# Rubric-based On-policy Distillation

**Junfeng Fang**[1], **Zhepei Hong**[2]*, **Mao Zheng**[3], **Mingyang Song**[3], **Gengsheng Li**[3], **Houcheng Jiang**[2], **Dan Zhang**[1], **Haiyun Guo**[1], **Xiang Wang**[2]†, **Tat-Seng Chua**[1]
[1]National University of Singapore, [2]University of Science and Technology of China, [3]Tencent
fangjf1997@gmail.com, hongzhepei@gmail.com

## Abstract

On-policy distillation (OPD) is a powerful paradigm for model alignment, yet its reliance on teacher logits restricts its application to white-box scenarios. We contend that structured semantic rubrics can serve as a scalable alternative to teacher logits, enabling OPD using only teacher-generated responses. To prove it, we introduce ROPD, a simple yet foundational framework for rubric-based OPD. Specifically, ROPD induces prompt-specific rubrics from teacher-student contrasts, and then utilizes these rubrics to score the student rollouts for on-policy optimization. Empirically, ROPD outperforms the advanced logit-based OPD methods across most scenarios, and achieving up to a 10× gain in sample efficiency. These results position rubric-based OPD as a flexible, black-box-compatible alternative to the prevailing logit-based OPD, offering a simple yet strong baseline for scalable distillation across proprietary and open-source LLMs. Code is available at https://github.com/Peregrine123/ROPD_official.

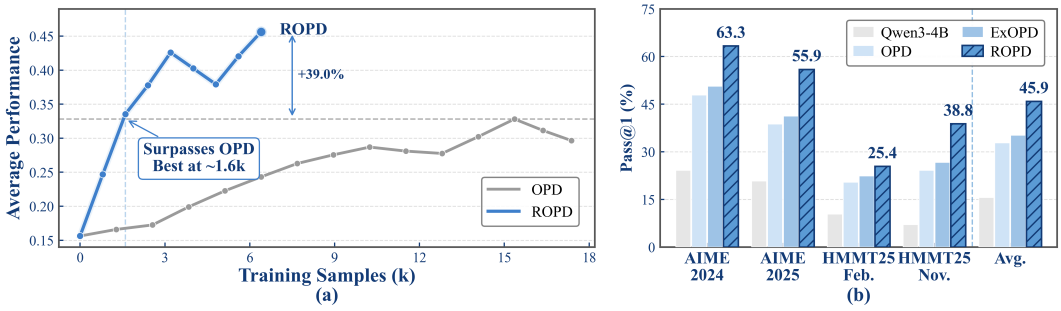


Figure 1: **ROPD efficiency and reasoning performance.** (a) Training dynamics averaged over four math benchmarks (*i.e.*, AIME 24/25 [1, 2] and HMMT 25 Feb./Nov. [3]). ROPD achieves a 10× sample efficiency boost. (b) Comparative results. For fair comparison, all models are trained on DAPO-Math-17K [4] using Qwen3-4B [5] (student) and Qwen3-30B-A3B [5] (teacher). See Section 3.1 for comprehensive experimental settings.

## 1 Introduction

The rapid evolution of Large Language Models (LLMs) has established On-Policy Distillation (OPD) as an essential paradigm for post-training and model alignment [6, 7]. By leveraging the teacher's output logits as a dense supervisory signal, OPD allows the student model to learn from its own rollout distribution [8]. This paradigm has demonstrated remarkable efficacy in transferring

*Equal contribution.
†Corresponding author: xiangwang1123@gmail.com

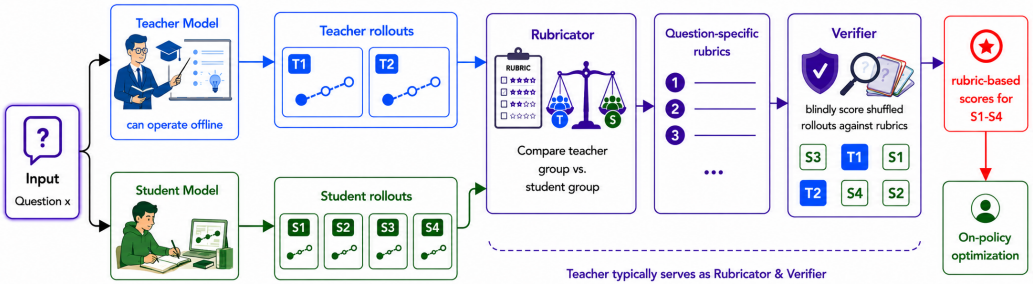


Figure 2: **The ROPD Pipeline.** A *Rubricator* induces prompt-specific rubrics by contrasting teacher and student rollouts, which a *Verifier* then utilizes to provide rewards for on-policy optimization.

complex reasoning capabilities and has become a standard practice in the development of advancing open-source models [5, 9, 10].

However, the above logit-based OPD is fundamentally tied to a "white-box" setting, requiring access to the teacher's full output logits [8]. This dependency restricts distillation to open-source models, rendering high-performance proprietary models inaccessible as teachers. This naturally raises the question: *can we retain the core on-policy nature of OPD without relying on logit-based signals?* Inspired by the recent success of rubric-based post-training, this work investigates a complementary path: **rubric-based OPD**, which seeks to provide distillation signals based on on-policy rubrics.

To demonstrate the potential of this paradigm, we establish ROPD, a simple and foundational instantiation of rubric-based OPD. As shown in Figure 2, for each question, a *Rubricator* first contrasts teacher and student rollouts to synthesize prompt-specific rubrics, and a *Verifier* then scores student rollouts against these rubrics to guide on-policy optimization. To streamline the design, the teacher model typically assumes both roles. Although the framework is deliberately simple, our empirical analysis in Section 4 reveals several non-trivial design principles foundational to ROPD. For example, the *Verifier* should blindly score both teacher and student rollouts together to calibrate bias arise from varying question difficulties. These findings suggest that rubric-based OPD is not merely a heuristic replacement for logit-based OPD, but a principled and robust distillation framework.

We extensively validate ROPD across diverse benchmarks (*e.g.*, AIME24/25 [1, 2], HMMT25 [3], GPQA-Diamond [11], HealthBench [12], and IFEval [13]) and model configurations (*e.g.*, Qwen3-4B [5] and Gemma3-4B [14] students with GPT-5.2 [15] and Qwen3-30B [5] teachers). In black-box settings, ROPD consistently outperforms existing black-box distillation methods, setting a new performance frontier (Table 1). More remarkably, in white-box settings, ROPD remains highly competitive with, and often surpasses, advancing logit-based OPD methods, despite never accessing teacher logits (Figure 1, Table 2). These results demonstrate that for complex reasoning tasks, **rubric-based signals can serve as a flexible alternative to logit-based signals**.

