EXPECTED REASONING-STEP RETURN UNIFIES ON-POLICY LEARNING FROM REWARDS AND TEACHERS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy reasoning models can learn from task rewards or teacher signals, but these sources differ in form and can favor conflicting updates, leaving unclear which should guide a given reasoning action. We introduce **Expected Reasoning-Step Return (ERSR)**, which treats semantic reasoning steps as macro-actions and uses Monte Carlo student-policy rollouts to estimate the expected final task reward of student-generated and teacher-proposed actions in a common return space for step-level comparison. ERSR analysis reveals an outcome-dependent asymmetry: student actions are more beneficial than teacher replacements on successful trajectories, whereas teacher replacements become more beneficial on failed trajectories. We further show that student answer-probe gains track student-step ERSR utility and distinguish beneficial from harmful reasoning steps. Based on these findings, we propose **Return-Referenced On-Policy Learning (R^2OPL)**, which reinforces student reasoning on successful trajectories and distills teacher signals on failed ones, while using group success rate for difficulty scaling and student-probe gains for step-level modulation. Experiments across reasoning benchmarks and teacher–student configurations show that R^2OPL consistently outperforms strong baselines. ERSR training dynamics further show that R^2OPL jointly exploits substantial utility from both reward- and teacher-side signals, whereas existing hybrids often leave substantial residual utility in one branch.

1 INTRODUCTION

Recent advances in LLM reasoning increasingly rely on on-policy post-training, where models learn from trajectories sampled by their current policy. Reward-driven methods such as GRPO(Shao et al., 2024), DAPO(Yu et al., 2026a), and GSPO(Zheng et al., 2025) reinforce behaviors associated with successful solutions, while on-policy distillation (OPD) provides dense teacher supervision on student-visited states(Agarwal et al., 2024; Li et al., 2026c; Ko et al., 2026). Yet these signals need not align: teacher supervision may undervalue correct student reasoning, favor teacher-preferred over student-beneficial actions, or become unreliable under teacher–student mismatch and noisy supervision(Akhondzadeh et al., 2026; Yang et al., 2026a; Li et al., 2026c; Ko et al., 2026; Ding & Zhang, 2026). This raises a fundamental question: when reward- and teacher-derived signals disagree, which should guide the optimization of a reasoning action?

Despite their different objectives, reward-based reinforcement and teacher-guided distillation ultimately seek to improve the student's expected task reward. Reward-based methods optimize this objective directly, whereas OPD uses teacher–student discrepancy as a surrogate for beneficial learning. Yet disagreement is only a coarse indicator of learning utility(Wang et al., 2026): even stronger teachers can provide limited or negative transfer when their signals are incompatible with the student(Wu et al., 2026b; Lu & Liu, 2026), and teacher likelihood may diverge from task-level correctness(Zhang et al., 2026c). Thus, what matters is not how strongly the teacher disagrees with the student, but whether the favored reasoning action increases the student's expected task reward.

To directly measure this utility, we introduce **Expected Reasoning-Step Return (ERSR)**, which treats each semantic reasoning step as a macro-action and evaluates its contribution to the expected final task reward. As illustrated in Figure 1, from the same prefix state, we consider three continuations: the student re-decides from the original prefix, continues from its original reasoning step, or continues after replacing that step with a teacher-provided step. All subsequent rollouts are sam-

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

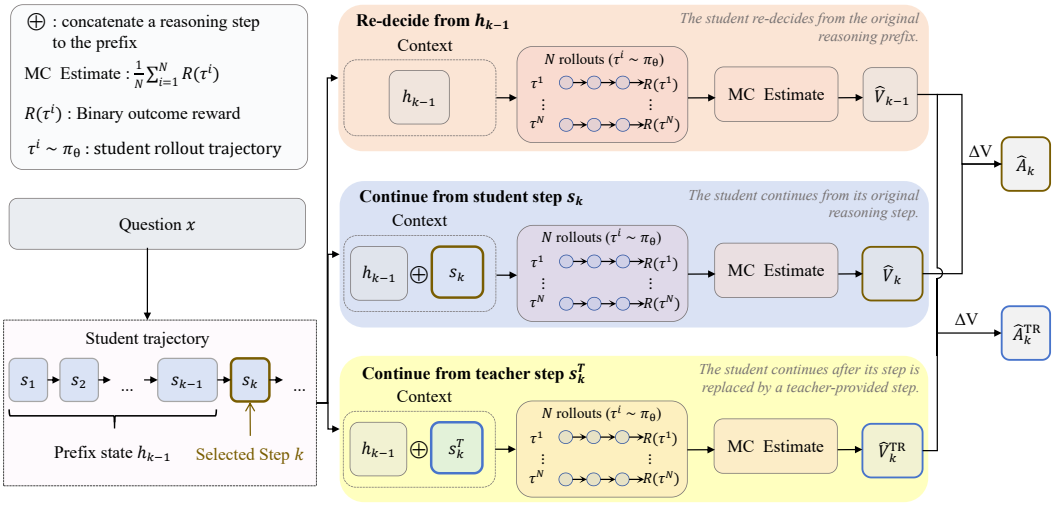


Figure 1: ERSR evaluates a reasoning step by estimating its student-step advantage $\widehat{A}_k$ and teacher-replacement advantage $\widehat{A}_k^{\mathrm{TR}}$, with all subsequent continuations sampled from the student policy.

pled from the student policy, and Monte Carlo estimation yields the expected final reward $\widehat{V}$ of each continuation. Their differences define the *student-step advantage* $\widehat{A}_k$ and *teacher-replacement advantage* $\widehat{A}_k^{\mathrm{TR}}$, placing student and teacher actions in the same expected-return space.

Offline ERSR analysis reveals a clear outcome-dependent asymmetry: on successful trajectories, student reasoning steps are more beneficial than teacher replacements, whereas on failed trajectories, teacher replacements become substantially more beneficial than the student's original actions. We further find that student answer-probe gains consistently track student-step ERSR utility across both outcomes, while signals computed on observed student steps do not reliably reveal teacher-replacement utility. Motivated by these findings, we propose **Return-Referenced On-Policy Learning (R^{2}OPL)**, which reinforces student reasoning on successful trajectories and distills teacher signals on failed ones, while using group success rate for problem-level difficulty scaling and student-probe gains for fine-grained reasoning-step modulation.

Experiments across eight benchmarks spanning mathematical, scientific, and logical reasoning show that R^{2}OPL consistently outperforms strong reward-based, distillation, and hybrid baselines. ERSR training dynamics show that R^{2}OPL jointly exploits both reward- and teacher-side utility, whereas hybrid methods often leave substantial residual utility in one branch. Our contributions are threefold: (1) we introduce ERSR, which compares and tracks student-step and teacher-replacement utility in a common expected-return space; (2) using ERSR, we reveal an outcome-dependent asymmetry between the two and show that student answer probes track student-step utility; and (3) we translate these findings into R^{2}OPL, which unifies on-policy learning from rewards and teacher signals through outcome-conditioned routing, difficulty scaling, and step-level modulation.

2 Understanding Reward and Teacher Signals with ERSR

2.1 Preliminaries

On-policy learning. We use *on-policy learning (OPL)* as a broad term for methods that learn from trajectories sampled from the current student policy, including both reward-based reinforcement learning and on-policy distillation. Let $x \sim \mathcal{D}$ denote a reasoning problem and $\tau = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$ a trajectory sampled from the student policy π_θ. Each trajectory receives a binary outcome reward $R(\tau) \in \{0, 1\}$ indicating whether its final answer is correct. The ultimate objective of OPL is to maximize the student's expected task reward,

$$J_{\mathrm{OPL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|x)}\left[R(\tau)\right]. \tag{1}$$

Reward-based reinforcement learning and on-policy distillation both seek to improve J_{OPL} in Eq. 1, but construct their learning signals differently. For a unified view, we express their token-level learning signals through a common policy-gradient-style surrogate. Let $h_t = (x, y_{<t})$ denote the current context, $\widehat{A}_t$ the learning signal assigned to token y_t, and $\mathrm{sg}[\cdot]$ the stop-gradient operator:

$$J_{\mathrm{PG}}(\theta; \widehat{A}) = \mathbb{E}_{x\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} \mathrm{sg}\Big[\widehat{A}_t\Big] \log \pi_\theta(y_t \mid h_t) \right]. \tag{2}$$

Reinforcement learning. Reward-based RL constructs learning signals directly from task outcomes. A representative example is GRPO(Shao et al., 2024), which samples a group of trajectories $\{\tau_i\}_{i=1}^{G}$ for the same prompt and normalizes their outcome rewards to obtain

$$\widehat{A}_i^{\mathrm{GRPO}} = \frac{R_i - \overline{R}}{\sigma_R + \epsilon}, \qquad \overline{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad \sigma_R = \sqrt{\frac{1}{G}\sum_{j=1}^{G} \big(R_j - \overline{R}\big)^2}. \tag{3}$$

The resulting trajectory-level advantage is shared across all tokens in τ_i. While this provides a direct signal from task rewards, its credit assignment remains inherently coarse, as different reasoning steps may contribute differently to the final reward.

On-policy distillation. OPD learns from a teacher policy π_T on states visited by π_θ. A standard setting in OPD assumes that the teacher achieves a higher expected task reward than the student,

$$\mathbb{E}_{x\sim\mathcal{D},\, \tau\sim\pi_T(\cdot|x)}[R(\tau)] > \mathbb{E}_{x\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|x)}[R(\tau)]. \tag{4}$$

Under this premise, OPD uses the teacher–student log-probability difference as the surrogate learning signal in Eq. 2 for dense token-level supervision on student-visited states,

$$\widehat{A}_t^{\mathrm{OPD}} = \log \pi_T(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t). \tag{5}$$

This signal assigns larger values to tokens favored more by the teacher than by the student. However, the teacher's higher expected reward does not guarantee that its local preferences improve the student's expected task reward; teacher–student disagreement may instead reflect preferences unrelated to task utility(Yang et al., 2026a; Ding & Zhang, 2026). Consequently, $\widehat{A}_t^{\mathrm{OPD}}$ remains a surrogate learning signal, and a larger distillation signal need not imply greater expected-return utility.

2.2 Expected Reasoning-Step Return

Reasoning steps as actions. To compare reward- and teacher-driven learning in a common task space, we evaluate their favored reasoning actions by their effect on the student's expected final reward. Token-level returns are formally well defined, but individual tokens rarely represent complete reasoning operations, teacher–student differences may reflect lexical variation, and token-level effects are often too small for efficient Monte Carlo (MC) estimation. A detailed analysis of action granularity for ERSR is provided in Appendix A.2. We therefore treat each semantic reasoning step as a macro-action, enabling direct comparison between the student's original step and a teacher-provided alternative in the same expected-return space. Steps are identified from decoded responses using structural and semantic boundaries and mapped back to the original token sequence while preserving the original tokenization, with details provided in Appendix A.3.

ERSR definition and estimation. Consider a student trajectory segmented into reasoning steps $\tau = (s_1, \ldots, s_K)$, and let $h_{k-1} = (x, s_{<k})$ denote the prefix before step k. As illustrated in Figure 1, we compare three continuations from this common prefix: the student re-decides from h_{k-1}, continues from its original step s_k, or continues after replacing s_k with a teacher-provided step s_k^T. Denoting $h_k = (h_{k-1}, s_k)$ and $h_k^T = (h_{k-1}, s_k^T)$, their true expected final rewards are

$$\begin{aligned} V_{k-1} = V^{\pi_\theta}(h_{k-1}), \qquad V_k = V^{\pi_\theta}(h_k), \qquad V_k^T = V^{\pi_\theta}(h_k^T), \\ A_k = V_k - V_{k-1}, \quad A_k^{\mathrm{TR}} = V_k^T - V_{k-1}. \end{aligned} \tag{6}$$

Here, all subsequent continuations are sampled from the same student policy π_θ. The *student-step advantage* A_k measures the expected benefit of preserving the student's reasoning step relative to

Table 1: Outcome-dependent ERSR utility. Positive, zero, and negative denote the percentages of steps by utility sign; Δ_{sign} is the positive–negative percentage-point gap.

Outcome	ERSR Utility	Mean	Positive (%)	Zero (%)	Negative (%)	Δ_{sign} (%)
$R = 1$	$\widehat{A}$	$+0.032$	**51.6**	15.9	32.6	$+19.0$
	$\widehat{A}^{\text{TR}}$	$+0.007$	44.8	14.0	41.2	$+3.7$
$R = 0$	$\widehat{A}$	-0.021	30.2	26.8	43.0	-12.9
	$\widehat{A}^{\text{TR}}$	$+0.064$	**49.8**	24.2	26.0	$+23.7$

re-decision from the same prefix, while the *teacher-replacement advantage* A_k^{TR} measures that of replacing it with the teacher-provided step. Together, they provide a task-grounded basis for comparing student reinforcement and teacher distillation. Since these values are not directly observable, we estimate them using N independent student-policy continuations,

$$\widehat{V}_N(h) = \frac{1}{N} \sum_{n=1}^{N} R\Big(\tau^{(n)}\Big), \qquad \tau^{(n)} \sim \pi_\theta(\cdot \mid h). \tag{7}$$

Substituting the corresponding MC estimates into Eq. 6 yields $\widehat{A}_k$ and $\widehat{A}_k^{\text{TR}}$. We use MC@$N$ to denote estimation with N student-policy rollouts. ERSR thus places student and teacher reasoning actions in a common expected-return space.

