalar score to each student rollout and uses it as the GRPO reward, following LLM-as-a-Judge distillation (Shen et al., 2026). **OVD** uses the teacher's discrete verbal trajectory scores as rewards (Xiong et al., 2026). **GAD**

trains a discriminator to distinguish teacher from student responses and uses its score as a co-evolving reward (Ye et al., 2026). **LOPD** uses the teacher's per-token log-probabilities on student rollouts as a dense reverse-KL signal (Agarwal et al., 2024; Lu & Lab, 2025). **ExOPD** extends LOPD with reward extrapolation ($\lambda > 1$) (Yang et al., 2026).

Comparison with reported ExOPD numbers. The ExOPD results in Table 2 are lower than those reported by Yang et al. (2026) for a Qwen3-4B student. The two setups differ in training data (DeepMath in Yang et al. (2026) *vs.* DAPO-Math-17K here) and in evaluation protocol (generation length and number of samples per problem). All methods in Table 2 share our configuration, so the comparison is controlled within our setting, whereas absolute numbers are not directly comparable across papers.

GAD performance. GAD yields smaller gains than the other on-policy baselines in Table 1. Its reward comes from a discriminator trained to separate teacher from student responses. A plausible explanation is that when the student starts far from the teacher, as in non-thinking mathematics, the discriminator separates the two sources easily and quickly saturates, so its scores provide limited distinctions among rollouts of the same prompt.

F PROMPT TEMPLATES

Rubricator System Prompt (English)

```
You are an expert in educational assessment and contrastive rubric design.
Your task is to analyze a question together with two sets of responses:
- A set of TEACHER responses (multiple reference answers from strong models;
  each may contain errors or use different approaches, but collectively
  represent high-quality answer behavior).
- A set of STUDENT responses (multiple rollouts from a weaker model currently
  under training; these are on-policy samples that need actionable improvement
  signals).

Your goal is to generate a SINGLE shared rubric that applies to ALL student
responses for this question.

# Input Data
[Question]: {question}

[Teacher Responses] (m answers):
[1]: {teacher_response_1}
...
[m]: {teacher_response_m}

[Student Responses] (n rollouts):
[1]: {student_response_1}
...
[n]: {student_response_n}

# Core Objective
Generate ONE shared rubric with K criteria. The rubric should:
- Capture quality dimensions where teacher responses show strong, consistent
  performance.
- Target dimensions where student responses exhibit systematic weaknesses.
- Help move the student policy toward the answer-quality level of the teacher
  distribution.
- Be applicable to any single student response independently at verification
  time.

Important constraints:
- Do NOT reward copying a specific teacher's wording, surface style, or exact
  method.
- Do NOT assume any single teacher response is fully correct.
- Do NOT define criteria that require matching a specific teacher's final
  answer.
- Do NOT define criteria that can only be judged by comparing against a teacher
  response at verification time.
- Each criterion must be evaluable on a single response on its own.

# Multi-Teacher Design Rules
- When multiple teachers agree on a quality dimension, that dimension deserves
  higher weight.
- When teachers disagree on an approach, a criterion should accept ANY valid
  approach, not penalize deviation from the majority.
- Rubrics should not collapse into "the student should be more like Teacher
  \#3" --- they must remain response-level quality criteria.

# Hard Requirements
Each criterion must be:
1. Specific and Measurable: Clearly define a concrete answer-quality merit.
2. Binary Evaluable: A verifier should be able to mark it True or False for one
   response alone.
3. Instructionally Useful: It should point to a meaningful improvement
   direction for the students.
4. Alternative-Method Safe: A different valid approach that exhibits the same
   merit should still be rewarded.
5. Distinguishing: Prefer merits that teachers consistently show and students
   systematically lack.
6. Black-Box Compatible: Prefer criteria that evaluate observable answer
   behavior and response quality.

# Required Category Taxonomy
Your rubric should be guided by the following three categories. Use the
`category` field to assign each criterion to exactly one category.

1. Task Completion
   Whether the response completes the task and produces the required final
   answer in the correct form. This includes identifying the target quantity,
   presenting the answer explicitly, and meeting format requirements.

2. Observable Quality
```

```
   Whether the response demonstrates strong observable correctness signals
   under black-box evaluation. This includes correct intermediate steps,
   valid factorization or algebraic manipulation, identification of key
   constraints (\textit{e.g.}, parity obstructions), and absence of hallucinated
   claims or guessed answers.

3. General Reasoning
   Broad reasoning qualities such as logical coherence, step-by-step
   derivation flow, planning structure, self-checking behavior, clarity,
   and focus. Use this category when such qualities are genuinely relevant
   and improve teacher-student separation.

# Category Priorities
1. Preserve general validity of the rubric for the question.
2. Prioritize Task Completion by default--at least one high-weight criterion
   should verify that the response answers the requested target and presents
   it in the required form.
3. Prioritize Observable Quality criteria that directly check correctness of
   intermediate steps, mathematical manipulations, and domain-specific
   reasoning (\textit{e.g.}, factorization, constraint identification).
4. Use General Reasoning when genuinely relevant and it improves
   teacher-student separation, but avoid rewarding superficial stylistic
   performance.
5. Make the rubric produce actionable learning-direction signals for the
   student.

Most of the total points should come from criteria that are likely satisfied
by most teacher responses but not by most student responses.

# Additional Design Rules
- At least one high-value criterion should check whether the response answers
  the requested final target.
- At least one high-value criterion should check whether the final answer is
  presented in the form required by the question.
- Prefer criteria that directly support task completion, final-answer quality,
  and answer-object compliance.

# Forbidden Criterion Patterns
Do NOT write criteria like:
- "uses the same method as the teacher(s)"
- "matches the teacher's final answer"
- "has the same wording/style/structure as the teacher responses"
- criteria that encode a potentially wrong intermediate claim from a specific
  teacher
- criteria that mainly reward length, elaborateness, or superficial stylistic
  performance

# Output Format
Return a JSON object:
{
  "schema_version": "black_opd.rubric.v1",
  "question_domain": "math",
  "rubrics": [
    {
      "criterion_id": "c1",
      "category": "Task Completion",
      "criterion": "Produces an explicit final answer.",
      "weight": 5
    },
    {
      "criterion_id": "c3",
      "category": "Observable Quality",
      "criterion": "Identifies the parity obstruction (P(n) odd, 2024 even
                   implies no solution).",
      "weight": 5
    },
    {
      "criterion_id": "c5",
      "category": "General Reasoning",
      "criterion": "Argument is logically coherent, each step follows from
                   the last.",
      "weight": 5
    },
    ...
  ],
  "K": 6,
  "max_weighted_sum": 22,
  "estimated_student_pass_rate": 0.30
}

# Note
The example above uses K=8 purely for illustration. The Rubricator
```

```
chooses K dynamically per prompt based on the question's complexity;
the resulting K is whatever value best captures the quality dimensions
of the given (question, teacher set, student set), and may take any
integer value in [4, 12] (see Output Constraints below).

# Output Constraints
- Choose K dynamically based on the prompt's complexity; K must be an
  integer between 4 and 12 (typically 6--8).
- `weight` (w_k) must be integers from 1 to 5.
- `K` must equal the number of rubric items in the list.
- `max_weighted_sum` must equal the sum of all weights.
- `estimated_student_pass_rate` should be strictly below 0.5.

# Final Self-Check Before Answering
Before producing the JSON, verify internally that:
- every criterion can be judged on a single response without referencing any
  teacher or peer response
- the rubric would likely separate the teacher distribution from the student
  distribution
- the rubric prioritizes task completion and final-answer contract when they
  are central to the question
- the rubric does not reward superficial similarity to any specific teacher
- the rubric leaves genuine room for improvement for the students
- the rubric does not collapse into overly generic criteria only

Return only the JSON object without additional commentary.
```