The advantages of the ROPD paradigm extend far beyond its inherent flexibility (*e.g.*, supporting cross-architecture distillation without tokenizer alignment). **Conceptually,** ROPD functions as a semantic filter: while token-level logits often reflect stochastic phrasing variations that offer negligible value for distillation [16], ROPD isolates task-level reasoning principles by distilling behavioral gaps into structured rubrics. This shift from logit-matching to semantic guidance yields a profound empirical gain: up to a **10× boost in sample efficiency** (Figure 1 (a)). **Architecturally,** the teacher's independence from the training loop enables offline execution, significantly lowering GPU memory overhead and accelerating training process (Figure 3). **Optimization-wise,** ROPD exhibits superior robustness to model divergence: while logit-based OPD typically requires the teacher and student to share similar reasoning patterns [17], ROPD's high-level semantic guidance ensures stable convergence even across models with markedly disparate reasoning trajectories (Table 3).

In summary, this work offers a complementary perspective to the prevailing logit-centric distillation landscape. Through ROPD, a simple framework requiring minimal hyperparameter, we demonstrate that high-level semantic rubrics can serve as an efficient and robust alternative to fine-grained logits. Our findings suggest that the future of OPD may lie not only in the refinement of denser numerical signals, but also in the extraction of clearer semantic guidance. By reconciling performance, efficiency, and accessibility, ROPD establishes a versatile baseline that paves the way for *scalable and interpretable distillation* in the ever-evolving system of both proprietary and open-source LLMs.

## 2 Method

### 2.1 Problem Setup

On-policy distillation facilitates knowledge transfer by supervising a student model on its self-generated trajectories [18]. Let $x$ denote an input prompt, $\pi_T$ a teacher model, and $\pi_\theta$ a trainable student policy. Traditional white-box OPD typically relies on the teacher's internal states, leveraging the next-token distribution $p_{\mathcal{T}}(\cdot \mid x, y_{<t})$ to provide dense supervision for the prompt $x$ and student prefix $y_{<t}$ [8, 6]. However, such access is often unrealistic for proprietary or API-governed teachers. In response, black-box OPD assumes teacher-side distributions are inaccessible [18]. For each prompt $x$, the student generates a rollout $y \sim \pi_\theta(\cdot \mid x)$ and obtains evaluative feedback from the teacher on this output. This feedback serves as the supervisory signal, abstracting teacher-side observations into rewards to guide the student's policy optimization. The core objective of black-box OPD is thus to design an effective reward function that faithfully distills the teacher's capabilities using only discrete textual interactions.

### 2.2 Rubric-based On-policy Distillation

ROPD instantiates black-box OPD by distilling textual teacher responses into structured, prompt-specific rubrics for student reward computation. As illustrated in Figure 2, the framework operates in two stages: (1) Rubric Induction, which extracts a common set of criteria from teacher and student responses, and (2) Rubric-based Verification, which evaluates student rollouts against these criteria to compute rewards for policy optimization. **Rubric Induction.** Given a prompt $x$, we first collect a set of teacher responses $\mathcal{Y}_x^T = \{y_j^T\}_{j=1}^m$ and student rollouts $\mathcal{Y}_x^S = \{y_i^S\}_{i=1}^n$ sampled from $\pi_t$ and $\pi_\theta$, respectively:

$$y_j^T \sim \pi_t(\cdot \mid x), \quad y_i^S \sim \pi_\theta(\cdot \mid x). \tag{1}$$

Here, $\mathcal{Y}_x^T$ provides high-level evidence of desirable solution strategies. We then employ a Rubricator to convert the teacher responses and student rollouts into a set of prompt-specific rubrics:

$$\mathcal{C}_x = \text{Rubricator}(x, \mathcal{Y}_x^T, \mathcal{Y}_x^S) = \{c_k\}_{k=1}^K, \tag{2}$$

where each rubric item $c_k = (\rho_k, w_k)$ consists of a textual criterion $\rho_k$ and its importance weight $w_k > 0$. Crucially, $\mathcal{C}_x$ is shared across all $n$ student rollouts for the same prompt, ensuring that the reward signal remains consistent within the rollout group — a property particularly beneficial for group-based optimization methods like GRPO [19]. **Rubric-based Verification.** With the induced rubric set $\mathcal{C}_x$, the Verifier evaluates each student rollout against every rubric item. For the $i$-th student rollout and the $k$-th criterion, we define

$$v_{i,k} = \text{Verifier}\big(x, y_i^S, c_k; \mathcal{Y}_x^T, \mathcal{Y}_x^S\big), \qquad v_{i,k} \in \{0, 1\}, \tag{3}$$

where $v_{i,k} = 1$ indicates that $y_i^S$ satisfies criterion $\rho_k$, and $v_{i,k} = 0$ otherwise. The response-level score is computed as the weighted pass rate:

$$s_i = \frac{\sum_{k=1}^K w_k v_{i,k}}{\sum_{k=1}^K w_k + \epsilon}, \tag{4}$$

where $\epsilon$ is a small constant for numerical stability. ROPD uses this verified score as the reward for on-policy optimization (see details in Appendix F). In our experiments, the teacher model typically assumes the roles of both Rubricator and Verifier. We also validate that replacing them with an auxiliary LLM has a marginal impact on final results, demonstrating the **flexibility** of our paradigm.

**Roadmap.** The remainder of this paper is structured to provide both empirical validation and mechanistic insight. Section 3 presents a comprehensive evaluation of ROPD across black-box and white-box distillation scenarios. Section 4 then interrogates the underlying drivers of performance, providing a deep dive into why rubrics surpass traditional logit-based signals. Finally, Section 5 situates ROPD within the broader landscape of on-policy distillation and alignment research.