2.3 Empirical Findings with ERSR

We analyze ERSR using Qwen3-1.7B as the student and Qwen3-4B-Instruct-2507(Yang et al., 2025) as the teacher on the mathematical dataset DAPO-17K(Yu et al., 2026a). We evaluate 30,000 reasoning steps and use MC@128 to estimate the student-step advantage $\widehat{A}$ and teacher-replacement advantage $\widehat{A}^{\text{TR}}$. Additional details on data sampling and evaluation are provided in Appendix A.4.1.

Outcome-dependent action utility. Table 1 reveals a clear outcome-dependent asymmetry in step utility across both mean advantage and sign distribution. On successful trajectories, student actions yield a higher mean advantage than teacher replacements, $+0.032$ versus $+0.007$, together with a stronger positive sign balance of $+19.0\%$ versus $+3.7\%$, indicating greater utility for preserving student reasoning. On failed trajectories, the preference reverses: student actions have a negative mean advantage of -0.021, whereas teacher replacements reach $+0.064$, with sign balances of -12.9% and $+23.7\%$, respectively. These results support outcome-conditioned source selection: successful trajectories favor student reinforcement, while failed trajectories benefit more from teacher-guided correction. Further analyses across teachers and problem difficulty are provided in Appendix A.4.2.

Answer probes track student-step utility. Trajectory outcome helps select the learning source, but does not distinguish the utility of individual reasoning steps. Since estimating ERSR at every step requires expensive MC rollouts, we seek a lightweight proxy whose variation tracks step-level ERSR utility. For each reasoning state h_k, we use the fixed answer template c_{probe}: `Therefore, the answer is \boxed{<answer>}.`, with `<answer>` replaced by the reference answer a^*. If a^* tokenizes as $(z_1, \ldots, z_M)$, we define

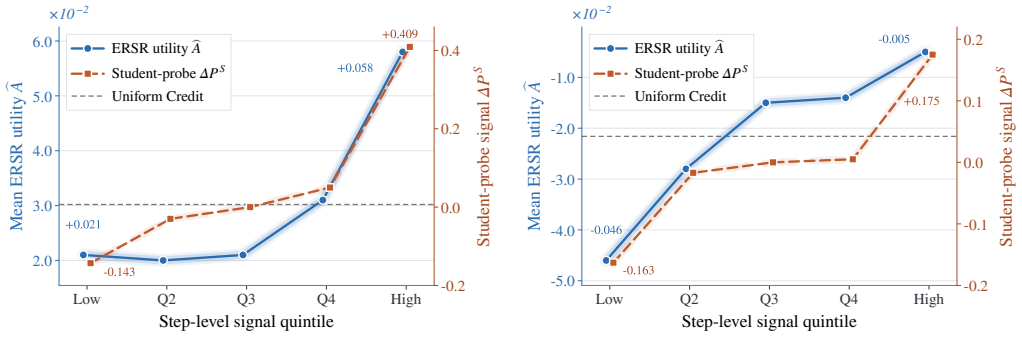
$$P_k^{\text{probe}} = \frac{1}{M} \sum_{m=1}^{M} \pi_{\text{probe}}(z_m \mid h_k, c_{\text{probe}}, z_{<m}), \qquad \Delta P_k = P_k^{\text{probe}} - P_{k-1}^{\text{probe}}, \tag{8}$$

where π_{probe} denotes the probing model. We then evaluate whether ΔP_k tracks the corresponding MC-estimated ERSR utility.

We instantiate π_{probe} with either the student policy π_θ or the teacher policy π_T, yielding student- and teacher-probe gains ΔP_k^S and ΔP_k^T, both evaluated on the same student-generated reasoning steps. For comparison with conventional teacher-side signals, we define the step-mean teacher log probability as $L_k^T = \frac{1}{|s_k|} \sum_{t \in s_k} \log \pi_T(y_t \mid h_t)$ and the step-mean teacher–student discrepancy as

Table 2: **Pearson correlations** between step-level signals and ERSR utility. $\Delta P^{T-S} = \Delta P^{T} - \Delta P^{S}$ and $P^{T-S} = P^{T} - P^{S}$. The highest absolute correlation per utility group is **bolded**; shaded cells mark negative teacher-side correlations on successful trajectories.

	$\widehat{A}$				$\widehat{A}^{\mathrm{TR}}$					
Outcome	ΔP^{S}	ΔP^{T}	$e^{D_{\mathrm{TS}}}$	e^{L_T}	ΔP^{S}	ΔP^{T}	ΔP^{T-S}	P^{T-S}	$e^{D_{\mathrm{TS}}}$	e^{L_T}
$R=1$	**0.246**	0.178	-0.092	-0.104	0.003	0.007	0.005	0.038	$\mathbf{-0.071}$	-0.048
$R=0$	**0.211**	0.131	$+0.109$	$+0.089$	0.008	-0.013	$\mathbf{-0.022}$	0.016	-0.012	-0.002



(a) Student-probe ranking on successful trajectories (b) Student-probe ranking on failed trajectories

Figure 2: **Student-probe ranking of step utility.** (a) Student-step utility across probe quintiles on successful trajectories. (b) Student-step utility across probe quintiles on failed trajectories.

$D_k^{\mathrm{TS}} = \frac{1}{|s_k|}\sum_{t\in s_k}[\log \pi_T(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t)]$, and use their probability-space forms $e^{L_k^T}$ and $e^{D_k^{\mathrm{TS}}}$ in the correlation analysis. Table 2, together with Figure 2, reveals three findings:

- **Student probes best track student-step utility.** The student-probe gain ΔP^S exhibits the strongest correlation with $\widehat{A}$ under both outcomes, reaching 0.246 on successful trajectories and 0.211 on failed trajectories, compared with 0.178 and 0.131 for the teacher probe. Figure 2 further shows that ΔP^S distinguishes more beneficial steps on successful trajectories and more harmful steps on failed trajectories, supporting its use for fine-grained modulation within the outcome-selected learning branch.
- **ERSR exposes a mismatch between teacher preference and task utility.** Among teacher-side evaluations of the student action, ΔP^T correlates most strongly with $\widehat{A}$, while $e^{D_{\mathrm{TS}}}$ and e^{L_T} are substantially weaker. Notably, both become negatively correlated with $\widehat{A}$ on successful trajectories, at -0.092 and -0.104, respectively, despite being weakly positive on failed trajectories. From the ERSR perspective, this suggests that teacher–student discrepancy and teacher likelihood may primarily reflect teacher-specific preferences rather than reward-aligned reasoning utility, consistent with prior observations that teacher preference need not align with reward(Yang et al., 2026a; Ding & Zhang, 2026).
- **Observed student-step signals do not reveal teacher-replacement utility.** In contrast to $\widehat{A}$, none of the examined probe, likelihood, or discrepancy signals shows meaningful correlation with $\widehat{A}^{\mathrm{TR}}$, with all absolute correlations below 0.08. Thus, signals computed from the observed student step do not reliably recover the utility of a teacher replacement.

3 Method

3.1 Return-Referenced On-Policy Learning

Building on the ERSR findings, we propose **Return-Referenced On-Policy Learning (R^{2}OPL)**, a lightweight on-policy method that unifies learning from task rewards and teacher signals within a single objective while retaining their distinct roles. For each prompt $x \sim \mathcal{D}$, we sample a group

of G responses $\{\tau_i\}_{i=1}^{G}$ from the student policy π_θ, where $\tau_i = (y_{i,1}, \ldots, y_{i,T_i})$ and each response receives a binary reward $R_i \in \{0, 1\}$. Let d_x denote the prompt-level difficulty weight and $\widehat{A}_{i,t}^{\mathrm{R^2OPL}}$ the branch-specific learning signal for token $y_{i,t}$, both defined below. Following the policy-gradient formulation in Eq. 2, R^{2}OPL maximizes

$$J_{\mathrm{R^2OPL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{\tau_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|x)}\left[\frac{d_x}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\mathrm{sg}\left[\widehat{A}_{i,t}^{\mathrm{R^2OPL}}\right]\log\pi_\theta(y_{i,t}\mid h_{i,t})\right]. \tag{9}$$

Outcome-conditioned routing and difficulty scaling. Following the outcome-dependent utility revealed by ERSR, R^{2}OPL routes successful trajectories to reward-based reinforcement and failed trajectories to teacher-guided OPD. We further use the group success rate $\overline{R}_x$ to adapt the overall update strength according to prompt difficulty:

$$\widehat{A}_{i,t}^{\mathrm{R^2OPL}} = \begin{cases}\mu\,\widehat{A}_{i,t}^{\mathrm{RL}}, & R_i = 1,\\ \lambda\,\widehat{A}_{i,t}^{\mathrm{OPD}}, & R_i = 0,\end{cases} \qquad \overline{R}_x = \frac{1}{G}\sum_{i=1}^{G}R_i, \qquad d_x = 1 - \overline{R}_x. \tag{10}$$

Here, μ and λ control the relative optimization scales of the reinforcement and distillation branches. The difficulty weight d_x assigns stronger updates to prompts the current student rarely solves and progressively attenuates both branches as the group success rate increases.

Step-level modulation with student probes. R^{2}OPL is optimized at the token level, whereas answer-probe gains are defined over semantic reasoning steps. We segment each response τ_i into steps $(s_{i,1}, \ldots, s_{i,K_i})$ using the ERSR procedure and let $k(t)$ denote the step containing token t, so all tokens within a step share the same modulation factor. Following the ERSR analysis, we use the student-probe gain $\Delta P_{i,k}^{S}$ to characterize the utility of the student's own reasoning step.

On successful trajectories, Figure 2(a) shows that student-step utility remains positive across probe quintiles and rises with ΔP^S, so larger probe gains identify more beneficial steps. We therefore use the binary reward as the RL advantage and reinforce steps with larger probe gains:

$$\widehat{A}_{i,t}^{\mathrm{RL}} = R_i\left(1 + \alpha_R\Delta P_{i,k(t)}^{S}\right), \qquad R_i = 1, \tag{11}$$

where α_R controls the modulation strength. The binary reward provides the base credit, while the probe modulates its magnitude according to student-step utility.

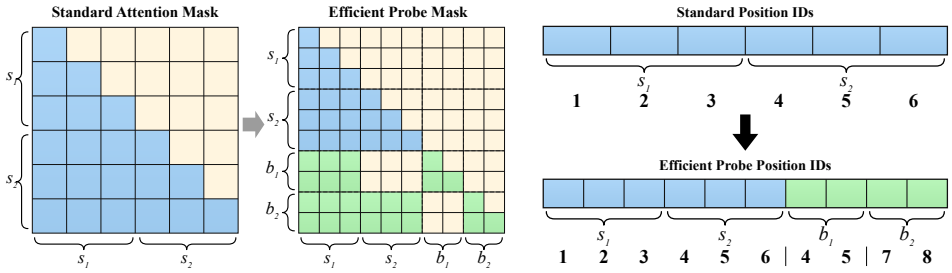
On failed trajectories, Figure 2(b) shows that student-step utility remains negative across probe quintiles but becomes progressively less negative as ΔP^S increases, indicating that lower probe gains identify more detrimental reasoning steps. We therefore retain the token-level OPD signal and assign stronger teacher supervision to steps with lower probe gains:

$$\widehat{A}_{i,t}^{\mathrm{OPD}} = [\log\pi_T(y_{i,t}\mid h_{i,t}) - \log\pi_\theta(y_{i,t}\mid h_{i,t})]\left(1 - \alpha_D\Delta P_{i,k(t)}^{S}\right), \qquad R_i = 0, \tag{12}$$

where α_D controls the modulation strength. Lower student-probe gains induce stronger distillation for more harmful steps, while higher probe gains yield weaker teacher correction.

3.2 Efficient Answer-Probe Computation

Naively evaluating the answer probe at every reasoning state would require repeatedly forwarding different prefixes. Inspired by the packed probing strategy of IGPO (Wang et al., 2025), we compute all student-probe scores within a single forward pass. For a sampled trajectory $\tau = (s_1, \ldots, s_K)$, we pack the required probe blocks after the response as $\widetilde{\tau} = (s_1, \ldots, s_K, b_0, \ldots, b_{K-1})$. Here, $b_0 = (c_{\mathrm{probe}}, a^*)$ probes the initial state before reasoning, while b_k denotes the probe block after step s_k; we omit b_K because the final step typically contains the answer and provides little additional probing value. As illustrated in Figure 3, a customized attention mask allows each b_k to attend only to the prompt, $s_{\le k}$, and its own preceding probe tokens, while its position IDs are remapped to follow the corresponding reasoning prefix rather than its packed position. This preserves the intended prefix-conditioned probe semantics despite packing all probes into a single sequence. Student-probe scores are therefore obtained directly from the student forward pass used for optimization. Probe tokens are excluded from the training loss, so probing requires no additional model forward passes and only adds a small number of processed tokens. Appendix A.5 provides the runtime analysis, showing that R^{2}OPL adds only 2%–8% training overhead over OPD across the three training settings.



(a) Efficient Probe Attention Mask (b) Probe-Aligned Position IDs

Figure 3: **Efficient answer-probe computation.** (a) Packed probes attend only to their reasoning prefixes. (b) Probe position IDs align with the corresponding reasoning steps.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We train on a mixed dataset spanning mathematical, scientific, and logical reasoning. Specifically, we combine DAPO-17K(Yu et al., 2026a) for mathematics, the MCQ subset of Nemotron-Science-v1 for science, and the training split of LogiQA2.0(Liu et al., 2023) for logic, resulting in approximately 32K training examples. For evaluation, we use eight benchmarks across these three domains. Mathematical reasoning is evaluated on AMC23, AIME24, AIME25, and MATH-500(Lightman et al., 2024); scientific reasoning on GPQA-Diamond(Rein et al., 2023) and SciBench(Wang et al., 2023); and logical reasoning on LogicBench(Parmar et al., 2024) and LogiQA2.0(Liu et al., 2023). Detailed dataset information is provided in Appendix A.6.

