BEYOND TOKEN-LOCAL IMITATION: REWARD-COMPATIBLE TEMPORAL CREDIT ASSIGNMENT FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as an effective approach for large language model post-training, yet existing objectives face a trade-off between objective fidelity and optimization stability. Token-level OPD provides stable but local supervision, whereas sequence-level OPD captures future credit at the cost of horizon-dependent variance. We establish a unified temporal-credit view of these formulations, showing that practical token-level OPD can be interpreted as a temporal approximation to the sequence-level reverse-KL gradient. Building on this connection, we propose γOPD, which uses discounted temporal credit assignment to balance long-horizon supervision and optimization stability, while admitting a horizon-independent variance bound. We further develop a reward-compatible bounded mixing (RBM) mechanism for γOPD that balances verifiable outcome feedback with the discounted OPD advantage to move beyond purely teacher-dependent optimization. Experiments on mathematical and code reasoning demonstrate consistent improvements over existing OPD methods across vanilla, size-mismatched, and multi-teacher distillation settings.

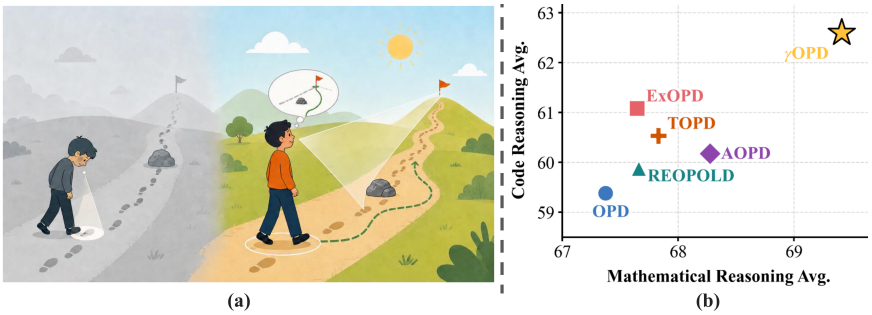


Figure 1: **Core Idea.** (a) Vanilla OPD (left) focuses only on the immediate next step and closely follows the teacher's local guidance. We argue that OPD should instead account for the influence of future reasoning steps and should not merely imitate the teacher token by token (right). (b) By incorporating future outcomes through temporal credit assignment and balancing teacher supervision with verifiable rewards, γOPD achieves strong overall performance across mathematical and code reasoning tasks in the multi-teacher distillation setting, as detailed in Section 4.

1 INTRODUCTION

Recent large language models (LLMs), including GPT-5 (OpenAI, 2025), GLM-5 (GLM-5 Team, 2026), DeepSeek-V4 (DeepSeek-AI, 2026), and Nemotron-Cascade 2 (Yang et al., 2026b), have demonstrated strong reasoning capabilities across mathematics, coding, and science. To efficiently transfer and consolidate such capabilities through distillation, on-policy distillation (OPD) (Lu & Lab, 2025) has emerged as a key approach in LLM post-training. It provides dense teacher supervision on trajectories sampled from the current student policy, promoting high-quality reasoning while mitigating exposure bias.

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.

To avoid costly full-vocabulary reverse-KL computation, existing OPD methods (Yang et al., 2026a; Jin et al., 2026; Oh et al., 2026) reformulate the objective in a policy-gradient form and approximate it using a one-sample token-level Monte Carlo estimator (Li et al., 2026). While computationally efficient, this practical estimator introduces non-negligible bias relative to the original sequence-level objective, effectively altering the optimization problem (Yang et al., 2026a). Other methods (Fu et al., 2026) instead adopt sequence-level Monte Carlo estimators to more faithfully preserve the original objective. However, accumulating future credit over the remaining response leads to increasing variance as the sequence grows, resulting in unstable optimization for long reasoning trajectories.

In this work, we present a unified analysis of token-level and sequence-level OPD, deriving their underlying objectives and clarifying the relationship between their policy-gradient formulations. Building on this insight, we propose γOPD, which introduces discounted temporal credit assignment to interpolate between token-local and sequence-level supervision while admitting a horizon-independent variance bound. To complement teacher supervision with outcome-level guidance, we further develop a reward-compatible bounded mixing mechanism that incorporates verifiable outcome feedback without sacrificing the stability benefits of temporal discounting.