Table 1: Performance comparison against black-box distillation baselines. 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 |
| **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 [20] | 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 [21] | 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** $_{+0.07}$ |
| **Thinking** | | | | | | | |
| 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 [20] | 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 [21] | 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** $_{+0.49}$ |

# 3 Main Result

## 3.1 Setup

**Models.** We employ Qwen3-4B [5] as our primary student model. To evaluate cross-architecture generalization, we further adopt Gemma3-4B-it [14] as the student in Section 3.5. *Black-box setting (Table 1).* The teacher is GPT-5.2-chat-latest [15] accessed via API. We compare ROPD with SFT (with static teacher outputs), T-Judge (directly employing the teacher as a judge to provide scores), and representative black-box distillation methods OVD [20] and GAD [21]. *White-box Setting.* Using Qwen3-30B-A3B [5] as the open-weight teacher, we compare ROPD with advanced logit-based methods OPD [6, 7] (hereafter LOPD) and ExOPD [22]. All experiments are conducted in *non-thinking* mode. Crucially, ROPD only accesses teacher text, intentionally ignoring available logit information to demonstrate its black-box robustness. **Data.** Training is conducted on DAPO-Math-17K [4] for math, and RaR-Science/Medical-20K [23] for science and medical tracks. For fair comparison, all methods share the same training samples within each domain. The SFT baseline employs pre-sampled teacher responses as static supervision. **Training.** We employ GRPO across all RL methods with a learning rate of $10^{-6}$, batch size of 32, and $n = 8$ rollouts per prompt (1 epoch). ROPD-specific parameters include $m = 4$ teacher references and $K \in [4, 12]$ rubric items. To maintain a streamlined pipeline, the teacher model acts as both the Rubricator and Verifier. Checkpoints are selected via a validation suite comprising AIME24, GPQA-Diamond, and HealthBench. See Appendix C for the complete hyperparameter list. **Evaluation.** We evaluate our models on AIME 24/25 [1, 2], HMMT 25 [3], GPQA-Diamond [11], and HealthBench [12], with IFEval [13] serving as an out-of-domain probe. For all experiments, we sample $k = 16$ responses using a temperature of 1.0 and top-$p$ of 0.95, capped at $32,768$ tokens. Teacher evaluation follows the same protocol. Full evaluation details are provided in Appendix C.

## 3.2 Performance in Black-Box Scenarios

Table 1 summarizes the Pass@1 performance across all benchmarks. ROPD consistently ranks first across all 14 benchmark configurations. Notably, on AIME25 (thinking), ROPD (68.75) transcends the GPT-5.2-chat-latest teacher (67.08), indicating that rubric-augmented optimization facilitates the elicitation of reasoning capabilities that surpass mere teacher imitation. The most substantial gains are observed on the most challenging benchmark HMMT25 (Nov.), where ROPD elevates the base model's score from 7.08 to 41.67, achieving a +34.6 absolute improvement. Furthermore, on IFEval, ROPD exhibits slight improvements over the base model, confirming that rubric-based distillation preserves broad instruction-following alignment without incurring catastrophic forgetting of out-of-domain capabilities.

Table 2: Performance comparison against white-box distillation baselines. All results are reported in Pass@1 (%). Bold and underline indicate the best and second-best performance, respectively.

| | 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 [6, 7] | logit | 47.92 $_{+23.8}$ | 38.75 $_{+17.9}$ | 20.42 $_{+10.0}$ | 24.17 $_{+17.1}$ | 32.82 $_{+17.2}$ |
| ExOPD [22] | 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** | **text** | **63.33** $_{+39.2}$ | **55.93** $_{+35.1}$ | **25.40** $_{+15.0}$ | **38.80** $_{+31.7}$ | **45.87** $_{+30.2}$ |

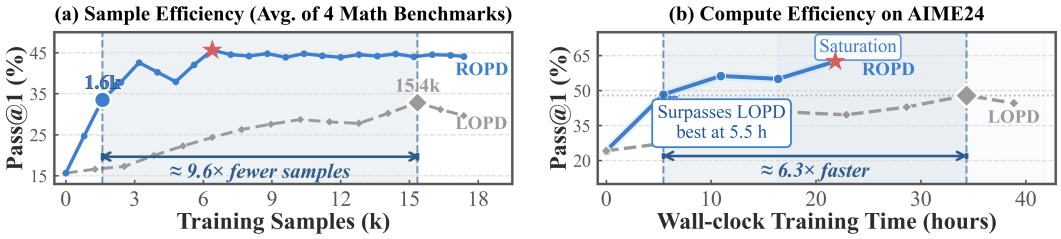


Figure 3: ROPD efficiency advantage over LOPD (Qwen3-30B-A3B teacher and Qwen3-4B student, non-thinking). (a) Average sample efficiency. ROPD recovers LOPD's best performance with $\sim$9.6$\times$ fewer samples (1.6k *vs.* 15.4k); the star ($\star$) marks its own performance plateau at 6.4k. (b) Compute efficiency on AIME24. ROPD yields a $\sim$6.3$\times$ wall-clock speedup, demonstrating that its superior sample efficiency far outweighs the increased per-step computational overhead.

### 3.3 Performance in White-Box Scenarios

Table 2 exhibits the Pass@1 performance in white-box scenarios. Despite its text-only constraints, ROPD consistently outperforms the white-box baselines. Specifically, while LOPD bridges only 42.1% of the student-teacher gap, ROPD closes 74.1% of the same interval — a 1.8$\times$ improvement achieved with significantly restricted information. Furthermore, the marginal gains from SFT confirm that static supervision is insufficient for complex reasoning tasks. While ExOPD improves upon LOPD through reward extrapolation, ROPD still maintains a +10.6 point lead, suggesting that refining reward architecture could yield higher returns than optimizing reward magnitude. More experimental results and case studies are exhibited in Appendix B and E. Why does black-box rubric supervision surpass dense, white-box logits? LOPD's token-level signals provide dense, per-token feedback, but this signal measures distributional similarity rather than *correctness* — a student can closely match the teacher's token distribution while producing an incorrect answer. ROPD's rubrics, by contrast, decompose response quality into discrete, verifiable criteria, providing *outcome-oriented* feedback that directly targets answer correctness. The result is that ROPD's signal, though derived from less teacher information, is more effective for complex reasoning tasks. A detailed mechanical exploration of this phenomenon follows in Section 4.

### 3.4 Efficiency and Convergence Analysis

As shown in Figure 3, ROPD significantly outperforms LOPD in data efficiency, achieving 48.3% on AIME24 with an order of magnitude fewer samples (1.6k *vs.* 15.4k). Despite a higher per-step computational overhead introduced by the *Rubricator* and the *Verifier*, ROPD yields a 6.3$\times$ wall-clock speedup to reach the same performance threshold (5.5h *vs.* 34.4h). Notably, ROPD exhibits superior generalization stability: unlike LOPD, which suffers from post-saturation degradation, ROPD remains robust throughout training. These results, obtained under identical hardware and teacher (*i.e.*, Qwen3-30B-A3B) constraints, underscore the information density of rubric-based rewards.