Models and Baselines. We evaluate R^{2}OPL under three teacher–student configurations: Qwen3-4B-Instruct-2507 $\rightarrow$ Qwen3-1.7B, Qwen3-4B-Instruct-2507 $\rightarrow$ Qwen3-4B(Yang et al., 2025), and Gemma-4-26B-A4B $\rightarrow$ Gemma-4-E2B-it(Gemma Team, 2026). We compare against three baseline categories under a common training and evaluation protocol. *Reward-only* methods include GRPO(Guo et al., 2025), DAPO(Yu et al., 2026a), and GSPO(Zheng et al., 2025); *distillation-only* methods include OPD(Agarwal et al., 2024), ExOPD(Yang et al., 2026d), and EOPD(Jin et al., 2026); and *reward–distillation hybrids* include SRPO(Li et al., 2026b), RLSD(Yang et al., 2026b), and OPDVR(Lin et al., 2026). Since the teachers substantially outperform the corresponding students in our setting, we replace self-distillation in SRPO and RLSD with teacher distillation to construct stronger baselines. We report Avg@16, defined as the mean accuracy over 16 independently sampled responses per problem. Detailed descriptions of baselines are provided in Appendix A.7.

Implementation Details. We implement all methods using `verl` (Sheng et al., 2025) throughout all experiments. All experiments are conducted on the same eight NVIDIA H20 GPUs throughout. Distillation-only methods are trained for 100 steps, while reward-only and reward–distillation hybrid methods are trained for 500 steps, all with a learning rate of 1×10^{-6}. During training, we use a maximum response length of 8,192, temperature 1.0, and top-p 1.0; during evaluation, these are set to 16,384, 0.6, and 0.95, respectively. For R^{2}OPL, we set the branch weights to $\mu = 10$ and $\lambda = 0.1$, the probe modulation strengths to $\alpha_R = 0.25$ and $\alpha_D = 0.5$. Each training batch contains 64 questions with four trajectories sampled per question, yielding 256 trajectories per optimization step. Additional implementation details and hyperparameters are provided in Appendix A.8.

4.2 MAIN RESULTS

Table 3 summarizes the main results across teacher–student configurations. R^{2}OPL achieves the strongest overall performance under both Qwen and Gemma settings, reaching Avg@16 scores of 59.4 and 63.1, respectively, and outperforming the strongest baselines by 3.6 and 2.6 points. The gains extend across mathematical, scientific, and logical reasoning: R^{2}OPL obtains the best or tied-best result on seven of eight benchmarks in both configurations. Notably, these improvements hold against reward-only, distillation-only, and hybrid baselines rather than arising from a single com-

Table 3: **Results across teacher–student configurations.** The best and second-best results in each column, excluding the teacher, are **bolded** and underlined, respectively.

Method	Math				Science		Logic		Avg.
	AMC23	AIME24	AIME25	MATH-500	GPQA-D	SciBench	LogicBench	LogiQA2.0	
Qwen3-4B-Instruct-2507 → Qwen3-1.7B									
Reference Models									
Student	44.4	13.8	9.6	75.4	31.0	37.5	79.1	59.8	43.8
Teacher	92.8	59.2	44.6	96.6	58.5	63.2	85.2	80.1	72.5
Reward Only									
GRPO	64.2	25.6	16.7	82.9	33.6	39.8	79.7	61.4	50.5
DAPO	67.2	31.7	19.4	83.6	35.1	39.9	80.2	66.4	52.9
GSPO	60.2	23.5	14.8	80.8	33.2	37.9	78.6	60.3	48.7
Distillation Only									
OPD	66.7	32.9	21.0	86.9	34.6	42.6	82.0	65.0	54.0
ExOPD	68.9	35.2	24.6	88.1	34.8	42.9	83.5	67.8	55.7
EOPD	69.7	33.8	22.7	88.4	35.6	44.2	**84.0**	68.1	55.8
Reward–Distillation Hybrid									
SRPO	69.2	35.8	25.2	87.1	35.8	43.7	82.7	66.6	55.8
RLSD	65.8	27.3	18.5	84.8	35.2	41.8	80.4	62.3	52.0
OPDVR	67.3	33.5	21.5	87.3	34.3	42.6	81.8	66.3	54.3
R^{2}OPL	**78.1**	**40.6**	**28.5**	**90.6**	**38.7**	**45.2**	83.7	**70.1**	**59.4**
Gemma-4-26B-A4B-it → Gemma-4-E2B-it									
Reference Models									
Student	67.5	36.0	24.0	81.3	38.5	48.9	76.6	54.9	53.5
Teacher	99.1	86.3	75.2	98.8	72.9	72.3	88.6	81.3	84.3
Reward Only									
GRPO	81.4	41.7	26.5	90.4	41.3	49.5	78.1	57.8	58.3
DAPO	83.4	43.5	29.8	91.1	**44.5**	52.8	79.2	59.4	60.5
GSPO	82.0	40.8	27.7	92.3	41.3	48.8	77.9	57.7	58.6
Distillation Only									
OPD	80.0	41.0	28.5	90.4	40.1	49.7	79.0	58.7	58.4
ExOPD	83.1	42.5	29.8	91.1	39.7	48.5	79.0	59.4	59.1
EOPD	82.7	41.9	30.2	92.7	40.6	49.2	79.8	60.7	59.7
Reward–Distillation Hybrid									
SRPO	83.6	41.7	31.5	91.3	41.8	51.6	**81.4**	59.4	60.3
RLSD	81.9	42.9	25.8	92.6	41.7	49.8	78.9	59.8	59.2
OPDVR	80.5	40.8	26.9	90.2	39.4	48.6	79.3	58.0	58.0
R^{2}OPL	**87.7**	**47.5**	**32.9**	**93.3**	**44.5**	**54.1**	81.2	**63.8**	**63.1**

parison category. This consistency across distinct model families, baseline types, and reasoning domains indicates that the improvements are not specific to a particular student model or task type.

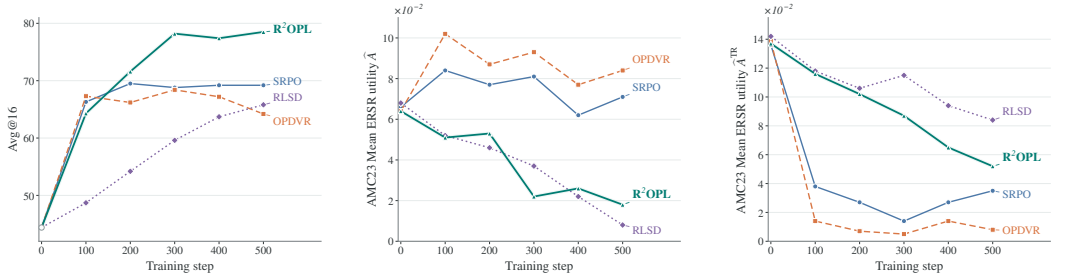
The additional Qwen3-4B-Instruct-2507 → Qwen3-4B results in Appendix A.9 show the same trend. R^{2}OPL reaches an Avg@16 of 73.1, exceeding the strongest baseline by 5.1 points while achieving the best performance on all eight benchmarks. Overall, these results demonstrate the strong and consistent performance of R^{2}OPL across teacher–student configurations.

4.3 Analysis and Ablations

Ablation Study. Table 4 evaluates each component of R^{2}OPL, with all variants trained under identical data, optimization, and evaluation settings. The full method achieves the best performance. Removing either failed-trajectory OPD or successful-trajectory RL causes substantial degradation, confirming their complementary roles under different trajectory outcomes. Removing difficulty scaling also reduces performance; without attenuating updates on easier prompts, overly strong positive reinforcement leads to entropy collapse, making this variant even weaker than using failed-trajectory OPD alone. Although removing answer-probe modulation yields the smallest performance drop, its

Table 4: **Ablation study of R^{2}OPL under the Qwen3-4B-Instruct-2507 → Qwen3-1.7B configuration.** The best result in each column is **bolded**.

Variant	Math				Science		Logic		Avg.
	AMC23	AIME24	AIME25	MATH-500	GPQA-D	SciBench	LogicBench	LogiQA2.0	
Full R^{2}OPL	**78.1**	**40.6**	**28.5**	**90.6**	**38.7**	45.2	83.7	**70.1**	**59.4**
w/o failed-trajectory OPD	63.9	23.8	16.5	82.9	31.1	38.7	79.9	63.4	50.0
w/o successful-trajectory RL	69.2	35.2	23.5	87.4	34.9	42.9	82.8	65.5	55.2
w/o difficulty scaling	67.8	32.9	21.3	84.8	34.1	39.7	80.6	66.0	53.4
w/o answer-probe modulation	77.5	40.2	26.9	90.1	36.4	**45.7**	**84.3**	69.5	58.8



(a) Avg@16 (b) Student-step utility $\widehat{A}$ (c) Teacher-replacement utility $\widehat{A}^{\mathrm{TR}}$

Figure 4: **ERSR dynamics during training.** AMC23 Avg@16 and the remaining ERSR utilities of student reinforcement and teacher distillation are evaluated across training checkpoints.

best checkpoint occurs at step 500, whereas full R^{2}OPL peaks at step 250, indicating that student-probe modulation accelerates training while further improving performance.

ERSR Dynamics. Appendix A.4.3 shows that MC@2 over only 500 sampled steps closely approximates aggregate ERSR statistics, enabling efficient tracking during training. We use $\mathbb{E}[\widehat{A} \mid R = 1]$ and $\mathbb{E}[\widehat{A}^{\mathrm{TR}} \mid R = 0]$ to measure the remaining utilities of student reinforcement and teacher distillation. Their reductions during training reflect the progressive internalization of high-value actions into the policy. Figure 4 reveals distinct utility-consumption patterns across methods. SRPO and OPDVR rapidly improve performance while primarily exhausting teacher-distillation utility: by step 100, $\widehat{A}^{\mathrm{TR}}$ drops from 0.136 to 0.038 for SRPO and from 0.139 to 0.014 for OPDVR, while $\widehat{A}$ remains largely unconsumed. RLSD exhibits the opposite tendency, progressively reducing $\widehat{A}$ from 0.068 to 0.008, while retaining substantially more teacher-distillation utility. In contrast, R^{2}OPL reduces both $\widehat{A}$ from 0.064 to 0.018 and $\widehat{A}^{\mathrm{TR}}$ from 0.137 to 0.052, while achieving the highest AMC23 accuracy of 78.1. Notably, these asymmetric utility trajectories persist even though all methods improve substantially over the initial policy. These dynamics indicate that existing hybrid methods predominantly internalize one source of expected-return utility, whereas R^{2}OPL jointly realizes both. Appendix A.10 further analyzes the RL–OPD strength-control parameters μ and λ and their effects on balancing the learning sources, measured through ERSR utility realization.

5 Conclusion

We introduced **Expected Reasoning-Step Return (ERSR)**, a task-grounded framework that compares and tracks student-step and teacher-replacement utility in a common expected-return space. ERSR reveals an outcome-dependent asymmetry: successful trajectories favor student reasoning, whereas failed trajectories benefit more from teacher replacement. We further showed that student answer-probe gains track student-step utility and distinguish beneficial from harmful reasoning steps. Building on these findings, we proposed **R^{2}OPL**, which combines outcome-conditioned routing, difficulty scaling, and student-probe modulation to coordinate learning from rewards and teacher signals. Experiments across reasoning domains and teacher–student configurations show strong and consistent gains, while ERSR training dynamics further reveal that R^{2}OPL jointly exploits both reward- and teacher-side utility, unlike existing hybrid methods.

AI USE STATEMENT

We used generative AI tools for limited implementation assistance, including code generation and debugging, and for language editing and polishing of the manuscript. All AI-assisted outputs were manually reviewed, verified, and revised by the authors before use. The authors made all research decisions and take full responsibility for the methodology, experiments, analyses, technical claims, and final content of this work.

ETHICS STATEMENT

This work studies on-policy learning from task rewards and teacher supervision for improving the reasoning capabilities of large language models. Our experiments use available models and datasets covering mathematical, scientific, and logical reasoning, and do not involve human subjects, personally identifiable information, or sensitive personal data. ERSR is an analytical framework for evaluating reasoning-step utility, while R^2OPL is a post-training method and does not introduce application-specific capabilities that pose societal or safety risks beyond those associated with the underlying language models. We follow the licenses and usage terms of the models and datasets used in our experiments, and our research adheres to the ICLR Code of Ethics. We are not aware of ethical concerns or conflicts of interest arising from this work.

REPRODUCIBILITY STATEMENT

We provide the complete formulation of Expected Reasoning-Step Return (ERSR) and Return-Referenced On-Policy Learning (R^2OPL) in the main text, including the Monte Carlo estimation of reasoning-step returns, outcome-conditioned routing, difficulty-aware scaling, and student-probe-based step-level modulation. The experimental section specifies the training datasets, evaluation benchmarks, teacher–student configurations, optimization settings, sampling parameters, and key hyperparameters used in our experiments. The appendix further provides details of ERSR evaluation and sampling, reasoning-step segmentation, answer-probe runtime analysis, compared methods, training configurations, additional teacher–student settings, and extended analyses of ERSR and reward–distillation interactions. Together, these descriptions provide the information necessary to reproduce the main experimental results and analyses reported in this work.