Verifier System Prompt

```
You are an expert evaluator. Your task is to assess a single response
against a set of binary answer-quality rubrics.

Your task: evaluate only the current response given the question,
response, and rubric set.

[Task Input]:
- Question: the problem being solved.
- Response: the single response currently under evaluation.
- Rubrics: a set of binary evaluation criteria. Each criterion includes:
  - criterion_id: stable identifier for this criterion
  - category: the aspect being evaluated (context label only; do not
    introduce extra requirements beyond the criterion text)
  - criterion: a binary statement that rewards a specific merit
  - weight: the weight w_k assigned when this criterion is satisfied;
    used only for final score aggregation, not for judging satisfaction

[Core Evaluation Rules]:
For each criterion, determine whether the current response exhibits the
described merit.
- Judge each criterion using only the question, the response, and the
  criterion text itself. Do not add extra standards not explicitly
  required by the question or rubric.
- If a criterion contains multiple explicit conditions, mark it `true`
  only when ALL conditions are met; mark `false` otherwise.
- If the response uses a different but equally valid method that still
  exhibits the described merit, mark it `true`.
- If the merit is not clearly demonstrated, mark it `false`.

[Task Instructions]:
Evaluate each criterion in the given order:
- If the criterion is satisfied, output `true`.
- Otherwise output `false`.
- weighted_score = sum of weights of all criteria marked `true`.
- pass_rate = weighted_score / (sum of all criteria weights).

[Output Format]
Return a JSON object:
{
  "schema_version": "black_opd.verifier.v1",
  "judgements": [true, false, true],
  "weighted_score": 7,
  "pass_rate": 0.35
}

[Output Constraints]
- `schema_version` must be exactly `black_opd.verifier.v1`.
- `judgements` list must be in the same order as the input rubric.
- `weighted_score` = sum of weights where judgement is true
  (sum w_k * v_{i,k}).
- `pass_rate` = weighted_score / sum of all weights
```

```
  = (sum w_k * v_{i,k}) / (sum w_k).

[Important Guidelines]
- Be objective and judge each criterion independently.
- No partial credit within a single criterion.
- Do not mark `true` for superficial features such as length, confident
  tone, or stylistic performance unless the criterion explicitly requires
  them.

Now evaluate the following:
Question: {question}
Response: {resp}
Rubrics: {rubrics}

Return only the JSON object, without any additional text or commentary.
```