Table 3: Cross-architecture generalization performance. Results are reported as Pass@1 (%) using 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 [20] | 7.38 | 13.00 | 2.05 | 6.36 | 7.20 |
| GAD [21] | 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** $_{+1.50}$ |

### 3.5 Cross-Architecture Generalization

As demonstrated in Table 3, ROPD exhibits robust cross-architecture transferability. To test the limits of our framework, we substitute the Qwen3-4B student with the significantly less capable Gemma3-it-4B (which scores only 6.67% on AIME24 compared to Qwen3's 24.17%). Maintaining identical experimental conditions, ROPD consistently elevates performance above the base model, *e.g.*, AIME24 performance rises to 10.00% (a +50% relative improvement). These results show that ROPD's criterion-referenced rubrics provide an absolute supervisory signal that remains informative even for low-quality responses. ROPD thus circumvents the inherent quality bottleneck, remaining effective under both architectural shifts and extremely low-resource starting policies.

## 4 Analysis

Having established ROPD's empirical effectiveness, we now interrogate the mechanisms underlying its success. We begin with a qualitative case study illustrating how rubric-based rewards achieve superior discriminative power over scalar judges (Section 4.1). We then quantify the alignment between reward signals and ground-truth correctness, illustrating the transition from logit mimicry to rubric-based optimization (Section 4.2). Finally, we ablate the core design choices to confirm the necessity of each reward component (Section 4.3).

### 4.1 Case Study: Rubric *vs.* Scalar Judge

To elucidate why ROPD outperforms scalar supervision, we analyze a representative case in Table 4 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, which identifies the correct conclusion but lacks the general parity proof (C2 false), and Rollout C, which fabricates a derivation to guess 337, passing only the formatting check (C1). While the rubric provides a stark separation between the two (0.77 *vs.* 0.23, $\Delta = 0.54$), the scalar judge barely distinguishes them (0.70 *vs.* 0.55, $\Delta = 0.15$), visibly swayed by Rollout C's superficial fluency. This $3.6\times$ wider margin is a structural advantage: scalar judges compress disparate quality dimensions into a single value, allowing "passable" formatting to dilute substantive logical failure. Conversely, the rubric decouples evaluation dimensions (*e.g.*, factorization (C3), coherence (C4), and factual accuracy (C5)) preventing fabricated derivations from hiding behind well-structured prose. Within the GRPO framework, this fine-grained discrimination ensures that the reward signal prioritizes substantive reasoning over stylistic mimicry, a property that translates into measurable per-criterion gains during training (see Section 4.2).

### 4.2 Mechanism: Why Rubric Rewards Transcend Teacher Logit

To unpack ROPD's empirical success, we now investigate the *informativeness paradox*: why do restricted rubric signals surpass dense logit-based supervision? We analyze signal reliability and training dynamics using a controlled pool of 3,120 AIME24 rollouts, evaluating (1) rubric rewards, (2) teacher logits, and (3) top-24 token overlap relative to ground-truth correctness. For a comprehensive breakdown of these results, see Appendix E. **Logit is a Misaligned Proxy for Correctness.** While LOPD treats teacher likelihood as a quality proxy, our analysis in Figure 4 (a) reveals a staggering inverse correlation: rubric rewards achieve 0.90 AUC versus the teacher's near-random 0.35. This inverse correlation indicates that logit often rewards fluent but logically flawed paths than correct but stylistically novel ones. As shown in Figure 5 (b), ROPD consistently generates more discriminative

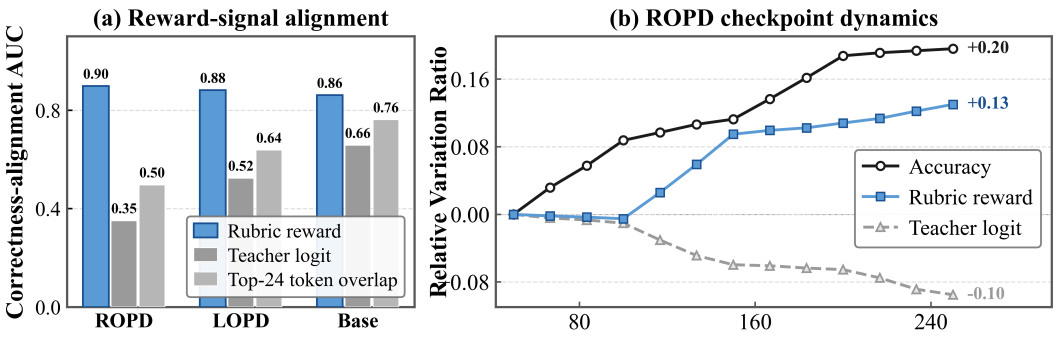


Figure 4: Reward signal alignment with correctness (AIME24). (a) Correctness-alignment AUC for rubric reward, teacher logit, and top-24 overlap across different rollout pools. (b) Training trajectories: ROPD accuracy and rubric reward scale synchronously, while teacher logit exhibits a divergent downward trend. The x-axis represents the training steps.

Table 4: 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 |

advantage signals across the majority of prompts. By filtering out the "stochastic noise" of token-level logit distributions, ROPD ensures the optimizer prioritizes logical fidelity over surface-form mimicry. **Mimicry for Understanding, Divergence for Transcendence.** The training trajectories reveal a fascinating "phase shift" in how ROPD utilizes teacher knowledge. Figure 5 (a) shows that in the earliest stages, ROPD's token overlap surges even faster than LOPD's, suggesting that rubrics effectively codify the teacher's basic formatting and linguistic norms. However, as shown in Figure 5 (a) and 4 (b), a sharp divergence soon follows: while LOPD remains trapped in logit mimicry, ROPD's accuracy and rubric rewards scale synchronously while its logit actively declines. This confirms a pivotal insight: **ROPD uses the teacher as a springboard, not a mirror.** Once the student masters the teacher's reasoning "language", it transcends the teacher's specific token distribution to seek higher-order correctness. **Decoupled Supervision as a Precision Anchor.** Why is ROPD's progress so stable? Table 5 breaks down the pass rates across three rubric categories, where ROPD achieves superior pass rate gains ($\Delta$) in every dimension. By decomposing quality into independent, verifiable milestones, ROPD enables granular credit assignment. Unlike LOPD's entangled logits, ROPD's per-rubric rewards facilitate directional advancement: the optimizer can explicitly penalize specific failures (*e.g.*, calculation errors) without eroding previously mastered milestones. Detailed transitions in Table A3 reveal a 15.9% regressed pass rate for LOPD, confirming that monolithic scalar signals suffer from inter-dimensional interference where improving one facet often erodes another.