REFERENCES

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

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint arXiv:2607.04037*, 2026.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025a.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025b.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 59637–59659, 2026.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

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

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

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *Advances in Neural Information Processing Systems*, 38:114399–114431, 2026.

Yaru Hao, Li Dong, Xun Wu, Shaohan Huang, Zewen Chi, and Furu Wei. On-policy rl with optimal reward baseline. *arXiv preprint arXiv:2505.23585*, 2025.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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

Jian Hu et al. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 1(4), 2025.

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

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1444–1466, 2024.

Yuxuan Jiang and Francis Ferraro. Bridging reasoning trajectories in on-policy distillation via near-future guidance. *arXiv preprint arXiv:2606.00305*, 2026.

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

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.

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.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Boyan Li, Bingsen Chen, Chenghao Yang, Ping Nie, Chen Zhao, and Xi Ye. Sequential beats joint: On the interplay between on-policy distillation and rlvr. *arXiv preprint arXiv:2609.04108*, 2026a.

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

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

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision. *arXiv preprint arXiv:2608.06296*, 2026d.

Yu Li, Shu Hong, and Tian Lan. Localizing credit at the divergence: Path-conditioned self-distillation for llm reasoning. *arXiv preprint arXiv:2606.15576*, 2026e.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Wenze Lin, Jiale Zhao, Xitai Jiang, Songde Rao, Yining Li, Shenzhi Wang, Bingxiang He, and Gao Huang. On-policy distillation with verifiable reward. *arXiv preprint arXiv:2608.24696*, 2026.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023.

Xinyu Liu, Kechen Jiao, Chunyang Xiao, Runsong Zhao, Junhao Ruan, Bei Li, Jiahao Liu, Qifan Wang, Xin Chen, Jingang Wang, et al. Teacher-guided policy optimization for on-policy reasoning distillation under large policy divergence. *arXiv preprint arXiv:2605.13230*, 2026a.

Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18041–18059, 2026b.

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

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, et al. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *arXiv preprint arXiv:2508.08221*, 2025b.

Taiming Lu and Zhuang Liu. Strong teacher not needed? on distillation in llm pretraining. *arXiv preprint arXiv:2605.23857*, 2026.

Andrei Liviu Nicolicioiu, Mohammad Pezeshki, and Aaron Courville. On-policy self-distillation with sampled demonstrations reduces output diversity. *arXiv preprint arXiv:2606.26091*, 2026.

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

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. Rlcsd: Reinforcement learning with contrastive on-policy self-distillation. *arXiv preprint arXiv:2606.11709*, 2026.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13679–13707, 2024.

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.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

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

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

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *arXiv e-prints*, pp. arXiv–2510, 2025.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026.

Sitong Wu, Haoru Tan, Xichen Zhang, Bin Xia, Wenhu Zhang, Xiaojuan Qi, Bei Yu, and Jiaya Jia. Trac: Teacher-guided token reward with adaptive calibration for robust policy optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47869–47884, 2026a.

Yecheng Wu, Song Han, and Han Cai. Lightning opd 2.0: Mitigating style bias in cross-teacher on-policy distillation for large reasoning models. *arXiv preprint arXiv:2607.28449*, 2026b.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Jian Xiong, Jingbo Zhou, Jingyong Ye, Qiang Huang, and Dejing Dou. Aapo: Enhancing the reasoning capabilities of llms with advantage margin. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24663–24680, 2026.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Haolei Xu, Xiaowen Xu, Haiwen Hong, Zixuan Ni, Hongxing Li, Yiwen Qiu, Weiming Lu, and Yongliang Shen. Pass the baton: Trajectory-relayed on-policy distillation. *arXiv preprint arXiv:2607.26057*, 2026a.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Beyond grpo and on-policy distillation: An empirical sparse-to-dense reward principle for language-model post-training. *arXiv preprint arXiv:2605.12483*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chen Yang, Haiyuan Wan, Rengrong Xiong, Yize Chen, and Danny HK Tsang. Beyond imitation: Filtering on-policy distillation by reasoning progress. *arXiv preprint arXiv:2608.19408*, 2026a.

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

Matthew Yang, Hao Bai, Ian Wu, Gene Yang, Amrith Setlur, and Aviral Kumar. Int: Self-proposed interventions enable credit assignment in llm reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 85054–85091, 2026c.

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

Junxi Yin, Haisen Luo, Zhenyu Li, Yihua Liu, Dan Liu, Zequn Li, and Xiaohang Xu. Pinpointing crucial steps: Attribution-based credit assignment for verifiable reinforcement learning. *arXiv preprint arXiv:2510.08899*, 2025.

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

Zichao Yu, Chengzhi Yu, Shengze Xu, Yujin Han, Bingqing Jiang, Xu Wang, and Difan Zou. Mismatch matters: On-policy distillation beyond token agreement. *arXiv preprint arXiv:2608.09836*, 2026b.

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. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, 2026a.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for llm reasoning. *arXiv preprint arXiv:2602.22495*, 2026b.

Zhu Zhang, Jixun Wang, Xiaoang Xu, Xiaorong Wang, Zihan Zhou, Zhiyuan Wang, Shuo Wang, Chaojun Xiao, and Yuezhi Zhou. Beyond teacher likelihood: Group-calibrated on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.19181*, 2026c.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026a.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, et al. Geometric-mean policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 41354–41374, 2026b.

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

Zhanke Zhou, Xiangyu Lu, Chentao Cao, Brando Miranda, Tongliang Liu, Bo Han, and Sanmi Koyejo. The easy, the hard, and the learnable: Confidence and difficulty-adaptive policy optimization for llm reasoning. *arXiv preprint arXiv:2606.07950*, 2026.

A Appendix

A.1 Related Work

On-Policy Reinforcement Learning. On-policy reinforcement learning has become a major paradigm for LLM post-training, with PPO(Schulman et al., 2017) establishing the clipped policy-gradient framework widely adopted in early RLHF systems(Ouyang et al., 2022). To reduce the computational overhead and complexity of actor–critic training, subsequent work revisits simpler critic-free policy gradients, including RLOO(Ahmadian et al., 2024), ReMax(Li et al., 2023), and REINFORCE++(Hu et al., 2025), which improve variance reduction, training stability, and efficiency while retaining on-policy optimization. For reasoning with verifiable rewards, DeepSeekMath(Shao et al., 2024) introduced Group Relative Policy Optimization (GRPO), replacing the learned critic with relative rewards from multiple responses to the same prompt, while DeepSeek-R1(Guo et al., 2025) demonstrated that large-scale reinforcement learning can substantially strengthen reasoning capabilities. Recent work has refined this paradigm along several directions: DAPO(Yu et al., 2026a) improves clipping, dynamic sampling, token-level loss aggregation, and handling of overlong responses, while Lite PPO(Liu et al., 2025b) systematically revisits normalization, clipping, and loss aggregation; Dr. GRPO(Liu et al., 2025a) identifies optimization biases induced by reward and length normalization, whereas Reinforce-Rej(Xiong et al., 2025) highlights the importance of filtering uninformative response groups. OPO(Hao et al., 2025) introduces an optimal reward baseline under exact on-policy training, while GPG(Chu et al., 2026) directly optimizes the original policy-gradient objective without surrogate losses. Other approaches focus on stabilizing policy updates: GSPO(Zheng et al., 2025) moves importance weighting and clipping to the sequence level, GMPO(Zhao et al., 2026b) uses geometric-mean aggregation to suppress outlier updates, CISPO(Chen et al., 2025a) clips importance-sampling weights rather than token updates, and SAPO(Gao et al., 2025) replaces hard clipping with smooth adaptive gating. AAPO(Xiong et al., 2026) refines group-relative advantage estimation through an advantage-margin mechanism, while SEED-GRPO(Chen et al., 2025b) modulates policy updates using semantic uncertainty. More recently, CoDaPO(Zhou et al., 2026) jointly adapts sampling and update weighting according to model confidence and problem difficulty. Despite these advances, reasoning-oriented RL methods still derive supervision primarily from final trajectory rewards, leaving credit assignment across individual reasoning steps comparatively coarse.

On-Policy Distillation. Traditional knowledge distillation transfers teacher knowledge through token- or sequence-level matching on fixed or teacher-generated data(Hinton et al., 2015; Kim & Rush, 2016), while LLM-oriented methods such as MiniLLM(Gu et al., 2024) and DistiLLM(Ko et al., 2024) improve divergence objectives and training efficiency. Generalized Knowledge Distillation (GKD)(Agarwal et al., 2024) moves distillation on-policy by training on student-generated outputs and querying the teacher on student-visited states, reducing train–inference mismatch. Recent work further studies the dynamics, efficiency, and stability of this paradigm. Rethinking OPD(Li et al., 2026c; Fu et al., 2026) analyzes teacher–student alignment and state coverage; Fast Prefix OPD(Zhang et al., 2026a) focuses supervision on informative reasoning prefixes; and On-Policy Delta Distillation(Heo et al., 2026), REOPOLD(Ko et al., 2026), and TrOPD(Xing et al., 2026) redesign token-level teacher signals or constrain updates under policy mismatch. A related branch replaces external teachers with privileged self-supervision: OPSD(Zhao et al., 2026a) conditions the same model on verified solutions, while SDPO(Hübotter et al., 2026), U-OPSD(Li et al., 2026d), and RLCSD(Pan et al., 2026) extend this idea through dense self-distillation, self-consistency, or contrastive privileged signals.

Several methods combine distillation signals with outcome rewards. SRPO(Li et al., 2026b) routes successful rollouts to GRPO and failed ones to self-distillation; RLSD(Yang et al., 2026b) uses the privileged teacher–student gap to modulate RLVR credit; RLAD(Zhang et al., 2026b) selectively imitates reward-compatible teacher guidance; RG-OPD(Akhondzadeh et al., 2026) gates supervision with verifier feedback; and OPDVR(Lin et al., 2026) reshapes the implicit distillation reward by trajectory correctness. TGPO(Liu et al., 2026a) and TRAC(Wu et al., 2026a) similarly integrate token-level teacher guidance with trajectory-level reward, while G-OPD(Yang et al., 2026d) generalizes OPD as dense KL-constrained RL with adjustable reward scaling. Other work explores staged integration through sparse-to-dense reward transfer(Xu et al., 2026b) or sequential OPD–RL training(Li et al., 2026a). Despite these advances, teacher supervision can remain noisy, mismatched,

or overly restrictive(Ding & Zhang, 2026; Yu et al., 2026b; Nicolicioiu et al., 2026; Zhang et al., 2026c). Existing methods therefore mainly select, filter, rescale, or combine reward- and teacher-derived signals within surrogate objectives, rather than directly comparing their utility in a common student-policy expected-return space.

Reasoning-Step Credit Assignment. Beyond trajectory-level rewards and token-level teacher supervision, recent work increasingly treats intermediate reasoning steps as the unit of learning and evaluation. Process supervision provides direct step-level correctness signals(Lightman et al., 2024), while PAVs(Setlur et al., 2025), VinePPO(Kazemnejad et al., 2024), and SPO(Guo et al., 2026) estimate reasoning progress or advantage from downstream success and intermediate continuations. Other methods localize influential or erroneous reasoning steps through resampling, targeted intervention, or attribution, including Self-Explore(Hwang et al., 2024), InT(Yang et al., 2026c), and ACPO(Yin et al., 2025). Parallel efforts refine teacher supervision at a coarser reasoning granularity. Fast Prefix OPD(Zhang et al., 2026a) and P-ALIGN(Liu et al., 2026b) focus distillation on informative reasoning prefixes, while TOPD(Jiang & Ferraro, 2026) and Hindsight Self-Distillation(Li et al., 2026e) localize supervision around meaningful trajectory divergences rather than isolated token discrepancies. Relay-OPD(Xu et al., 2026a) further enables localized teacher intervention when student reasoning begins to deviate. These approaches primarily improve credit assignment for student reasoning or localize teacher guidance, but evaluate them through different learning signals. ERSR instead evaluates the student step and teacher replacement from the same prefix under an identical downstream student policy, placing both in a common expected-final-reward space.

A.2 Action Granularity for ERSR

Token-level returns are formally well defined: after any generated token, one can estimate the expected final task reward under subsequent student-policy rollouts. However, formal validity does not make tokens an efficient or stable unit for reasoning credit assignment. We show that token-level estimation faces two fundamental limitations. First, when the return of a semantic reasoning operation is distributed across multiple tokens, each token carries only a small fraction of the overall effect, making its return substantially harder to resolve with finite Monte Carlo (MC) samples; meanwhile, the number of actions to be evaluated grows directly with sequence length. Second, token-level credit depends on the particular lexical realization and tokenization-induced decomposition of a reasoning operation, whereas the return of the complete semantic operation depends only on its boundary states and is invariant to its internal decomposition.

Monte Carlo estimation error. Let

$$V(h) = \mathbb{E}_{\tau' \sim \pi_S(\cdot|h)} \left[R(\tau') \right] \tag{13}$$

denote the expected final reward obtained by continuing from state h with the student policy π_S. Since $R \in \{0, 1\}$, $V(h)$ is the success probability of student-policy continuations from h.