Overall, our main contributions are summarized as follows:

- We establish a unified temporal-credit view of OPD, showing that practical token-level OPD can be interpreted as a temporally truncated approximation to the sequence-level reverse-KL gradient.
- Building on this connection, we propose γOPD, a discounted temporal-credit surrogate that interpolates between token-level and sequence-level credit assignment, while admitting a horizon-independent variance bound.
- We further develop Reward-Compatible Bounded Mixing (RBM) for γOPD, which combines discounted teacher-derived credit with verifiable outcome rewards while preventing the teacher signal from dominating task-level supervision.

2 Preliminaries

2.1 Notation

We consider autoregressive reasoning tasks over a discrete vocabulary $\mathcal{V}$, where we aim to optimize a student policy π_θ parameterized by θ under the guidance of a fixed teacher policy π^*. Let $\boldsymbol{x} \sim \mathcal{D}$ denote a prompt sequence sampled from the reasoning dataset. Given $\boldsymbol{x}$, the policy generates a complete response trajectory $\boldsymbol{y} = (y_1, y_2, \ldots, y_T) \in \mathcal{V}^T$, where $T = |\boldsymbol{y}|$ is the number of tokens in the trajectory, and $y_t \in \mathcal{V}$ is the token generated at step t. At decoding step t, the policy conditions on the historical context prefix $\boldsymbol{h}_t := (\boldsymbol{x}, \boldsymbol{y}_{<t})$, which consists of the prompt and all previously generated reasoning tokens. We use $\mathbb{D}_{\mathrm{KL}}(\cdot\|\cdot)$ to denote the KL divergence. For notational simplicity, we define the token-level and sequence-level log-ratios as

$$\Delta_t \triangleq \log \pi_\theta(y_t \mid \boldsymbol{h}_t) - \log \pi^*(y_t \mid \boldsymbol{h}_t), \tag{1a}$$

$$\Delta(\boldsymbol{y}) \triangleq \log \pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) - \log \pi^*(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{t=1}^{T} \Delta_t. \tag{1b}$$

2.2 On-Policy Distillation (OPD)

On-policy distillation (OPD) (Agarwal et al., 2024; DeepSeek-AI, 2026) trains the student by minimizing the reverse KL divergence from the student policy π_θ to the teacher policy π^*, evaluated on trajectories induced by the current student itself:

$$J_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi^*(\boldsymbol{y} \mid \boldsymbol{x})\right)\right]. \tag{2}$$

By training on trajectories sampled from the current student policy, on-policy learning reduces exposure bias and is well suited to long chain-of-thought reasoning tasks such as mathematical problem solving (GLM-5 Team, 2026). Following recent practice (Lu & Lab, 2025; Li et al., 2026), the empirical implementation of OPD typically converts the reverse-KL minimization in equation 2 into an

RL-style surrogate gradient:

$$\nabla_\theta J_{\mathrm{OPD}}(\theta) \approx -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T} A_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid \boldsymbol{h}_t)\right], \tag{3}$$

where $A_t^{\mathrm{OPD}} \triangleq -\Delta_t = \log \pi^*(y_t \mid \boldsymbol{h}_t) - \log \pi_\theta(y_t \mid \boldsymbol{h}_t)$ is the token-level OPD advantage.

2.3 RL WITH VERIFIABLE REWARDS (RLVR)

Reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Guo et al., 2025) has also become a key approach for improving the reasoning ability of LLMs. For a prompt $\boldsymbol{x} \sim \mathcal{D}$, the policy π_θ generates an output sequence $\boldsymbol{y} = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid \boldsymbol{x})$. The generated output is evaluated by an external verifier, such as a code compiler or a mathematical rule checker, which provides a sparse sequence-level reward scalar $R(\boldsymbol{x}, \boldsymbol{y}) \in \{-1, 1\}$. The RLVR objective is

$$J_{\mathrm{RLVR}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[R(\boldsymbol{x}, \boldsymbol{y})\right].$$

Compared with OPD, RLVR provides a sparse but verifier-grounded sequence-level correctness signal, whereas OPD supplies dense token-level supervision through teacher guidance, which may nevertheless be limited by the teacher's suboptimal behavior.