### 4.3 Ablation Study: Deconstructing the Reward Signal

ROPD's performance is predicated on three key design choices: multi-teacher seeding, cross-rollout rubric sharing, and blind verification. Table 6 presents a leave-one-out ablation. Specifically,

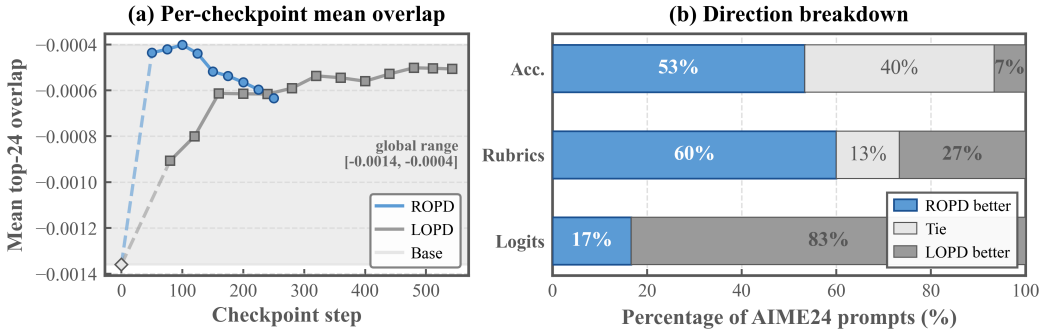


Figure 5: Evolution of stylistic mimicry and comparative performance. (a) Mimicry Trajectories: Per-checkpoint mean top-24 token overlap; ROPD rapidly saturates stylistic alignment before pivoting toward reasoning correctness, whereas LOPD exhibits persistent, monotonic mimicry of the teacher's distribution. (b) Prompt-wise Comparative Advantage: Head-to-head breakdown on AIME24; ROPD outperforms LOPD in reasoning accuracy and rubric satisfaction across the majority of prompts, while LOPD's advantage is largely confined to mimicking teacher logit distributions.

Table 5: Comparative rubric-level pass rates (ROPD *vs.* LOPD). Rubric-wise performance 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** |

Table 6: Leave-one-out reward-component ablation on AIME24. Pass@1 (%) under non-think; $m$ denotes the number of teacher rollouts.

| Reward Design | $m$ | AIME24 |
|---|---|---|
| Qwen3-4B (base) | – | 24.17 |
| w/o multi-teacher (single answer) | 1 | 47.08 $_{+22.91}$ |
| w/o sharing (per-student rubrics) | 4 | 61.25 $_{+37.08}$ |
| w/o blind scoring (verifier sees teacher) | 4 | 61.75 $_{+37.58}$ |
| **Full ROPD** | **4** | **65.02** $_{+40.85}$ |

- **Multi-teacher coverage is the primary performance driver.** Transitioning from $m = 4$ to $m = 1$ causes a catastrophic 17.9 point drop in Pass@1. A single teacher answer over-anchors the rubric to a specific solution trajectory, causing criteria to collapse into "path-matching" rather than "correctness-checking". By contrast, diverse teacher strategies empower the Rubricator to induce generalizable criteria that reward logical validity regardless of the specific reasoning path.
- **Sharing aggregates cross-rollout contrast.** Utilizing a single shared rubric per prompt (rather than one per {teacher, student} pair) yields a +3.75 point gain. This global view allows the rubric to surface systematic reasoning gaps shared across the rollout distribution, which are invisible to per-pair rubrics isolated from the wider group dynamics.
- **Blind scoring prevents identity-driven bias while preserving the reward spread.** Revealing identities costs 3.25 points. However, retaining teacher responses in the blind pool is essential as a difficulty anchor. Evaluating students in a vacuum often causes the Verifier to collapse toward mean scores regardless of task complexity. The teacher's presence ensures the reward distribution remains properly calibrated across diverse problem difficulties, maintaining the discriminative power of GRPO advantages.

## 5 Related Work

**On-policy Distillation.** OPD has become a promising post-training paradigm that replaces sparse rewards with dense feedback on student-generated trajectories, thereby not only mitigating exposure bias but also improving sample efficiency [8, 6, 7, 18]. Existing work strengthens OPD from several angles, including objective design and reward extrapolation [24, 22], training efficiency and signal calibration [25, 26, 27, 16, 28], cross-tokenizer distillation [29], and empirical analyses of failure modes and practical recipes [17, 30]. Frontier open-source models have also adopted OPD as a key component of post-training [5, 9, 10]. Despite this progress, the dominant line still assumes dense teacher probabilities or aligned token spaces, limiting proprietary-teacher and cross-architecture distillation. 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 Distillation.** Recent black-box methods use various response-level signals: ORPO-Distill constructs preference pairs from mixed-policy traces [31]; GAD trains a discriminator for co-evolving rewards [21]; OVD uses discrete verbal trajectory scores [20]; and RL-based KD trains from scalar evaluator rewards [32]. Their signals remain largely implicit: preferences compare whole traces, while discriminators hide criteria behind learned scores. 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 achieved significant breakthroughs in reasoning [19], yet its reliance on binary outcomes often restricts it to deterministic domains. To bridge this gap, structured rubrics have been introduced to decompose quality into fine-grained dimensions for open-ended tasks. While RaR [23] and OpenRubrics [33] focused on formalizing instance-specific rewards, Rubicon [34] addressed the "seesaw effects" between conflicting criteria. More recent works like RLER [35] and SibylSense [36] have pioneered evolving rubrics grounded in search evidence or adversarial memory to capture emergent behaviors. While prior work treats rubrics as evaluation instruments, ROPD repurposes them as a dynamic distillation interface.

## 6 Limitation and Future Work

While ROPD demonstrates the efficacy and flexibility of rubric-based rewards for OPD, we identify two primary limitations. **First,** our evaluation mainly focuses on formal reasoning, such as Mathematics, Medicine, and Science. Although IFEval results indicate that general instruction-following is preserved, the performance of rubric-based OPD in subjective or creative tasks remains to be established. **Second,** ROPD depends on the instruction-following of the Rubricator and Verifier. Our preliminary results show that ROPD remains robust even when these components are replaced with alternative models — likely due to the asymmetry between evaluation and generation: verifying a solution's integrity is inherently simpler than its derivation. Despite this resilience, its reliance on such meta-evaluation components calls for broader validation across diverse model architectures. More broadly, these limitations point to a larger research opportunity. If logit-based OPD treats distillation as token-level imitation, rubric-based OPD reframes it as the transfer of structured semantic principles. Understanding how to design, validate, and calibrate such principles may be essential for **scalable distillation**, especially as frontier models become increasingly opaque and heterogeneous. We hope ROPD provides a simple starting point for this direction.