Consider two post-action states h^a and h^b, with

$$p_a = V(h^a), \qquad p_b = V(h^b), \qquad \Delta = p_a - p_b. \tag{14}$$

Using N independent student-policy rollouts from each state, their MC estimates are

$$\widehat{p}_a = \frac{1}{N} \sum_{n=1}^{N} R_n^a, \qquad \widehat{p}_b = \frac{1}{N} \sum_{n=1}^{N} R_n^b, \tag{15}$$

and

$$\widehat{\Delta} = \widehat{p}_a - \widehat{p}_b. \tag{16}$$

The estimator is unbiased,

$$\mathbb{E}[\widehat{\Delta}] = \Delta, \tag{17}$$

with variance

$$\mathrm{Var}(\widehat{\Delta}) = \frac{p_a(1 - p_a) + p_b(1 - p_b)}{N} \leq \frac{1}{2N}. \tag{18}$$

Thus, changing the action granularity does not improve the worst-case MC convergence rate: for binary final rewards, the estimation error remains of order $O(N^{-1/2})$.

What changes with granularity is the magnitude of the return difference that must be resolved. By Hoeffding's inequality,

$$\Pr\left(|\widehat{\Delta} - \Delta| \geq \epsilon\right) \leq 2\exp\left(-\frac{N\epsilon^2}{2}\right). \tag{19}$$

In particular, a sufficient condition for identifying the sign of a nonzero return difference with error probability at most α is

$$N \geq \frac{2}{\Delta^2}\log\frac{2}{\alpha}. \tag{20}$$

Hence, the MC budget required to distinguish beneficial from harmful actions scales quadratically with the inverse effect size,

$$N = O\left(\frac{1}{\Delta^2}\right). \tag{21}$$

Statistical inefficiency at token granularity. Consider a reasoning step containing m tokens,

$$h_0 \to h_1 \to \cdots \to h_m, \tag{22}$$

where h_j denotes the state after the first j tokens of the step. We define the token-level value increment and the return of the complete reasoning step as

$$\delta_j = V(h_j) - V(h_{j-1}), \qquad A_{\text{step}} = V(h_m) - V(h_0). \tag{23}$$

These quantities satisfy the exact telescoping identity

$$A_{\text{step}} = \sum_{j=1}^{m} \delta_j. \tag{24}$$

Equation 24 does not imply that every token has a smaller effect than the complete step, since credit may concentrate on a few decisive tokens. Token-level inefficiency instead arises in the *diffuse-credit* regime, where the return effect of a reasoning operation is distributed across multiple tokens. In this regime, for a substantial fraction of the tokens,

$$|\delta_j| = \Theta\left(\frac{|A_{\text{step}}|}{m}\right). \tag{25}$$

Combining Eq. 25 with the sign-identification bound in Eq. 20, the sufficient MC budgets for comparable confidence scale as

$$N_{\text{token}} = O\left(\frac{m^2}{A_{\text{step}}^2}\log\frac{1}{\alpha}\right), \qquad N_{\text{step}} = O\left(\frac{1}{A_{\text{step}}^2}\log\frac{1}{\alpha}\right). \tag{26}$$

Thus, under diffuse reasoning credit, resolving a token-level effect incurs a quadratic penalty in the number of tokens composing the step:

$$\frac{N_{\text{token}}}{N_{\text{step}}} = \Theta(m^2). \tag{27}$$

The difference arises not because step-level MC samples are intrinsically less noisy, but because they resolve the aggregate return effect of a complete semantic operation rather than the substantially smaller effects of its constituent lexical actions.

Token granularity incurs an additional cost when evaluating an entire response. Suppose a response contains K reasoning steps with an average of m tokens per step, so that $T = mK$. Step-level evaluation requires K return estimates, whereas token-level evaluation considers approximately m times more actions. Combining this coverage factor with the per-action penalty in Eq. 27, the total MC cost scales under diffuse reasoning credit as

$$\frac{C_{\text{token}}}{C_{\text{step}}} = \Theta(m^3), \tag{28}$$

up to logarithmic factors for simultaneous confidence. This reflects both the smaller return effects that must be resolved at token granularity and the substantially larger number of actions that must be evaluated.

Representation dependence of token-level credit. The limitation of token-level return is not solely statistical. Tokens are representation-dependent units whose identity and boundaries are determined by lexical realization and tokenization rather than the underlying reasoning operation. Even the same surface form may admit substantially different token decompositions. For example, `indivisible` can be decomposed as

$$\begin{aligned} &\texttt{in|div|isible}, \quad \texttt{ind|iv|isible}, \quad \texttt{indi|visible}, \\ &\texttt{in|div|is|ible}, \quad \texttt{indi|vis|ible}. \end{aligned} \tag{29}$$

These decompositions preserve the same lexical content while changing both the number and identity of token-level actions.

Representation also varies under minor surface changes that preserve essentially the same semantic function. For instance, the discourse connective `but` may correspond to distinct token forms such as

$$\texttt{but}, \quad \texttt{␣but}, \quad \texttt{But}, \quad \texttt{␣But}, \quad \texttt{␣BUT}, \tag{30}$$

with further variants incorporating punctuation, such as `.But`, `,but`, `-but`, and `---but`. Thus, token-level actions can change with segmentation, whitespace, capitalization, or punctuation even when their underlying lexical or discourse function remains unchanged.

This creates a representation-fragmentation problem for reasoning credit. Consider a semantic reasoning operation that induces a transition from boundary state h_{in} to h_{out}, with return

$$A_{\text{sem}} = V(h_{\text{out}}) - V(h_{\text{in}}). \tag{31}$$

Now consider two internal decompositions of the same transition,

$$\begin{aligned} \mathcal{P}: &\quad h_{\text{in}} = h_0 \to h_1 \to \cdots \to h_m = h_{\text{out}}, \\ \mathcal{P}': &\quad h_{\text{in}} = \tilde{h}_0 \to \tilde{h}_1 \to \cdots \to \tilde{h}_n = h_{\text{out}}. \end{aligned} \tag{32}$$

Their corresponding micro-action credits are

$$\delta_j = V(h_j) - V(h_{j-1}), \qquad \tilde{\delta}_\ell = V(\tilde{h}_\ell) - V(\tilde{h}_{\ell-1}). \tag{33}$$

The two decompositions may contain different numbers of actions and assign different values to individual actions. Nevertheless, both telescope to the same semantic-step return:

$$\sum_{j=1}^{m} \delta_j = \sum_{\ell=1}^{n} \tilde{\delta}_\ell = V(h_{\text{out}}) - V(h_{\text{in}}) = A_{\text{sem}}. \tag{34}$$

Equation 34 exposes the key distinction: micro-action credits depend on how the reasoning operation is decomposed, whereas their aggregate return depends only on its boundary states. Changing token boundaries or lexical decomposition can therefore alter the number and distribution of token-level credits without changing the utility of the complete semantic operation. Token-level attribution is representation-dependent, while semantic-step return is invariant to internal decomposition.

Taken together, these results motivate semantic reasoning steps as the action granularity for ERSR. Although token-level returns are formally well defined, finite-sample estimation must resolve smaller effects across more actions, while the resulting credit remains sensitive to lexical realization and token decomposition. Semantic steps provide a suitable intermediate granularity, retaining finer credit assignment than trajectory-level rewards while aggregating tokens into complete reasoning operations whose return effects are easier to estimate and more stable across representations.

A.3 Structured Prompting and Reasoning-Step Segmentation

Structured Prompting. Reliable ERSR evaluation requires consistent boundaries between semantic reasoning steps. A standard chain-of-thought prompt encourages step-by-step reasoning but does not explicitly mark where one reasoning operation ends and the next begins, making boundary identification less reliable. We denote this prompting format as **Standard CoT**:

Standard CoT

```
Solve the problem step by step.  Put the final answer in
\boxed{{...}}.
```

Table 5: **Effect of explicit step-structured prompting on Qwen3-1.7B.** Both prompting formats use identical decoding settings.

Prompt	Math				Science		Logic		Avg.
	AMC23	AIME24	AIME25	MATH-500	GPQA-D	SciBench	LogicBench	LogiQA2.0	
Standard CoT	43.4	13.3	8.5	75.4	31.4	37.2	78.8	59.4	43.4
Explicit-Step CoT	44.4	13.8	9.6	75.4	31.0	37.5	79.1	59.8	43.8

Table 6: Compliance with Explicit-Step CoT on 1,024 sampled trajectories.

	Qwen3-1.7B	Gemma-4-E2B-it
Explicit-step compliance	1020/1024 (99.6%)	772/1024 (75.4%)

In practice, responses generated with Standard CoT often contain continuous derivations whose semantic boundaries are difficult to determine unambiguously. We therefore introduce **Explicit-Step CoT**, which requires the model to organize its reasoning using explicit step headings:

Explicit-Step CoT

```
Solve the problem step by step. Organize the reasoning with
headings ### Step 1, ### Step 2, and so on. Put the final answer
in \boxed{{...}}.
```

These headings make the reasoning structure explicit without sacrificing model performance. As shown in Table 5, Explicit-Step CoT matches or slightly improves Qwen3-1.7B across the evaluated benchmarks, increasing Avg@16 from 43.4 to 43.8. We therefore adopt Explicit-Step CoT as the unified prompting format throughout the paper, including training and evaluation for all methods as well as ERSR evaluation.

To assess how reliably different models follow the structured prompting format, we independently sample 1,024 trajectories from Qwen3-1.7B and Gemma-4-E2B-it and measure whether each response uses the required explicit step headings. As shown in Table 6, Qwen3-1.7B follows the format in 99.6% of trajectories, while Gemma-4-E2B-it achieves a lower compliance rate of 75.4%.

Reasoning-Step Segmentation. As shown in Table 6, explicit `Step N` headings are nearly universal for Qwen3-1.7B but appear in only 75.4% of Gemma-4-E2B-it trajectories. A segmentation procedure based solely on explicit step headings would therefore work well for Qwen but leave a substantial fraction of Gemma responses unsplit. To accommodate this model-dependent formatting behavior, we use a conservative hierarchical procedure. When a response contains a valid top-level sequence of `Step 1`, `Step 2`, ..., these headings directly define the reasoning steps. An independent `Final Answer` section is treated as a terminal segment, whereas a terminal `\boxed{...}` expression contained within the final reasoning section remains part of that step. If no valid step sequence is present, we apply a semantic-heading fallback, primarily for Gemma outputs, using repeated top-level headings that denote reasoning stages such as *Analysis*, *Evaluation*, *Inference*, *Calculation*, or *Conclusion*. Responses without reliable structural cues are retained as a single step to avoid speculative segmentation. Table 7 summarizes the hierarchy.

To minimize spurious boundaries, we do not split on ordinary paragraphs, generic numbering, answer choices, formulas, or local subheadings. Candidate headings inside protected regions such as code blocks, mathematical environments, or tables are also ignored. For semantic fallback, short or incomplete preambles are merged with the first reasoning section, and boundaries are retained only when the resulting sections contain sufficiently complete reasoning content.

Figure 5 presents a representative Gemma-4-E2B-it response that does not follow the prescribed `Step N` format. Instead, the model organizes its reasoning through a sequence of repeated semantic headings, including *Analysis*, *Evaluation*, and *Conclusion*. Our fallback rule recognizes these top-level headings as reliable reasoning boundaries and segments the response accordingly, while preserving the introductory preamble and terminal answer within the resulting semantic structure.

Table 7: Reasoning-step segmentation hierarchy.

Rule	Boundary signal	Safeguard
Explicit steps	Top-level consecutive `Step N` headings	Used whenever a valid sequence is present
Terminal answer	Independent `Final Answer` heading or terminal `\boxed{...}`	Separate an independent answer section; otherwise retain the answer in the final reasoning step
Semantic fallback	Repeated top-level reasoning headings	Requires at least two complete reasoning sections and excludes non-reasoning headings
Conservative fallback	No reliable structural signal	Keep the remaining response as a single step

Case: Semantic-Heading Fallback

Question. In a 10-year study of 400 fifth-graders, children praised for being smart were more likely to give up, lack confidence, and avoid risks. The same pattern was observed across different social stages and for both men and women. Which statement best supports this research conclusion?

A. Not all praise is positive, and effects vary depending on its content and form.
B. Praised students became more risk averse and less independent.
C. Praise and no-praise groups showed no academic-performance difference.
D. Praise had its greatest effect on the cleverest girls.

Generated response → four semantic segments

Segment A *initial preamble*

```
Here is the step-by-step analysis to determine the best supporting statement for the
research conclusion.
```

Segment B *semantic-heading boundary*

```
### Analysis of the Research Conclusion
```

The study concluded that kids who were praised for being smart were more likely to give up, lack confidence, and be afraid to take risks. The finding was repeated across different social stages and for both men and women.
The key takeaway is that praise for intelligence can have negative psychological consequences, including giving up, reduced confidence, and risk aversion.

Segment C *semantic-heading boundary*

```
### Evaluation of the Choices
```

The response evaluates the four choices individually. Choice A emphasizes that the effect of praise depends on its content and form; Choice B largely restates the observed finding; Choice C concerns academic performance rather than the reported psychological effects; and Choice D restricts the effect to a particular subgroup. *[intermediate details omitted]*

Segment D *semantic heading + terminal answer*

```
### Conclusion
```

Choice A provides the strongest explanatory context. It acknowledges that the effect of praise depends on the specific intervention. The correct choice is A.

```
\boxed{A}
```

Figure 5: **Representative semantic-heading fallback case.** Repeated top-level headings provide semantic boundaries when no explicit `Step N` sequence is present, yielding four segments with the terminal answer retained in the final segment.