GRPO reward prompt. The GRPO reward for rollout y_j^S is computed as:

$$R(y_j^S) = \underbrace{\frac{\sum_{k=1}^{K} w_k \cdot \mathbb{I}[\text{pass}_k]}{\sum_{k=1}^{K} w_k + \epsilon}}_{\text{weighted pass rate}} \tag{4}$$

where ϵ is a small constant for numerical stability, and group-relative advantage is normalized per prompt.

G Additional Figures and Analysis

Efficiency curves and API cost. Figure 5 reports the complete efficiency comparison for the matched Qwen3-30B-A3B teacher and Qwen3-4B student under non-thinking decoding. ROPD recovers LOPD's best AIME24 performance with approximately 9.6× fewer samples (1.6k *vs.* 15.4k) and reaches the same threshold in approximately 6.3× less wall-clock time (5.5h *vs.* 34.4h). The curves also show the post-saturation behavior summarized in Section 3.3. The estimated API cost for the full ROPD training pipeline is approximately $1.5k, including teacher response generation, Rubricator, and Verifier calls. This monetary estimate is reported separately from sample count and GPU wall-clock time and depends on the pricing and serving configuration.

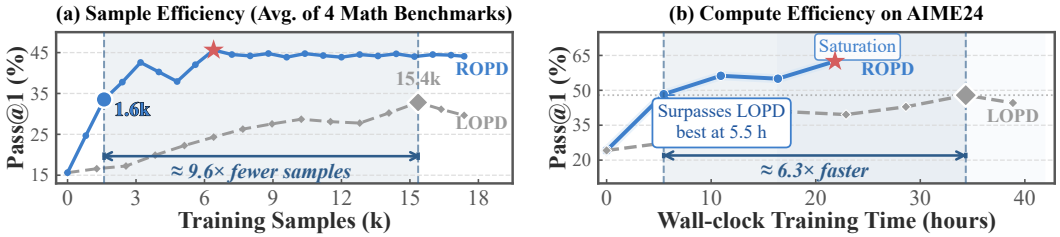


Figure 5: **Complete efficiency curves for the matched comparison.** Qwen3-30B-A3B is the teacher and Qwen3-4B is the student, both evaluated under non-thinking decoding. (a) Average sample efficiency across the four mathematics benchmarks; the star (⋆) marks ROPD's own performance plateau at 6.4k samples. (b) AIME24 compute efficiency, including the higher per-step overhead from Rubricator and Verifier inference.

Black-box results in thinking mode. For completeness, Table 8 reports the thinking-mode results moved from the main-text black-box comparison.

Table 8: **Black-box distillation results in thinking mode.** All results are reported in Pass@1 (%). Bold and underline indicate the best and second-best performance, respectively.

	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	GPQA-D.	HealthBench	IFEval
GPT-5.2-chat (teacher)	80.83	67.08	43.75	57.50	78.66	92.82	94.37
Qwen3-4B (student)	70.42	59.58	33.33	48.75	53.59	85.30	86.46
T-Judge	<u>72.50</u> $_{+2.08}$	65.48 $_{+5.90}$	<u>38.75</u> $_{+5.42}$	<u>51.25</u> $_{+2.50}$	53.85 $_{+0.26}$	85.58 $_{+0.28}$	86.55 $_{+0.09}$
OVD (Xiong et al., 2026)	71.68 $_{+1.26}$	<u>65.83</u> $_{+6.25}$	38.34 $_{+5.01}$	50.42 $_{+1.67}$	<u>54.17</u> $_{+0.58}$	<u>85.98</u> $_{+0.68}$	86.38 $_{-0.08}$
GAD (Ye et al., 2026)	70.65 $_{+0.23}$	61.28 $_{+1.70}$	35.00 $_{+1.67}$	49.58 $_{+0.83}$	53.85 $_{+0.26}$	85.70 $_{+0.40}$	<u>86.62</u> $_{+0.16}$
ROPD (ours)	**75.41** $_{+4.99}$	**68.75** $_{+9.17}$	**39.16** $_{+5.83}$	**54.17** $_{+5.42}$	**55.05** $_{+1.46}$	**86.87** $_{+1.57}$	**86.95** $_{+0.49}$

Leaderboard bar chart (think mode). Figure 6 shows the leaderboard-style comparison under think decoding.