3 METHODOLOGY

3.1 TOKEN-LEVEL OPD AS A TEMPORAL APPROXIMATION

Although the practical OPD objective in equation 3 is widely used, it should be understood as a temporal approximation rather than the exact policy gradient of the OPD objective in equation 2. To clarify this distinction, we formally define token-level OPD and sequence-level OPD as follows:

$$J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi^*(\boldsymbol{y} \mid \boldsymbol{x})\right)\right], \tag{4a}$$

$$J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{h}_t\sim d_{\bar\theta}(\cdot|\boldsymbol{x})}\left[\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\boldsymbol{y}_t \mid \boldsymbol{h}_t) \,\|\, \pi^*(\boldsymbol{y}_t \mid \boldsymbol{h}_t)\right)\right]\right], \tag{4b}$$

where $J_{\mathrm{OPD}}^{\mathrm{seq}}$ denotes the sequence-level reverse-KL objective, i.e., the original OPD objective defined in equation 2, which compares the student and teacher distributions over complete response trajectories. In contrast, $J_{\mathrm{OPD}}^{\mathrm{token}}$ denotes the token-level objective, which compares their next-token distributions at sampled prefixes. The distribution $d_{\bar\theta}(\boldsymbol{h}_t \mid \boldsymbol{x})$ is the prefix distribution induced by the current student policy, where $\boldsymbol{h}_t = (\boldsymbol{x}, \boldsymbol{y}_{<t})$. The notation $\bar\theta$ indicates stop-gradient, namely this prefix distribution is treated as fixed when differentiating with respect to θ.

The two objectives are equivalent at the level of function values when the stop-gradient reference parameter is evaluated at the current policy parameter. Specifically, if $\bar\theta = \theta$, then the sampled prefix distribution in $J_{\mathrm{OPD}}^{\mathrm{token}}$ matches the autoregressive prefix distribution induced by π_θ, and we have

$$J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = J_{\mathrm{OPD}}^{\mathrm{token}}(\theta)\big|_{\bar\theta=\theta}.$$

However, this value-level equivalence does not imply gradient-level equivalence. The reason is that $J_{\mathrm{OPD}}^{\mathrm{seq}}$ differentiates through the autoregressive distribution over future prefixes, whereas $J_{\mathrm{OPD}}^{\mathrm{token}}$ stops this dependence through $d_{\bar\theta}(\boldsymbol{h}_t \mid \boldsymbol{x})$. This distinction leads to different temporal credit assignments, as shown below.

Proposition 3.1 (Token-level OPD gradient)**.** *Under the stop-gradient treatment of the prefix distribution $d_{\bar\theta}(\boldsymbol{h}_t \mid \boldsymbol{x})$, the gradient of $J_{\mathrm{OPD}}^{\mathrm{token}}(\theta)$ in equation 4b is given by*

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T} \Delta_t \nabla_\theta \log \pi_\theta(y_t \mid \boldsymbol{h}_t)\right]. \tag{5}$$

The proof is provided in Appendix A.

Consequently, the gradient of $J_{\mathrm{OPD}}^{\mathrm{token}}(\theta)$ recovers the practical OPD update in equation 3. Meanwhile, the gradient of the sequence-level OPD objective $J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta)$ takes the following form:

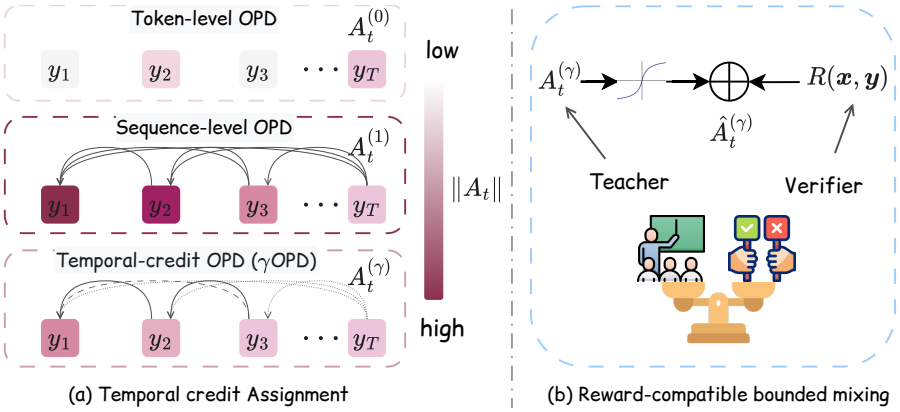


Figure 2: **Overview of γOPD.** (a) Token-level OPD considers only the log-ratio of the current token, whereas sequence-level OPD accumulates the log-ratios of all future tokens. γOPD introduces a discount factor γ to interpolate between these two extremes. (b) The γOPD advantage is first normalized and then combined with the verifiable task reward, balancing dense teacher guidance with task-level verification signals.

Proposition 3.2 (Sequence-level OPD gradient)**.** *The gradient of the sequence-level OPD objective* $J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta)$ *in equation* 4a *is given by*

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\left(\sum_{t'=t}^{T}\Delta_{t'}\right)\nabla_\theta \log \pi_\theta(y_t \mid \boldsymbol{h}_t)\right]. \tag{6}$$

The proof is provided in Appendix B.