Segmentation Statistics. Table 8 summarizes the distribution of segmented reasoning steps produced by Qwen3-1.7B across the eight evaluation benchmarks. Mathematical tasks generally induce longer reasoning chains, with AIME24 and AIME25 reaching mean step counts of 8.60 and 8.10, respectively, compared with 4.50 on LogicBench and 4.21 on LogiQA2.0. Most benchmarks exhibit median step counts between 4 and 8.5, indicating that the structured prompting typically yields

Table 8: Reasoning-step statistics for Qwen3-1.7B across eight evaluation benchmarks. Statistics are computed over the number of segmented reasoning steps per trajectory.

Statistic	**Math**				**Science**		**Logic**	
	AMC23	**AIME24**	**AIME25**	**MATH-500**	**GPQA-D**	**SciBench**	**LogicBench**	**LogiQA2.0**
Median	6.0	8.5	8.0	5.0	6.0	5.0	4.5	4.0
Mean	7.0	8.6	8.1	5.8	6.0	5.1	4.5	4.2
Min.	1.0	3.0	4.0	3.0	3.0	1.0	2.0	2.0
Max.	21.0	17.0	13.0	48.0	18.0	8.0	6.0	7.0

compact multi-step trajectories rather than excessively fragmented reasoning. The observed ranges also capture substantial variation in reasoning depth, with MATH-500 reaching up to 48 steps while LogicBench and LogiQA2.0 remain below 8. Overall, the statistics show that the segmentation procedure adapts naturally to the differing reasoning complexity of the evaluated domains.

A.4 ERSR Evaluation and Additional Analyses

A.4.1 Evaluation Pipeline

Data and trajectory sampling. We first filter DAPO-17K to retain only English-language problems, removing Chinese examples and obtaining 14,042 questions. For each question, we sample five rollouts from Qwen3-1.7B using a maximum response length of 8,192, temperature 1.0, and top-p 1.0, with thinking mode disabled. We use the explicit step-structured prompting format described in Appendix A.3 to facilitate subsequent reasoning-step segmentation. We then select 1,200 questions stratified by the within-question success rate of the five student rollouts. Specifically, we sample 200 questions from each success-rate group $\{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$, yielding 6,000 student trajectories in total. Group success rates are computed from binary answer correctness. We additionally restrict selection to questions whose five rollouts are all non-truncated, preventing truncation-induced failures from distorting the group-level success rate.

Reasoning-step and teacher-replacement sampling. We segment the 6,000 trajectories using the procedure in Appendix A.3 and sample approximately five reasoning steps per trajectory, resulting in 30,000 evaluated steps. We exclude the final reasoning step because it typically contains the final answer and therefore provides a less informative intervention point. For every sampled student step s_k, we generate a teacher replacement s_k^T from Qwen3-4B-Instruct-2507 conditioned on the same prefix h_{k-1}. Since semantic step boundaries cannot be enforced reliably with a generation delimiter, we allow the teacher to generate up to 2,048 tokens with temperature 1.0 and top-p 1.0, and then apply the segmentation procedure from Appendix A.3 to retain only the first complete teacher reasoning step. The relatively long generation budget is used to maximize the likelihood that this first semantic step is generated in full.

Monte Carlo return estimation. For each of the 30,000 sampled reasoning steps, we evaluate the three states defined by ERSR: the prefix before the student step h_{k-1}, the state after preserving the student step h_k, and the state after replacing it with the teacher step h_k^T. From each state, we independently sample 128 continuations using the student policy and estimate

$$\widehat{V}_{k-1} = \widehat{V}_{128}(h_{k-1}), \qquad \widehat{V}_k = \widehat{V}_{128}(h_k), \qquad \widehat{V}_k^T = \widehat{V}_{128}(h_k^T). \tag{35}$$

These estimates yield the student-step and teacher-replacement advantages,

$$\widehat{A}_k = \widehat{V}_k - \widehat{V}_{k-1}, \qquad \widehat{A}_k^{\mathrm{TR}} = \widehat{V}_k^T - \widehat{V}_{k-1}. \tag{36}$$

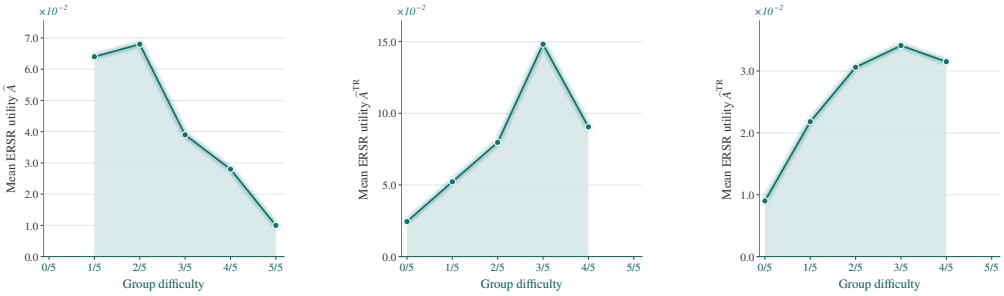
Overall, the evaluation requires $30{,}000 \times 3 \times 128 = 11.52$ million student-policy continuations, providing high-fidelity MC@128 estimates for the offline ERSR analysis.

A.4.2 ERSR across Teachers and Problem Difficulty

Robustness across Teachers. To assess whether the observed ERSR patterns depend on a particular teacher, we additionally use Qwen3-4B to generate replacement actions for the same 30,000 sampled reasoning steps and estimate the corresponding teacher-replacement utility $\widehat{A}^{\mathrm{TR}}$ with MC@128

Table 9: **Teacher-replacement ERSR utility across teachers.** Values report mean $\widehat{A}^{\mathrm{TR}}$ with confidence intervals under successful and failed student trajectories.

Teacher	$R = 1$	$R = 0$
Qwen3-4B-Instruct-2507	$+0.007$ [$+0.000$, $+0.013$]	$\mathbf{+0.0638}$ [$+0.054$, $+0.075$]
Qwen3-4B	$+0.0146$ [$+0.009$, $+0.020$]	$+0.0218$ [$+0.017$, $+0.027$]



(a) Student-step utility $\widehat{A}$ (b) $\widehat{A}^{\mathrm{TR}}$ with Qwen3-4B-Instruct-2507 (c) $\widehat{A}^{\mathrm{TR}}$ with Qwen3-4B

Figure 6: **ERSR utility across problem difficulty.** Student-step utility $\widehat{A}$ is evaluated on successful trajectories, while teacher-replacement utility $\widehat{A}^{\mathrm{TR}}$ is evaluated on failed trajectories for the two teachers.

under the identical evaluation protocol. Table 9 shows that both teachers exhibit the same outcome-dependent pattern: teacher replacement is more beneficial on failed than on successful trajectories. For Qwen3-4B-Instruct-2507, the mean $\widehat{A}^{\mathrm{TR}}$ increases from $+0.007$ on successful trajectories to $+0.0638$ on failed ones; Qwen3-4B shows the same direction, increasing from $+0.0146$ to $+0.0218$. This consistency indicates that the outcome-dependent ERSR asymmetry is robust to teacher choice. Moreover, Qwen3-4B-Instruct-2507 substantially outperforms Qwen3-4B as a teacher in Table 16 and yields substantially larger replacement utility on failed trajectories. Although based on a limited set of teachers, this alignment suggests that ERSR may also reflect differences in the learning benefit available from the selected teacher.

ERSR across problem difficulty. Figure 6 examines ERSR across groups defined by the question success rate. We compute $\widehat{A}$ only over successful trajectories ($R = 1$) and $\widehat{A}^{\mathrm{TR}}$ only over failed trajectories ($R = 0$); consequently, $\widehat{A}$ is undefined for the $0/5$ group, while $\widehat{A}^{\mathrm{TR}}$ is undefined for the $5/5$ group. Student-step utility generally decreases as group accuracy increases, with $\widehat{A}$ falling sharply from the intermediate-difficulty groups toward nearly zero at $5/5$, indicating diminishing benefit from self-reinforcement once the student already solves a problem reliably. Teacher-replacement utility instead exhibits a non-monotonic pattern for both teachers, peaking at intermediate difficulty. Importantly, the low $\widehat{A}^{\mathrm{TR}}$ on $0/5$ problems should not be interpreted as low value of teacher intervention. On problems where all student rollouts fail, a trajectory may contain multiple erroneous decisions, so replacing a single reasoning step is often insufficient to recover the final answer. ERSR therefore measures the local utility of a single-step replacement rather than the cumulative benefit of repeatedly distilling teacher supervision across an entire trajectory.

A.4.3 Low-Cost ERSR Estimation

High-fidelity ERSR estimation with MC@128 is suitable for offline analysis but becomes prohibitively expensive when repeated across training checkpoints. Since our goal during training is not to recover the utility of every individual reasoning step, but to estimate the aggregate expected benefit of continued optimization, a substantially cheaper approximation may be sufficient. We therefore study whether ERSR can be reliably estimated using fewer sampled reasoning steps and a much smaller Monte Carlo budget per step, enabling practical training-time tracking of the remaining utility of reinforcement and teacher distillation.

Table 10: **Low-cost approximation of aggregate ERSR.** Deviations are measured against the full MC@128 estimates ($\widehat{A} = +0.032$, $\widehat{A}^{\mathrm{TR}} = +0.064$). The shaded row denotes the configuration used for training-time tracking.

Estimation Budget		$\widehat{A}$		$\widehat{A}^{\mathrm{TR}}$	
Steps	**MC@N**	**Typical Dev.**	**95% Bound**	**Typical Dev.**	**95% Bound**
250	4	0.0146	0.0358	0.0226	0.0554
500	2	0.0133	0.0326	0.0177	0.0435
500	8	0.0084	0.0207	0.0150	0.0368
1000	8	0.0060	0.0147	0.0106	0.0260

We use the aggregate MC@128 estimates over the full ERSR evaluation set as reference values and construct low-cost estimates by subsampling fewer reasoning steps and reducing the number of continuations per state. We evaluate the branch-level quantities $\mathbb{E}[\widehat{A} \mid R = 1]$ and $\mathbb{E}[\widehat{A}^{\mathrm{TR}} \mid R = 0]$, rather than individual step utilities. Table 10 reports the resulting estimation deviations for representative budgets.

The aggregate estimates remain informative even under substantially reduced budgets. Notably, 500 steps with MC@2 yields smaller deviations than 250 steps with MC@4 for both utilities despite the same nominal sampling budget, suggesting that broader coverage of reasoning steps is more valuable than allocating additional continuations to fewer steps when estimating population-level ERSR. Increasing either step coverage or the per-step MC budget further reduces error, as expected.

Based on this trade-off, we use 500 sampled steps with MC@2 for training-time ERSR tracking. As each checkpoint is independently re-estimated, these low-cost estimates may vary from the full MC@128 values due to sampling error. This approximation is intended to recover the overall remaining utility of a learning branch—whether reinforcement or teacher correction still provides substantial expected benefit—rather than to accurately estimate individual-step utilities or fine-grained subgroup statistics. We retain MC@128 for the offline ERSR analyses where such fine-grained comparisons are required.

A.5 Answer-Probe Runtime Analysis

The packed answer-probe design introduces additional probe tokens into the student forward pass but does not require separate model evaluations for individual reasoning states. All probe blocks are evaluated jointly with the original trajectory using the customized attention mask and position IDs described in Section 3.2, while probe tokens are excluded from the optimization loss. Consequently, the additional cost mainly arises from processing the packed probe tokens rather than repeatedly recomputing each reasoning prefix.

We measure the practical training overhead under three teacher–student configurations using identical settings for OPD and R^{2}OPL. All experiments run on the same 8×NVIDIA H20 setup with a maximum response length of 8,192. We report the average wall-clock time per training step, together with the time spent on rollout generation and actor updates. Evaluation, model initialization, and checkpointing are excluded from the measurements.

As shown in Table 11, the packed probe computation introduces only modest wall-clock overhead across all three configurations. Total training time increases by 2.0% for Gemma-4-E2B-it, 8.1% for Qwen3-1.7B, and 6.0% for Qwen3-4B. Generation time changes only slightly, while most of the additional cost appears in the actor update where the packed probe tokens are processed. These results show that step-level answer-probe modulation can be incorporated into on-policy training without requiring separate probe forwards, with training overhead remaining below approximately 8% and dropping to as little as 2% in the Gemma configuration.

A.6 Dataset Details

Training Data Construction. We construct a 32K mixed training set covering mathematical, scientific, and logical reasoning. The mathematical portion is drawn from DAPO-17K, the scientific

Table 11: **Training overhead of R^{2}OPL.** All methods use identical training settings. Times are average seconds per training step and exclude evaluation, initialization, and checkpointing.

Teacher → Student	Method	Time per Step (s)			Overhead
		Total	Generation	Actor Update	
Gemma-4-26B-A4B → Gemma-4-E2B-it	OPD	230.7	74.5	119.4	–
	R^{2}OPL	235.3	75.3	123.3	+2.0%
Qwen3-4B-Instruct-2507 → Qwen3-1.7B	OPD	102.7	41.4	43.6	–
	R^{2}OPL	111.0	43.3	49.5	+8.1%
Qwen3-4B-Instruct-2507 → Qwen3-4B	OPD	147.5	55.2	68.8	–
	R^{2}OPL	156.3	55.5	78.0	+6.0%

Table 12: Training-data composition across the three reasoning domains.