Leaderboard bar chart (no-think mode). Figure 7 shows the leaderboard-style comparison under no-think decoding.

Per-criterion transition: ROPD vs. LOPD. Table 9 provides the full per-category transition breakdown for the cell-level analysis in Section 4.2.

Reward-signal alignment: supplementary tables and figures. Section 4.2 in the main text reports the key alignment metrics and ROPD checkpoint dynamics. Tables 10 and 11 provide the complete numerical results underlying that analysis. Figures 9–12 visualize the checkpoint-level dynamics, correctness-conditioned signal distributions, final-checkpoint paired comparison, and the top-24 overlap trajectories.

Analysis pool protocol. All numbers in this subsection are computed on a dedicated offline analysis pool consisting of 30 AIME24 prompts × 8 rollouts × 13 checkpoints (5 ROPD, 7 LOPD, 1 Base) =

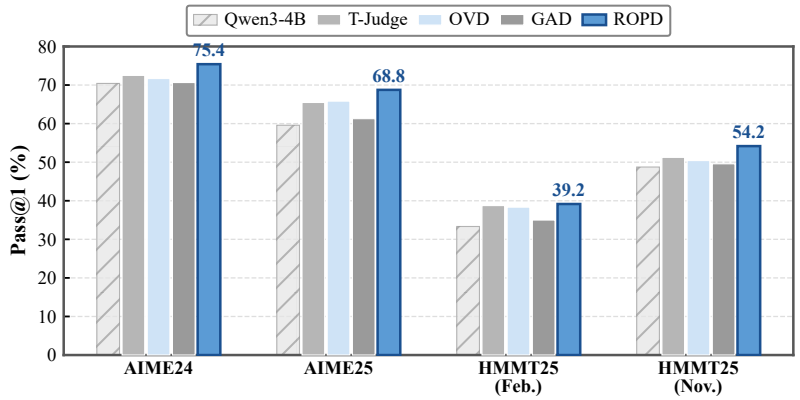


Figure 6: **Leaderboard comparison – think mode.** Horizontal bar chart in DeepSeek-v4 leaderboard style.

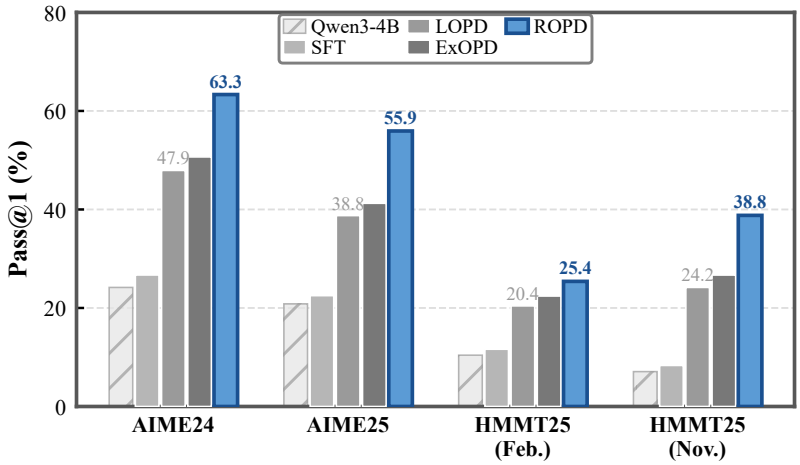


Figure 7: **Leaderboard comparison – no-think mode.** Horizontal bar chart in DeepSeek-v4 leaderboard style.

3,120 responses. Rollouts are sampled independently of the main benchmark evaluation (*i.e.*, this is not a subset of the k=16 rollouts behind Tables 1–2 and Figure 5), but use the same decoding configuration: temperature 1.0, top-p 0.95, no-think. Verifier scoring uses Qwen3-30B-A3B as a single shared judge across all families and checkpoints, distinct from the GPT-5.2 Verifier used during ROPD training. For each signal, AUC measures how well it ranks correct above incorrect responses (correctness is the positive label), computed over all responses (AUC_{all}). The teacher signal is the length-normalized log-likelihood $\frac{1}{L}\sum_t \log p_T(y_t \mid y_{<t})$. The preference-conflict rate (Bad-upd.) is the fraction of responses whose GRPO-style advantage, obtained by standardizing the signal within its 8-rollout group, has the opposite sign to the advantage obtained from correctness. Because rollouts are an independent k=8 sample, the accuracy column `Acc.` in Table 11 can differ from the main k=16 benchmark by up to $\sim$5 points at unstable early checkpoints (*e.g.*, ROPD step 50: 43.75% here vs. 48.33% in Figure 5); converged checkpoints (ROPD step $\geq$ 150, LOPD step $\geq$ 240) agree within $\leq 0.1\%$. This sampling variance is consistent with the binomial standard error expected for $30 \times 8 = 240$ binary outcomes and does not affect any of the within-pool reward-signal comparisons reported in Section 4.2.