## 7 Conclusion

In this work, we introduce ROPD, a minimalist yet potent framework for rubric-based OPD. By shifting the supervisory signal from probabilities to high-level rubrics, ROPD reconciles competitive performance with accessibility. ROPD not only achieves a $10\times$ boost in data utilization efficiency but also exhibits superior robustness across disparate model capabilities. These findings suggest that the future of OPD may lie in the cultivation of clearer semantic guidance rather than solely in the pursuit of denser numerical signals. As a versatile and scalable baseline, ROPD paves the way for efficient and interpretable distillation in the era of increasingly opaque, high-performance LLMs.

## References


[1] MAA. Aime 2024: American invitational mathematics examination, 2024.

[2] MAA. Aime 2025: American invitational mathematics examination, 2025.

[3] HMMT. Hmmt 2025: Harvard-mit mathematics tournament, 2025.

[4] 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.

[5] 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.

[6] 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.

[7] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025.

[8] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] Gemma Team, Google DeepMind. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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

[16] 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*, 2026.

[17] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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*, 2026.

[28] 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.

[29] 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.

[30] 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.

[31] Aasheesh Singh, Vishal Vaddina, and Dagnachew Birru. Orpo-distill: Mixed-policy preference optimization for cross-architecture llm distillation. *arXiv preprint arXiv:2509.25100*, 2025.

[32] Yiyang Shen, Lifu Tu, and Weiran Wang. Reinforcement learning-based knowledge distillation with llm-as-a-judge. *arXiv preprint arXiv:2604.02621*, 2026.

[33] 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.

[34] 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.

[35] 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.

[36] 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*, 2026.

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

[38] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

# Appendix

## Appendix Overview

# A 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 [37], and Sequence-Level Knowledge Distillation (SeqKD) extends this to generation by substituting teacher-decoded sequences for label-level targets [38]. 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 [8], Generalized Knowledge Distillation (GKD) learns from self-generated mistakes with teacher feedback [6], and recent work scales this recipe to reasoning post-training [5, 17]. 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 [31]; GAD trains a discriminator to distinguish teacher from student responses and uses its score as a co-evolving reward [21]; On-policy Verbal Distillation (OVD) asks the teacher for discrete verbal trajectory scores, avoiding token alignment and reducing memory cost [20]; and RL-based KD with LLM-as-a-Judge trains from scalar evaluator rewards over unlabeled data [32]. 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 [19], 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 [23]. On rubric generation, OpenRubrics scales synthesis via contrastive prompting [33]. 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 [34]. 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 [35], while SibylSense pursues adversarial memory tuning [36]. 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.

## B 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 primarily in early training (steps $< 1k$) and it self-corrects as the Verifier is prompted with explicit correctness checks.

**Rubric item examples.** Table A1 shows representative rubric items generated by the Rubricator for different prompt types.

Table A1: **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"<br>"Intermediate steps are explicitly justified with theorems or algebraic rules"<br>"The final answer is boxed and matches the required format"<br>"No arithmetic errors in the numerical computation chain" |
| Science (GPQA) | "The answer identifies the relevant physical/chemical principle"<br>"Quantitative reasoning includes correct unit conversions"<br>"Alternative hypotheses are considered and ruled out"<br>"The conclusion explicitly addresses the question asked" |
| Medicine (HealthBench) | "Diagnosis is supported by specific findings from the case description"<br>"Differential diagnosis lists at least 2 alternative conditions"<br>"Treatment recommendation follows guideline-concordant reasoning"<br>"Referral or follow-up plan is specified when appropriate" |

## C Hyperparameters and Training Configuration

**Complete hyperparameter specification.** Table A2 lists all hyperparameters used in ROPD experiments.

Table A2: **Complete hyperparameter configuration.**

| 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 | $1 \times 10^{-6}$ | $1 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| 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 ($\beta_1$, $\beta_2$) | (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 tokens (no-think / think) | 8192 | 8192 | 8192 |
| Teacher temperature | 0.0 | 0.0 | 0.0 |
| 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 500 steps on the validation split and select the best checkpoint based on AIME24 pass@1 (math track), GPQA-Diamond pass@1 (science track), and HealthBench pass@1 (medical track). 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.

# D 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:

$$R(y_j^S) = \underbrace{\frac{\sum_{k=1}^{K} w_k \cdot \mathbb{I}[\text{pass}_k]}{\sum_{k=1}^{K} w_k}}_{\text{weighted pass rate}} \tag{5}$$

where group-relative advantage is normalized per prompt.

## E Additional Figures and Analysis

**Leaderboard bar chart (think mode).** Figure A1 shows the leaderboard-style comparison under think decoding.

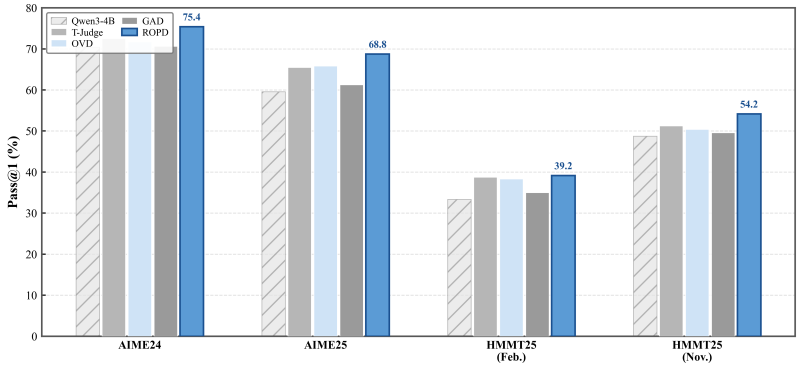


Figure A1: **Leaderboard comparison – think mode.** Horizontal bar chart in DeepSeek-v4 leaderboard style.

**Leaderboard bar chart (no-think mode).** Figure A2 shows the leaderboard-style comparison under no-think decoding.

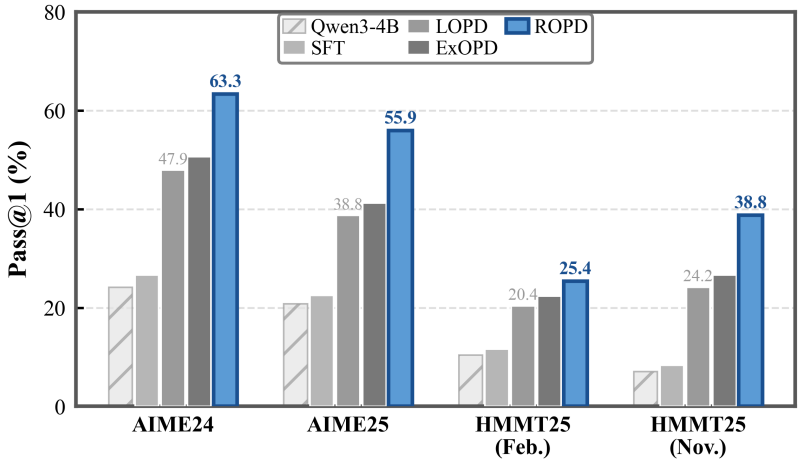


Figure A2: **Leaderboard comparison – no-think mode.** Horizontal bar chart in DeepSeek-v4 leaderboard style.