Dataset	Domain	Public Size	Eligible Pool	Selected
DAPO-17K	Math	∼17.4K	14,042	13,000
Nemotron-Science-v1 MCQ	Science	174,155	174,155	10,000
LogiQA2.0	Logic	12,567	12,567	9,000
Total	–	–	–	**32,000**

portion from Nemotron-Science-v1 MCQ, and the logical portion from the LogiQA2.0 training split. For DAPO-17K, we first remove Chinese-language problems, leaving 14,042 English examples, and randomly sample 13,000 from this filtered pool. We further sample 10,000 examples from Nemotron-Science-v1 MCQ and 9,000 from LogiQA2.0, yielding 32,000 training examples in total. Table 12 summarizes the resulting composition.

Evaluation Set Construction. We evaluate on eight benchmarks spanning mathematical, scientific, and logical reasoning. To reduce evaluation cost, benchmarks containing more than 200 examples are randomly subsampled to 200 examples, while smaller benchmarks are evaluated in full, yielding 1,098 evaluation problems in total. All methods are evaluated on exactly the same subsets using identical decoding settings. Table 13 summarizes the original and evaluated sizes together with the corresponding task formats. For all benchmarks except SciBench, correctness is determined by exact numerical or symbolic equivalence between the extracted `\boxed{...}` answer and the reference answer, including option labels for multiple-choice tasks. For SciBench, we follow its official protocol and allow a 5% numerical tolerance. When a target unit is required, we append it to the question to avoid errors caused solely by unit or scale mismatches.

A.7 Details of Compared Methods

We provide additional details of the compared methods evaluated in our experiments. Following the categorization in the main paper, existing approaches can be broadly divided into three groups: reward-only optimization, distillation-only optimization, and reward–distillation hybrid optimization. For each method, we briefly summarize its core optimization objective and highlight the key difference from other approaches. All methods are described using their representative objectives for clarity, while implementation details follow the original formulations unless otherwise specified.

GRPO. Group Relative Policy Optimization (GRPO) (Guo et al., 2025) optimizes the policy using group-relative rewards from multiple sampled responses for the same prompt. For a sampled group $\{\tau_i\}_{i=1}^{G}$, GRPO optimizes

$$J_{\mathrm{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(r_{i,t}(\theta)\hat{A}_i, \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)\right]. \tag{37}$$

Table 13: Evaluation-set composition across mathematical, scientific, and logical benchmarks.

Dataset	Domain	Task Format	Original Size	Evaluated
AMC23	Math	Open-ended short answer	40	40
AIME24	Math	Numerical fill-in	30	30
AIME25	Math	Numerical fill-in	30	30
MATH-500	Math	Open-ended short answer	500	200
GPQA-Diamond	Science	4-choice MCQ	198	198
SciBench	Science	Open-ended numerical	580	200
LogicBench	Logic	4-choice MCQ	500	200
LogiQA2.0	Logic	4-choice MCQ	1,572	200
Total	–	–	3,450	**1,098**

The group-relative advantage and policy ratio are defined as

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R}, \qquad r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid h_{i,t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid h_{i,t})}, \tag{38}$$

where μ_R and σ_R denote the mean and standard deviation of rewards within the group.

DAPO. Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2026a) extends GRPO with four stabilization techniques: asymmetric clipping, dynamic sampling, token-level loss normalization, and overlong reward shaping. DAPO adopts the same group-relative advantage $\hat{A}_i$ and policy ratio $r_{i,t}(\theta)$ as GRPO, while modifying the policy objective as

$$J_{\text{DAPO}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_i T_i}\sum_{i=1}^{G}\sum_{t=1}^{T_i}\min\left(r_{i,t}(\theta)\hat{A}_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_i\right)\right], \tag{39}$$

where ϵ_{high} and ϵ_{low} are asymmetric clipping thresholds. Unlike GRPO, DAPO over-samples and filters groups with uniformly correct or incorrect responses, and applies soft overlong punishment to gradually penalize responses approaching the maximum generation length.

GSPO. Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) extends group-based policy optimization by performing optimization at the sequence level instead of the token level. Given a sampled group $\{\tau_i\}_{i=1}^{G}$, GSPO uses the same group-relative advantage $\hat{A}_i$ as GRPO, but defines a sequence-level importance ratio:

$$s_i(\theta) = \left(\frac{\pi_\theta(\tau_i \mid x)}{\pi_{\theta_{\text{old}}}(\tau_i \mid x)}\right)^{\frac{1}{|\tau_i|}}, \tag{40}$$

where $|\tau_i|$ denotes the sequence length. The optimization objective is

$$J_{\text{GSPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(s_i(\theta)\hat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)\right]. \tag{41}$$

Compared with GRPO, GSPO avoids token-level policy ratios and performs policy updates using sequence-level importance weighting.

OPD. On-Policy Distillation (OPD) (Agarwal et al., 2024) trains the student on trajectories sampled from its current policy, aligning distillation with student-visited states. For each student-generated token, we define the teacher–student log-likelihood discrepancy as

$$A_t^{\text{OPD}} = \log \pi_{\text{T}}(y_t \mid h_t) - \log \pi_\theta(y_t \mid h_t). \tag{42}$$

We use this discrepancy as the token-level optimization signal:

$$J_{\text{OPD}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=1}^{T}\text{sg}\big(A_t^{\text{OPD}}\big)\log \pi_\theta(y_t \mid h_t)\right], \tag{43}$$

where π_{T} denotes the fixed teacher policy and $\text{sg}(\cdot)$ denotes the stop-gradient operator.

ExOPD. Exploratory On-Policy Distillation (ExOPD) (Yang et al., 2026d) extends OPD through reward extrapolation, amplifying teacher-preferred actions relative to a reference policy. Specifically, it modifies the OPD advantage as

$$\widehat{A}_t^{\mathrm{ExOPD}} = \widehat{A}_t^{\mathrm{OPD}} + (\lambda - 1)\left[\log \pi_{\mathrm{T}}(y_t|h_t) - \log \pi_{\mathrm{ref}}(y_t|h_t)\right], \tag{44}$$

where λ controls the extrapolation strength. ExOPD then optimizes the corresponding policy-gradient objective

$$J_{\mathrm{ExOPD}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{T} \widehat{A}_t^{\mathrm{ExOPD}} \log \pi_\theta(y_t|h_t)\right]. \tag{45}$$

EOPD. Entropy-Aware On-Policy Distillation (EOPD) (Jin et al., 2026) augments standard OPD with forward-KL supervision at token positions where the teacher exhibits high uncertainty. For each context h_t, the teacher entropy is

$$H_t^T = -\sum_{v\in\mathcal{V}} \pi_T(v \mid h_t) \log \pi_T(v \mid h_t), \tag{46}$$

where $\mathcal{V}$ denotes the vocabulary. EOPD then defines

$$\mathcal{L}_t^{\mathrm{EOPD}} = \mathcal{L}_t^{\mathrm{OPD}} + \alpha\, \mathbb{I}\big[H_t^T > \tau\big]\, \mathcal{L}_t^{\mathrm{FKL}}, \qquad \mathcal{L}_t^{\mathrm{FKL}} = \mathrm{KL}(\pi_T(\cdot \mid h_t) \,\|\, \pi_\theta(\cdot \mid h_t)), \tag{47}$$

where τ is the entropy threshold and α controls the forward-KL strength. Thus, EOPD reduces to standard OPD when the teacher is confident, while additionally applying mode-covering forward KL at high-entropy positions.

SRPO. Sample-Routed Policy Optimization (SRPO) (Li et al., 2026b) combines reinforcement learning and self-distillation through outcome-conditioned routing. For rollout i, let $c_i = \mathbb{I}[R_i = 1]$ and

$$z_i^{\mathrm{SD}} = (1 - c_i)m_i, \qquad z_i^{\mathrm{RL}} = 1 - z_i^{\mathrm{SD}}, \tag{48}$$

where m_i indicates teacher-information availability. Correct trajectories are routed to GRPO, while failed trajectories with teacher information are distilled. SRPO further applies entropy-aware weighting,

$$\widetilde{w}_{i,t} = \exp(-\beta H_{i,t}), \qquad w_{i,t} = \frac{\widetilde{w}_{i,t}}{\frac{1}{|\Omega_{\mathrm{SD}}|}\sum_{(j,s)\in\Omega_{\mathrm{SD}}} \widetilde{w}_{j,s}}, \tag{49}$$

and optimizes

$$\mathcal{L}_{\mathrm{SRPO}} = \frac{\sum_{i,t} z_i^{\mathrm{RL}} \ell_{i,t}^{\mathrm{GRPO}} + \sum_{i,t} z_i^{\mathrm{SD}} w_{i,t} \ell_{i,t}^{\mathrm{SD}}}{\sum_{i,t} z_i^{\mathrm{RL}} + \sum_{i,t} z_i^{\mathrm{SD}}}. \tag{50}$$

In our experiments, we replace the original feedback-conditioned self-teacher with the external teacher while retaining the routing and entropy-aware weighting.

RLSD. RLVR with Self-Distillation (RLSD) (Yang et al., 2026b) combines trajectory-level reward supervision with token-level distillation-based credit assignment. Given a group-relative advantage A_i, RLSD computes

$$\Delta_{i,t} = \mathrm{sg}[\log \pi_T(y_{i,t} \mid h_{i,t}) - \log \pi_\theta(y_{i,t} \mid h_{i,t})], \qquad w_{i,t} = \exp(\mathrm{sign}(A_i)\Delta_{i,t}), \tag{51}$$

and redistributes the trajectory advantage across tokens as

$$\widehat{A}_{i,t}^{\mathrm{RLSD}} = A_i\left[(1-\lambda) + \lambda\, \mathrm{clip}(w_{i,t}, 1-\epsilon_w, 1+\epsilon_w)\right]. \tag{52}$$

Here, reward determines the update direction, while distillation modulates its token-level magnitude. In our experiments, we replace the original privileged self-teacher with the external teacher while retaining the same reweighting mechanism.

OPDVR. On-Policy Distillation with Verifiable Reward (OPDVR) (Lin et al., 2026) uses trajectory correctness to filter the token-level OPD signal. Let

$$r_{i,t}^{\mathrm{OPD}} = \log \pi_T(y_{i,t} \mid h_{i,t}) - \log \pi_\theta(y_{i,t} \mid h_{i,t}). \tag{53}$$

OPDVR retains only reward-consistent updates:

$$\widehat{A}_{i,t}^{\mathrm{OPDVR}} = \begin{cases} \max(0, r_{i,t}^{\mathrm{OPD}}), & R_i = 1, \\ \min(0, r_{i,t}^{\mathrm{OPD}}), & R_i = 0. \end{cases} \tag{54}$$

Thus, verifiable rewards determine the update direction, while the teacher–student discrepancy controls its magnitude.

Table 14: **Shared training and evaluation configurations.** Unless otherwise specified, all methods use the same optimization, rollout, and evaluation settings.

Hyperparameter	Value
Training framework	`verl`
Numerical precision	`bfloat16`
Optimizer	AdamW
Learning rate	1×10^{-6}
Learning-rate schedule	Cosine
Warmup	None
Adam coefficients	$(\beta_1, \beta_2) = (0.9, 0.999)$
Weight decay	0.01
Gradient clipping	1.0
KL regularization	None
Optimization epochs per rollout batch	1
PPO mini-batch size	64
Micro-batch size per GPU	1
Questions per batch	64
Responses per question	4
Responses per optimization step	256
Training response length	8,192
Training temperature	1.0
Training top-p	1.0
Rollout backend	vLLM (asynchronous)
Rollout precision	`bfloat16`
Tensor parallel size	1
Random seed	42
Evaluation response length	16,384
Evaluation temperature	0.6
Evaluation top-p	0.95
Evaluation samples	16 (Avg@16)
Training steps	100 (distillation-only); 500 (reward-based and hybrid)
Prompt format	Explicit-Step CoT

A.8 Training Configurations and Hyperparameters

Table 14 summarizes the shared training and evaluation configuration used across all methods. We implement all experiments with `verl` using AdamW optimization in `bfloat16` precision, a learning rate of 1×10^{-6} with cosine decay and no warmup, and no auxiliary KL regularization. Each optimization step samples four responses for each of 64 prompts, yielding 256 trajectories, with a PPO mini-batch size of 64 and a per-GPU micro-batch size of 1. Rollouts are generated asynchronously with vLLM using tensor parallel size 1 and a fixed random seed of 42. All experiments are conducted on eight NVIDIA H20 GPUs. Unless otherwise specified, all methods share these settings, while method-specific hyperparameters are reported separately below.

A.9 Results with Qwen3-4B as the Student

Table 16 further evaluates R^{2}OPL with the stronger Qwen3-4B student. R^{2}OPL achieves an Avg@16 of 73.1, outperforming the strongest baseline, OPDVR, by 5.1 points and improving substantially over both GRPO (62.0) and standard OPD (67.6). The gains are consistent across all eight benchmarks, with R^{2}OPL achieving the best result among the compared training methods on every task. Notably, the optimized Qwen3-4B student also surpasses its same-scale Qwen3-4B-Instruct-2507 teacher in average performance, achieving 73.1 versus 72.5. It exceeds the teacher on six of eight benchmarks, including AMC23, AIME25, MATH-500, GPQA-Diamond, SciBench, and LogicBench, showing that R^{2}OPL can integrate reward-based self-improvement with teacher supervision to improve beyond the performance of the teacher itself.

Table 15: **Method-specific hyperparameters.** All unlisted optimization, rollout, and evaluation settings follow Table 14.