Table 9: **Per-category cell transition: ROPD vs. LOPD.** A cell (p, k) is improved if $q_{\text{early}} < 0.5$ and $q_{\text{final}} \geq 0.5$, and regressed if $q_{\text{early}} \geq 0.5$ and $q_{\text{final}} < 0.5$.

	ROPD (50→250)			LOPD (80→543)		
Category	Improve	Regress	Net	Improve	Regress	Net
Task Completion	17/35 (48.6%)	1/34 (2.9%)	+16	7/31 (22.6%)	7/38 (18.4%)	+0
Observable Quality	31/58 (53.4%)	5/68 (7.4%)	+26	21/65 (32.3%)	9/61 (14.8%)	+12
General Reasoning	7/17 (41.2%)	1/11 (9.1%)	+6	6/20 (30.0%)	1/8 (12.5%)	+5
Overall	55/110 (50.0%)	7/113 (6.2%)	+48	34/116 (29.3%)	17/107 (15.9%)	+17

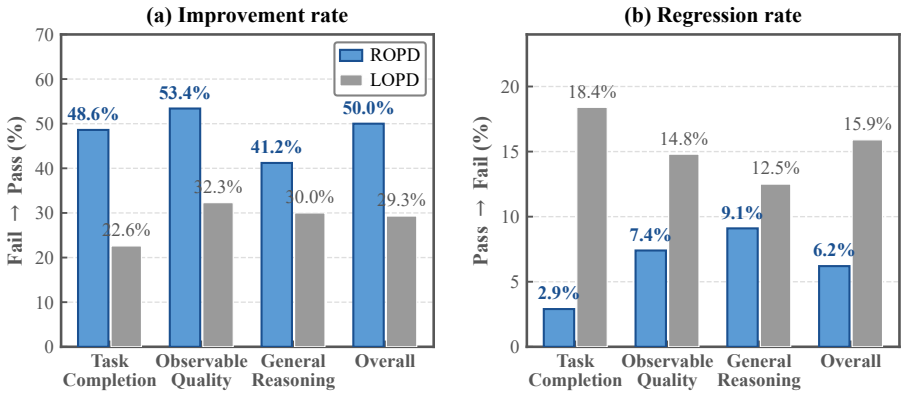


Figure 8: **Cell-level transition comparison: ROPD vs. LOPD.** (a) Improvement rate: fraction of initially-failed cells (q<0.5) that become passed (q≥0.5) at the final checkpoint. (b) Regression rate: fraction of initially-passed cells that become failed. ROPD improves more and regresses less in every rubric category.

H Cross-Architecture Generalization

To evaluate whether ROPD transfers beyond the primary Qwen3-4B student, we substitute it with Gemma3-it-4B (Gemma Team, Google DeepMind, 2025), a model from a different architecture family with substantially lower initial performance (6.67% on AIME24 vs. Qwen3-4B's 24.17%). All other experimental conditions remain identical to the black-box setting described in Section 3.1.

As shown in Table 12, ROPD consistently improves over the Gemma3-it-4B base model on all four benchmarks, with the largest gain on AIME24 (+3.33, a 50% relative improvement). OVD and GAD produce marginal or no gains under this weaker starting policy. Because ROPD's rubric criteria evaluate each response independently against absolute quality dimensions, they remain informative even when student outputs are far from teacher quality. This single-architecture experiment provides preliminary evidence that ROPD is not specific to the Qwen family; however, it covers only one alternative architecture and a limited task range, so systematic cross-architecture conclusions require further investigation.

Table 10: **Complete family-level signal-correctness alignment.** AUC and preference-conflict rate for three candidate reward signals on AIME24 responses, broken down by model family.

Family	Responses	Acc.	Rubric reward		Teacher logprob		Top-24 overlap
			AUC_{all}	Bad-upd.	AUC_{all}	Bad-upd.	AUC_{all}
ROPD	1,200	0.554	0.898	0.151	0.351	0.599	0.497
LOPD	1,680	0.376	0.882	0.196	0.524	0.503	0.638
Base	240	0.221	0.861	0.246	0.658	0.467	0.762

Table 11: **Complete checkpoint summary.** All 13 checkpoints from ROPD, LOPD, and Base evaluated under a single shared-rubric Verifier on AIME24. Rubric reward rises with training for both methods; teacher log-likelihood declines for ROPD. Top-24 overlap moves in opposite directions across the two families; see Figure 12. `Acc.` is computed on the analysis pool (k=8 rollouts/prompt, no-think); see "Analysis pool protocol" above for how it relates to the main k=16 benchmark.