**Per-criterion transition: ROPD vs. LOPD.** Table A3 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 A4 and A5 provide the complete numerical results underlying that analysis. Figures A4–A7 visualize the checkpoint-level dynamics, correctness-conditioned signal distributions, final-checkpoint paired comparison, and top-24 overlap saturation.

**Analysis pool protocol.** All numbers in this subsection are computed on a dedicated offline analysis pool consisting of 30 AIME24 prompts $\times$ 8 rollouts $\times$ 13 checkpoints (5 ROPD, 7 LOPD, 1 Base) $=$ 3,120 responses. Rollouts are sampled independently of the main benchmark evaluation (*i.e.*,

Table A3: **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 |

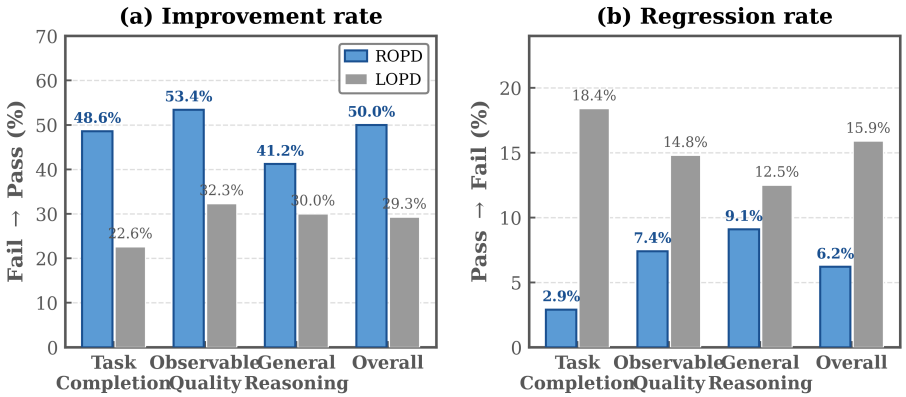


Figure A3: **Cell-level transition comparison: ROPD vs. LOPD.** (a) Improvement rate: fraction of initially-failed cells (q<0.5) that become passed (q≥0.5) at the final checkpoint. (b) Regression rate: fraction of initially-passed cells that become failed. ROPD improves more and regresses less in every rubric category.

this is not a subset of the $k$=16 rollouts behind Tables 1–2 and Figures3), 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. Because rollouts are an independent $k$=8 sample, the accuracy column `Acc.` in Table A5 can differ from the main $k$=16 benchmark by up to ∼5 points at unstable early checkpoints (*e.g.*, ROPD step 50: 43.75% here vs. 48.33% in Figure 3); 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 A4: **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 |
|---|---|---|---|---|---|---|---|
| | | | $\text{AUC}_{\text{all}}$ | Bad-upd. | $\text{AUC}_{\text{all}}$ | Bad-upd. | $\text{AUC}_{\text{all}}$ |
| 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 A5: **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. `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 |

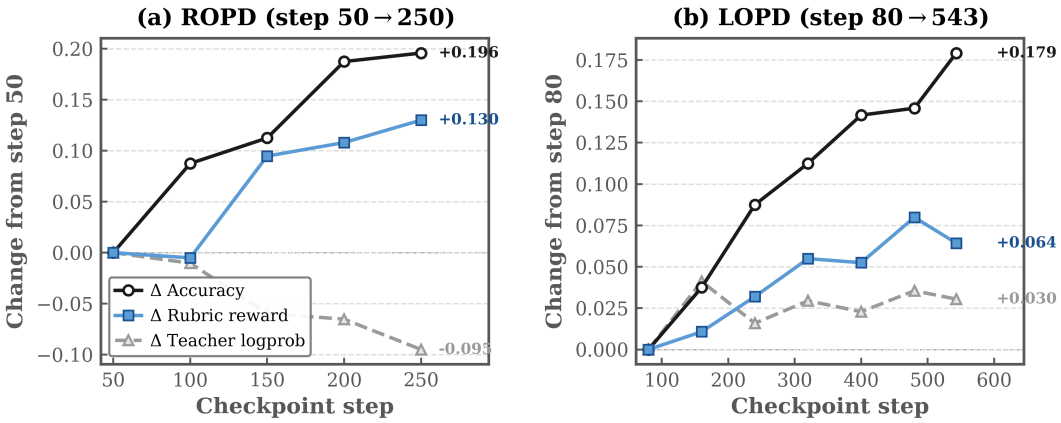


Figure A4: **Checkpoint dynamics: relative change from earliest checkpoint.** ROPD (left) and LOPD (right). Accuracy and rubric reward are normalized relative to their values at the first checkpoint; teacher log-likelihood is shown on the same relative scale. ROPD's accuracy and rubric reward rise together while teacher likelihood falls; LOPD shows weaker coupling between the three quantities.

## F 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) [19] 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:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^n)}{\text{std}(\{r_j\}_{j=1}^n) + \epsilon}, \tag{6}$$