Method	Hyperparameter	Setting
GRPO	Policy clipping	$\epsilon = 0.2$; group-relative reward standardization
DAPO	Asymmetric clipping; dynamic sampling; loss aggregation; overlong shaping	$\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$; filter groups with uniform rewards; token-mean loss aggregation; overlong penalty factor 1.0
GSPO	Sequence-level clipping; loss aggregation	$\epsilon_{\text{low}} = 3 \times 10^{-4}$, $\epsilon_{\text{high}} = 4 \times 10^{-4}$; sequence-level importance ratio; seq-mean-token-mean aggregation
OPD	Distillation estimator	Sampled-token reverse-KL signal $\log \pi_T - \log \pi_\theta$; no additional method-specific coefficient
ExOPD	Reward extrapolation coefficient	$\lambda_{\text{Ex}} = 1.25$
EOPD	Entropy threshold; forward-KL coefficient; top-k	$\tau = 0.8$; $\alpha = 1.0$; $k = 16$
SRPO	Routing; entropy weighting; distillation support	Correct trajectories $\rightarrow$ GRPO; failed trajectories $\rightarrow$ distillation; $\beta = 1$; top-$K = 100$
RLSD	Distillation mixing; evidence-weight clipping	$\lambda_{\text{RLSD}} : 0.5 \rightarrow 0$ linearly over the first 50 steps; $\epsilon_w = 0.2$
OPDVR	Correctness gating	Sampled-token OPD with correctness gating; no additional tunable hyperparameters
R^{2}OPL	Branch scaling; probe modulation;	$\mu = 10$, $\lambda = 0.1$, $\alpha_R = 0.25$, $\alpha_D = 0.5$

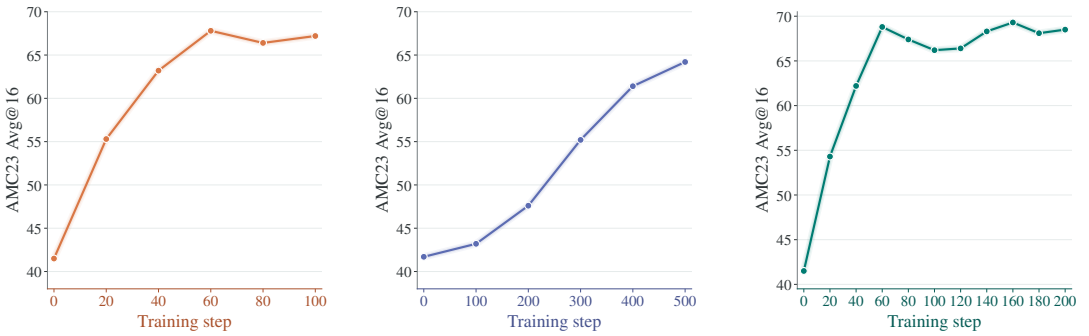
Table 16: **Results on Qwen3-4B-Instruct-2507 $\rightarrow$ Qwen3-4B.**

Method	Math				Science		Logic		Avg.
	AMC23	AIME24	AIME25	MATH-500	GPQA-D	SciBench	LogicBench	LogiQA2.0	
Qwen3-4B-Instruct-2507 $\rightarrow$ Qwen3-4B									
Reference Models									
Student	65.3	19.6	18.1	84.8	40.9	49.6	84.8	65.8	53.6
Teacher	92.8	59.2	44.6	96.6	58.5	63.2	85.2	80.1	72.5
Reward Only									
GRPO	80.2	38.5	27.9	92.2	46.0	55.8	84.4	70.9	62.0
Distillation Only									
OPD	88.0	49.0	41.9	93.2	49.6	59.6	85.9	73.8	67.6
Reward–Distillation Hybrid									
OPDVR	88.4	49.6	42.7	93.9	50.2	59.4	86.1	73.7	68.0
R^{2}OPL	**93.4**	**55.8**	**46.0**	**96.9**	**59.3**	**65.8**	**88.3**	**79.5**	**73.1**

A.10 Balancing RL and OPD in R^{2}OPL

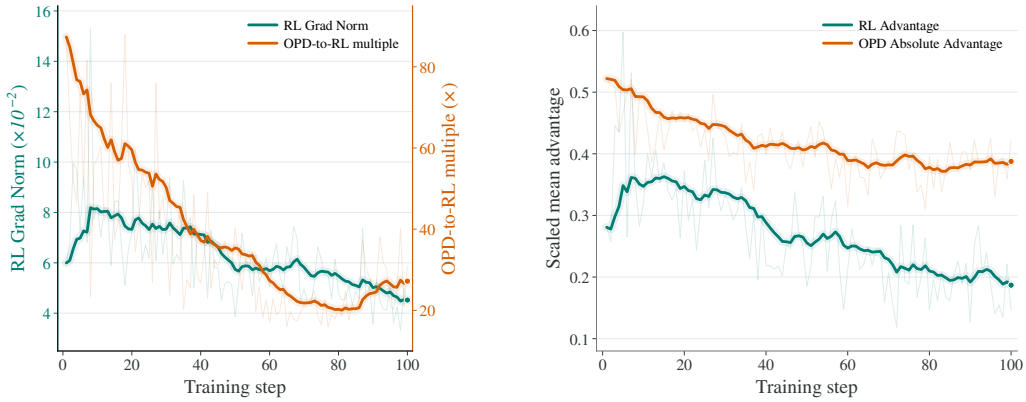
Why Are μ and λ Highly Asymmetric? In R^{2}OPL, μ and λ control the relative optimization strengths of the reinforcement and distillation branches. Although a symmetric setting may appear natural, the two branches differ substantially in their learning dynamics and effective optimization influence. We therefore treat μ and λ as balancing coefficients for the two learning processes, rather than as direct indicators of the relative importance of reward and teacher supervision.

Figure 7 first highlights the markedly different optimization timescales of the two learning sources. OPD improves rapidly and has largely converged by around 40 steps, whereas GRPO continues to improve through 500 steps without showing a clear convergence trend. Under the symmetric setting $\mu = \lambda = 1$, R^{2}OPL closely follows the OPD-like dynamics, converging by roughly 60 steps



(a) OPD, 100 steps (b) GRPO, 500 steps (c) R^{2}OPL ($\mu = 1, \lambda = 1$), 200 steps

Figure 7: **AMC23 training dynamics under OPD, GRPO, and R^{2}OPL with equal branch scaling.** Curves report Avg@16 across training checkpoints.



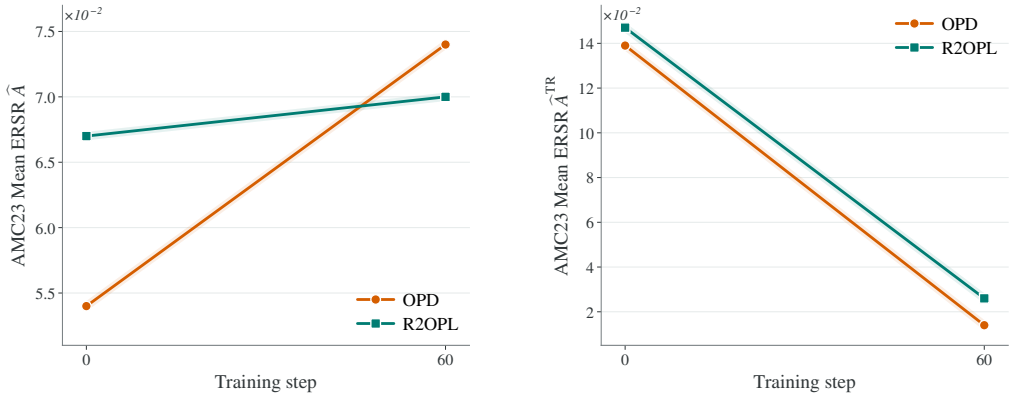
(a) RL–OPD gradient dynamics (b) RL–OPD advantage magnitudes

Figure 8: **Branch-scale imbalance in R^{2}OPL under equal weighting ($\mu = \lambda = 1$).** We compare the gradient norms and advantage magnitudes of the successful-trajectory RL branch and failed-trajectory OPD branch during training.

and reaching a performance within two points of OPD. Thus, simply combining the two branches with equal coefficients largely preserves the fast convergence behavior of distillation rather than the slower continued improvement of reinforcement learning.

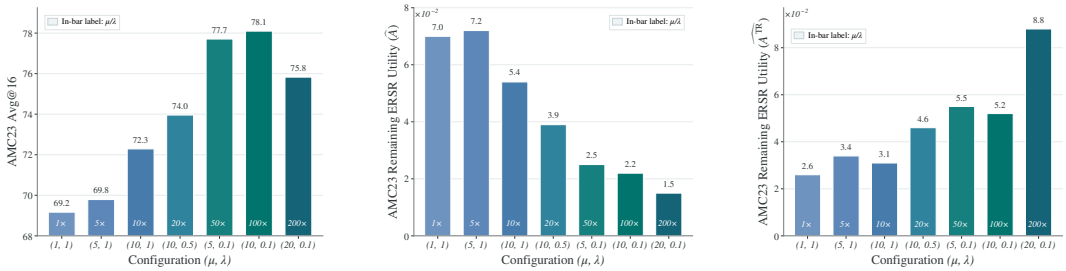
To understand this imbalance, Figure 8 compares the branch-wise gradient norms and mean absolute advantages under $\mu = \lambda = 1$. The OPD gradient norm is nearly two orders of magnitude larger than the RL gradient at the beginning of training and remains more than an order of magnitude larger throughout the first 100 steps. In contrast, the mean absolute OPD advantage is only about 1.5–$2\times$ that of RL. The much larger disparity in gradient scale therefore cannot be explained by advantage magnitude alone, indicating that token-level OPD gradients aggregate substantially more coherently during optimization. This pronounced effective imbalance motivates substantially down-scaling the distillation branch beyond what would be suggested by the nominal advantage scales alone.

ERSR provides a direct view of which learning utility is realized under equal branch weighting. From step 0 to 60, both OPD and R^{2}OPL leave the student-step utility $\widehat{A}$ largely unchanged, while sharply reducing the teacher-replacement utility $\widehat{A}^{\mathrm{TR}}$. R^{2}OPL with $\mu = \lambda = 1$ therefore exhibits essentially the same utility-consumption pattern as OPD: teacher-distillation utility is rapidly internalized, whereas reinforcement utility remains largely unexploited. Despite explicitly combining both branches, equal weighting thus leads to OPD-dominated optimization and fails to effectively internalize the student's own high-value actions.



(a) AMC23 Mean ERSR $\widehat{A}$ (b) AMC23 Mean ERSR $\widehat{A}^{\mathrm{TR}}$

Figure 9: **ERSR utility dynamics under equal branch scaling ($\mu = \lambda = 1$).** We compare the remaining student-step utility $\widehat{A}$ and teacher-replacement utility $\widehat{A}^{\mathrm{TR}}$ on AMC23 at steps 0 and 60 for OPD and R^{2}OPL.



(a) Best AMC23 Avg@16 (b) Remaining ERSR utility $\widehat{A}$ (c) Remaining ERSR utility $\widehat{A}^{\mathrm{TR}}$

Figure 10: **Effect of RL–OPD branch scaling in R^{2}OPL.** For each (μ, λ) configuration, we report the best AMC23 Avg@16 over training and the remaining student-step and teacher-replacement ERSR utilities at the corresponding best checkpoint.

Asymmetric Scaling Enables Dual Utility Realization. The above results show that nominally balanced RL and OPD signals can lead to strongly imbalanced optimization. We therefore vary μ and λ over a wide range of relative branch strengths and train each configuration for 500 steps. For each setting, Figure 10 reports the best AMC23 Avg@16 together with the remaining reinforcement utility $\widehat{A}$ and teacher-distillation utility $\widehat{A}^{\mathrm{TR}}$ at the corresponding best checkpoint. As the relative RL strength μ/λ increases, the remaining $\widehat{A}$ generally decreases, indicating progressively stronger realization of reinforcement utility, while $\widehat{A}^{\mathrm{TR}}$ eventually increases as teacher distillation becomes underweighted. This trade-off is reflected directly in performance: Avg@16 rises from 69.2 at the symmetric $1\times$ setting to 78.1 at $100\times$, where both remaining utilities are kept relatively low. Further increasing the ratio to $200\times$ reduces $\widehat{A}$ to 0.015 but leaves substantially more teacher-distillation utility ($\widehat{A}^{\mathrm{TR}} = 0.088$), accompanied by a performance drop to 75.8. These results identify a regime in which neither branch dominates the optimization and both sources of expected-return utility are effectively internalized. We therefore use $\mu = 10$ and $\lambda = 0.1$ in R^{2}OPL, corresponding to a $100\times$ relative scaling between the RL and OPD branches.

This persistent OPD dominance can be understood from the directional coherence of the two learning signals. OPD provides dense token-level supervision toward a common teacher policy, causing gradients to align more consistently both within a batch and across successive batches; consequently, comparable advantage magnitudes can produce substantially larger aggregate gradients than reward-based reinforcement. This coherence also interacts with AdamW's moment accumulation. Consistently directed OPD gradients accumulate in the first moment, whereas the more variable RL

gradients can partially cancel across updates while still contributing to the second moment. As a result, the effective update, which is governed by the ratio between the accumulated first and second moments, can remain strongly biased toward OPD even when its instantaneous gradient norm is smaller than that of RL. This provides an optimization-level explanation for why teacher supervision can continue to dominate joint on-policy learning after substantial down-scaling.

Unless otherwise specified, all training-time ERSR evaluations in Section 4.3 and Appendix A.10 use 500 sampled reasoning steps with MC@2, following Appendix A.4.3.