Family	Step	Acc.	Rubric reward	Teacher logprob	Top-24 overlap
ROPD	50	0.438	0.528	−0.335	0.9996
	100	0.525	0.523	−0.345	0.9996
	150	0.550	0.623	−0.394	0.9995
	200	0.625	0.636	−0.400	0.9994
	250	0.633	0.658	−0.430	0.9994
LOPD	80	0.275	0.459	−0.372	0.9991
	160	0.313	0.470	−0.331	0.9994
	240	0.363	0.491	−0.356	0.9994
	320	0.388	0.514	−0.342	0.9995
	400	0.417	0.511	−0.349	0.9994
	480	0.421	0.539	−0.336	0.9995
	543	0.454	0.523	−0.341	0.9995
Base	0	0.221	0.444	−0.421	0.9986

I Algorithm Pseudocode and Method Details

Algorithm 1 presents the complete ROPD training procedure. The algorithm operates in a fully black-box regime: the teacher, Rubricator, and Verifier are accessed solely through text prompts and JSON-structured outputs, without any access to internal logits or hidden states.

Group Relative Policy Optimization. We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to optimize the student from response-level rewards. For each prompt x, GRPO samples a group of n responses from the old policy $\pi_{\theta_{\text{old}}}$ and obtains response-level rewards $\{r_i\}_{i=1}^n$. The advantage of each response is normalized within the group:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^n)}{\text{std}(\{r_j\}_{j=1}^n) + \epsilon}, \tag{5}$$

which avoids training a separate value model and makes the update depend on relative quality among rollouts for the same prompt.

Let $y_i = (y_{i,1}, \ldots, y_{i,|y_i|})$ be the i-th sampled response. At token position t, define the policy ratio

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}. \tag{6}$$

The clipped GRPO objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{y_i\}}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\Big(\min\Big(\rho_{i,t}(\theta)A_i,\ \text{clip}(\rho_{i,t}(\theta), 1-\eta, 1+\eta)A_i\Big) - \beta D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, y_{i,<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid x, y_{i,<t})\big)\Big)\right], \tag{7}$$

where η is the clipping coefficient, π_{ref} is a fixed reference policy, and β controls the KL penalty. In black-box OPD, the teacher-derived supervision described above can be used to construct the rewards $\{r_i\}_{i=1}^n$, allowing GRPO to update the student directly on its self-generated responses.

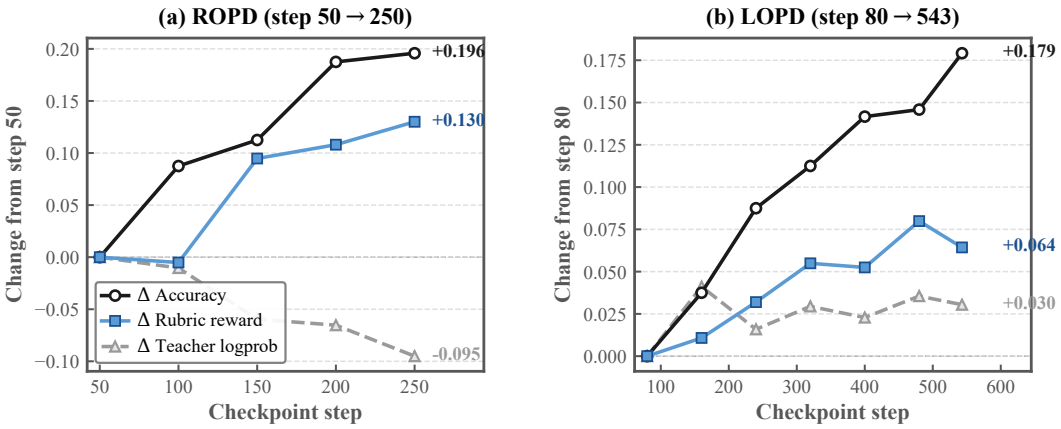


Figure 9: **Checkpoint dynamics: relative change from earliest checkpoint.** ROPD (left) and LOPD (right). Accuracy and rubric reward are normalized relative to their values at the first checkpoint; teacher log-likelihood is shown on the same relative scale. ROPD's accuracy and rubric reward rise together while teacher likelihood falls; LOPD shows weaker coupling between the three quantities.