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

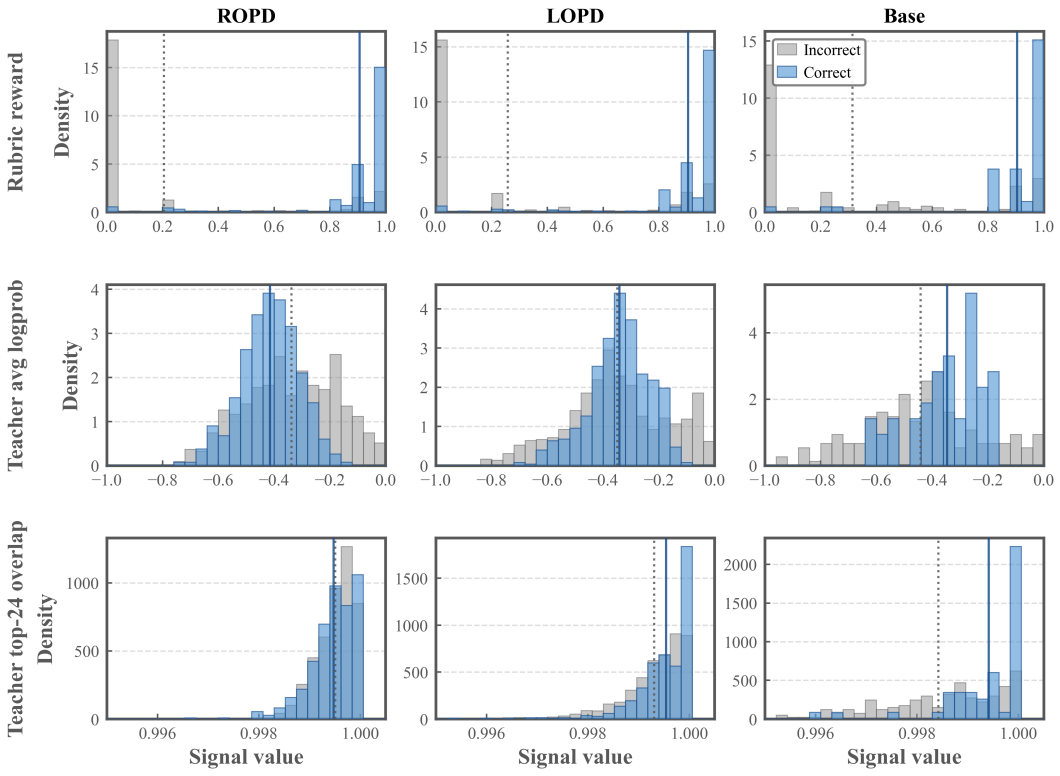


Figure A5: **Signal distributions conditioned on correctness.** Rubric reward (top) strongly separates correct from incorrect responses in all three families. Teacher average log-likelihood (middle) shows weak or reversed separation, particularly for ROPD where correct responses have *lower* teacher likelihood. Teacher top-24 overlap (bottom) distributions are nearly identical for correct and incorrect responses.

position $t$, define the policy ratio

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}. \tag{7}$$

The clipped GRPO objective is

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{y_i\}} \Bigg[ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \Big( \min \Big( \rho_{i,t}(\theta) A_i,\ \text{clip}(\rho_{i,t}(\theta), 1-\eta, 1+\eta) A_i \Big) \\ - \beta D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, y_{i,<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid x, y_{i,<t})\big) \Big) \Bigg], \end{aligned} \tag{8}$$

where $\eta$ is the clipping coefficient, $\pi_{\text{ref}}$ is a fixed reference policy, and $\beta$ 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.

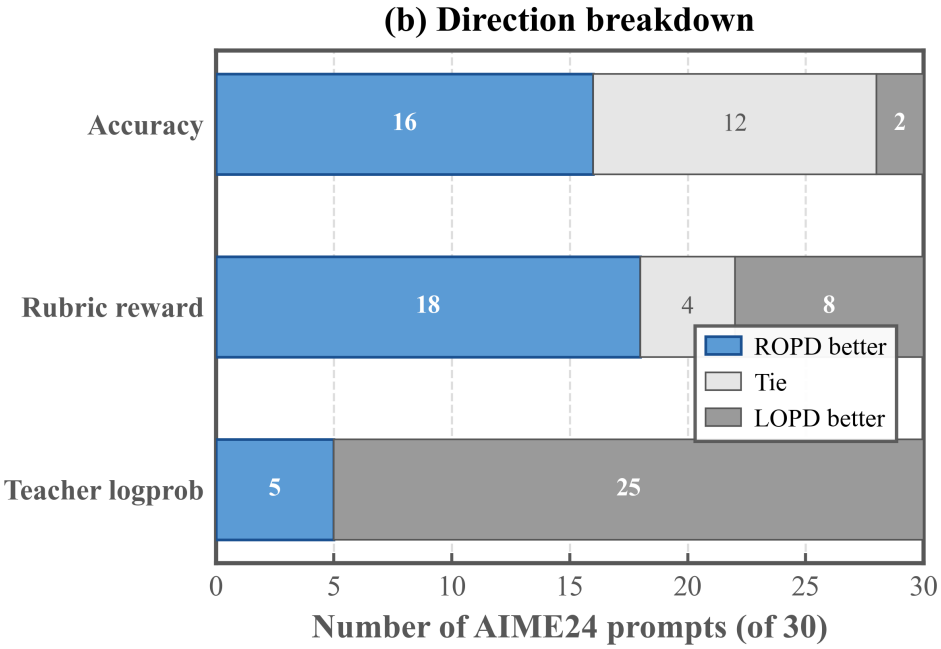


Figure A6: **Final-checkpoint paired comparison (Black step 250 vs. White step 543).** Per-prompt deltas with bootstrap 95% confidence intervals. ROPD final is more accurate (+0.179, CI excludes zero), achieves higher rubric reward (+0.135, CI excludes zero), yet has *lower* teacher log-likelihood (−0.089, CI excludes zero). Prompts are AIME24 (30 prompts).

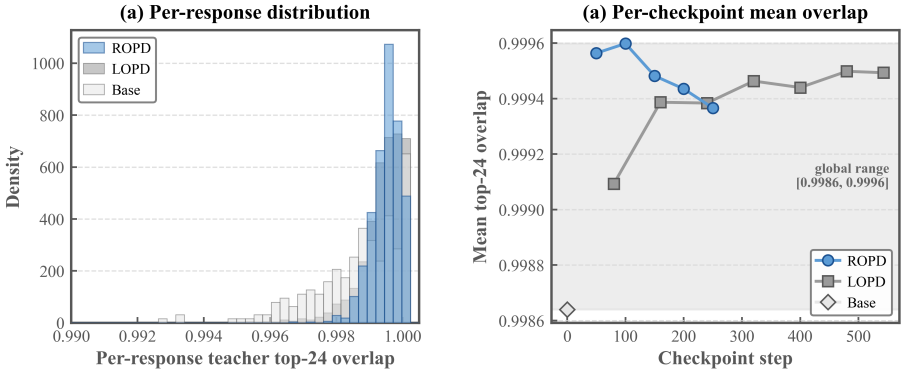


Figure A7: **Teacher top-24 overlap saturation.** Across all checkpoints and families, mean top-24 overlap lies between 0.9986 and 0.9996, leaving negligible within-group dynamic range for advantage computation. This saturation explains why top-24 overlap AUC is near 0.5 for ROPD despite being a white-box signal.

**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 $\pi_\theta$ (initialized from $\pi_{\text{ref}}$)
**Hyperparameters:** teacher answers $m$, student rollouts $n$, rubric criteria count $K$, clip range $\epsilon_{\text{clip}}$, learning rate $\eta$, training steps $N$
**Output:** Trained student policy $\pi_\theta$
**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** $\pi_\theta$