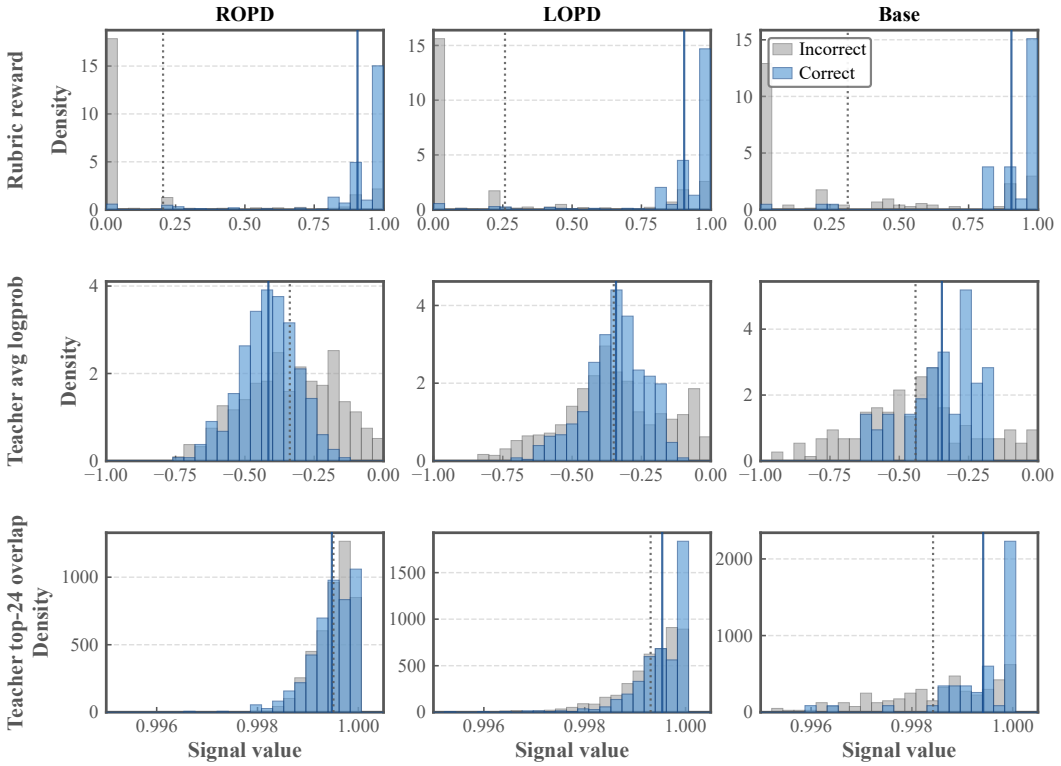


Figure 10: **Signal distributions conditioned on correctness.** Rubric reward (top) strongly separates correct from incorrect responses in all three families. Teacher average log-likelihood (middle) shows weak or reversed separation, particularly for ROPD where correct responses have *lower* teacher likelihood. Teacher top-24 overlap (bottom) separates correct from incorrect responses for Base and LOPD but not for ROPD, consistent with the AUC values in Table 10.

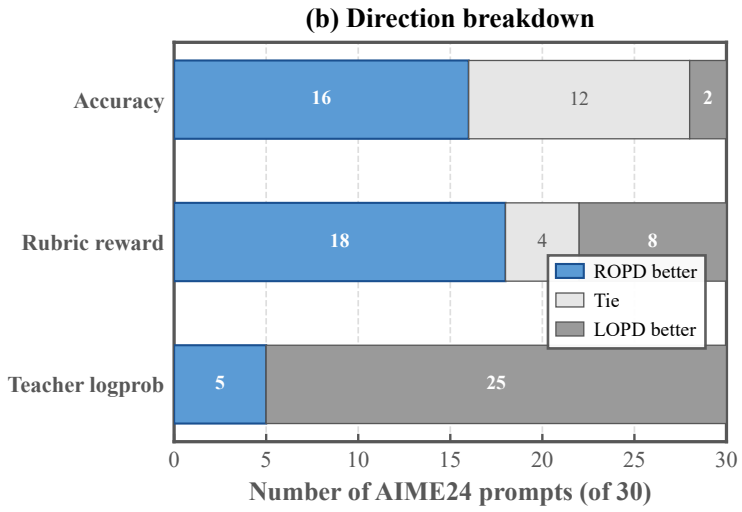


Figure 11: **Final-checkpoint paired comparison (Black step 250 vs. White step 543).** Per-prompt deltas with bootstrap 95% confidence intervals. ROPD final is more accurate (+0.179, CI excludes zero), achieves higher rubric reward (+0.135, CI excludes zero), yet has *lower* teacher log-likelihood (−0.089, CI excludes zero). Prompts are AIME24 (30 prompts).

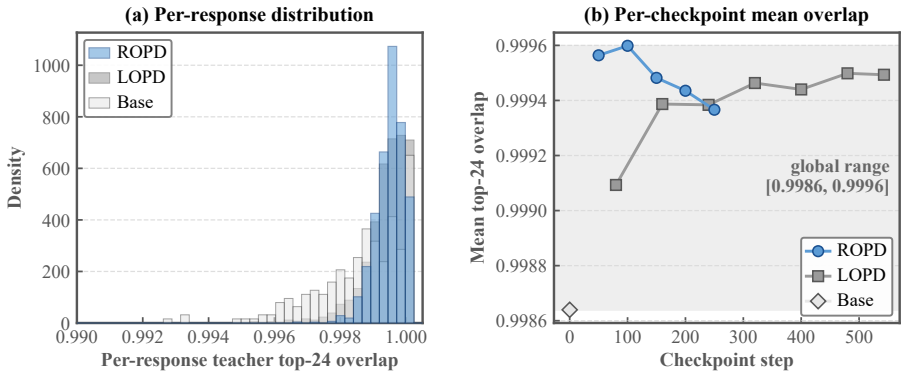


Figure 12: **Teacher top-24 overlap trajectories.** Mean top-24 overlap stays high across all checkpoints and families (0.9986–0.9996), but the two trained families move in opposite directions (Table 11): ROPD reaches 0.9996 at its first checkpoint and then declines to 0.9994, whereas LOPD rises from 0.9991 to 0.9995 and remains there. This metric discriminates correctness most clearly for the least saturated family (Base, AUC 0.762) and least clearly for ROPD (AUC 0.497), so its low pooled AUC reflects proximity to the ceiling rather than an absence of signal.

Table 12: Cross-architecture generalization. Pass@1 (%) with Gemma3-it-4B as the student (non-thinking) and GPT-5.2-chat-latest as the teacher.

	AIME24	AIME25	HMMT (Feb.)	HMMT (Nov.)	Avg
Gemma3-4B (base)	6.67	12.92	1.67	6.25	6.88
OVD (Xiong et al., 2026)	7.38	13.00	2.05	6.36	7.20
GAD (Ye et al., 2026)	6.92	12.50	1.83	6.08	6.83
ROPD (ours)	**10.00** $_{+3.33}$	**13.72** $_{+0.80}$	**2.92** $_{+1.25}$	**6.88** $_{+0.63}$	**8.38** $_{+1.50}$

Algorithm 1 ROPD: Black-box On-policy Distillation via On-policy Rubrics

Input: Dataset $\mathcal{D}$, teacher model $\mathcal{T}$, Rubricator $\mathcal{R}$, Verifier $\mathcal{V}$, student policy π_θ (initialized from π_{ref})
Hyperparameters: teacher answers m, student rollouts n, rubric criteria count K, clip range ϵ_{clip}, learning rate η, training steps N
Output: Trained student policy π_θ
for step $= 1$ to N **do**
  Sample a mini-batch of questions $\{x^{(1)}, \ldots, x^{(B)}\} \sim \mathcal{D}$
  Initialize gradient accumulator $\Delta\theta \leftarrow 0$
  for each question x in the mini-batch **do**
    // Step 1: Collect multi-teacher answers
    $\mathcal{Y}^T \leftarrow \{\, \mathcal{T}(x) \text{ sampled } m \text{ times} \,\}$ ▷ m teacher responses
    // Step 2: On-policy student rollout
    $\mathcal{Y}^S \leftarrow \{\, y_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid x) \,\}_{i=1}^{n}$ ▷ n student responses
    // Step 3: Rubricator generates shared rubrics
    $R_x \leftarrow \mathcal{R}(x,\ \mathcal{Y}^T,\ \mathcal{Y}^S)$ ▷ K criteria $\{c_k\}$ with weights $\{w_k\}$
    // Step 4: Verifier scores each student rollout
    for $i = 1$ to n **do**
      $\{v_{i,k}\}_{k=1}^{K} \leftarrow \mathcal{V}(x,\ y_i,\ R_x)$ ▷ $v_{i,k} \in \{0, 1\}$ — binary judgements
      $r_i \leftarrow \dfrac{\sum_{k=1}^{K} w_k \cdot v_{i,k}}{\sum_{k=1}^{K} w_k}$ ▷ Weighted pass rate $\in [0, 1]$
    end for
    // Step 5: Group-relative advantage (GRPO)
    $\bar{r} \leftarrow \frac{1}{n}\sum_{i=1}^{n} r_i,\ \sigma_r \leftarrow \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2} + \epsilon$
    for $i = 1$ to n **do**
      $A_i \leftarrow (r_i - \bar{r}) \,/\, \sigma_r$
    end for
    // Step 6: Accumulate per-question policy gradient
    $\Delta\theta \leftarrow \Delta\theta + \nabla_\theta \frac{1}{n}\sum_{i=1}^{n} \min\Big(\rho_i(\theta)A_i,\ \text{clip}\big(\rho_i(\theta),\ 1-\epsilon_{\text{clip}},\ 1+\epsilon_{\text{clip}}\big)A_i\Big]$
  end for
  // Step 7: Update policy parameters
  $\theta \leftarrow \theta + \eta \cdot \Delta\theta$
end for
return π_θ