Propositions 3.1 and 3.2 show that practical OPD is a token-level approximation to sequence-level OPD that neglects the influence of subsequent tokens. As illustrated in Figure 2, under the sequence-level objective, the score term at step t is weighted by the future log-ratio $\sum_{t'=t}^{T}\Delta_{t'}$, since y_t affects all future histories $\boldsymbol{h}_{t+1}, \ldots, \boldsymbol{h}_T$. In contrast, practical OPD keeps only the local term Δ_t. This gap stems from the stop-gradient treatment of $d_{\bar{\theta}}(\boldsymbol{h}_t \mid \boldsymbol{x})$ in equation 4b.

3.2 Temporal-Credit On-Policy Distillation (γOPD)

Compared with the token-level gradient in Proposition 3.1, the sequence-level gradient in equation 6 accounts for the indirect effect of each token on future autoregressive prefixes, and is therefore unbiased with respect to the sequence-level objective in equation 2. However, for long reasoning trajectories, the accumulated future log-ratio $\sum_{t'=t}^{T}\Delta_{t'}$ can have large variance, which may destabilize training.

To balance bias and variance in temporal credit assignment, we propose *Temporal-Credit On-Policy Distillation (γOPD)*. The key idea is to introduce a discount factor $\gamma \in [0, 1]$ to control how much future OPD credit is assigned to the current token. Specifically, we define the temporal-credit surrogate gradient as

$$g_\gamma(\theta) \triangleq -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T} A_t^{(\gamma)}\nabla_\theta \log \pi_\theta(y_t \mid \boldsymbol{h}_t)\right],$$

where the discounted γOPD advantage is defined as

$$A_t^{(\gamma)} \triangleq -\sum_{t'=t}^{T}\gamma^{t'-t}\Delta_{t'}. \tag{7}$$

The discount factor γ controls the temporal horizon of OPD credit assignment. When $\gamma = 1$, $A_t^{(1)}$ becomes the sequence-level OPD return-to-go, recovering equation 6. When $\gamma = 0$, it reduces to the token-level OPD advantage $A_t^{(0)} = -\Delta_t$, recovering equation 5. Thus, for $0 < \gamma < 1$, $g_\gamma(\theta)$ is

deliberately introduced as a biased surrogate that interpolates between these two endpoint gradients, allowing its variance to be controlled through γ, as formalized in the following theorem.

Theorem 3.3 (Variance Stability of γOPD)**.** *Assume that the token-level OPD advantage has a bounded second moment, i.e.,* $\mathbb{E}[(A_t^{(0)})^2] \le \sigma_\Delta^2$ *for all* t*. Then, the sequence-level OPD credit admits a horizon-dependent variance bound, whereas the* γOPD *credit admits a horizon-independent variance bound:*

$$\mathrm{Var}\Big(A_t^{(1)}\Big) \le (T-t+1)^2\sigma_\Delta^2,$$
$$\mathrm{Var}\Big(A_t^{(\gamma)}\Big) \le \frac{\sigma_\Delta^2}{(1-\gamma)^2}, \qquad \forall \gamma \in [0,1).$$

The proof is provided in Appendix C.

Theorem 3.3 shows that the variance of sequence-level OPD credit can grow quadratically with the remaining sequence horizon in the worst case, whereas γOPD admits a horizon-independent variance bound for any fixed $\gamma < 1$. Adjusting γ therefore provides a principled trade-off between training stability and long-horizon credit propagation.

3.3 γOPD WITH REWARD-COMPATIBLE BOUNDED MIXING (RBM)

Although γOPD improves temporal credit assignment, its optimization signal is still fundamentally derived from the teacher distribution. Consequently, the student may remain constrained by the teacher policy and lack an explicit task-level optimization signal for solving verifiable reasoning problems.

To move beyond this teacher-imitation ceiling, we incorporate the verifiable task reward $R(\boldsymbol{x}, \boldsymbol{y}) \in \{-1, 1\}$ into γOPD. A naive combination, however, can be problematic because the magnitude of the teacher-derived advantage $A_t^{(\gamma)}$ may vary substantially across responses and potentially dominate the bounded task reward. We therefore introduce a *reward-compatible bounded mixing (RBM)* mechanism: the γOPD credit is first calibrated by its response-wise mean absolute magnitude, then bounded through a softsign transformation, and finally combined with the verifiable reward.

These operations can be written compactly as

$$\widehat{A}_t^{(\gamma)} \triangleq \frac{A_t^{(\gamma)}}{\frac{1}{T}\sum_{k=1}^{T}\left|A_k^{(\gamma)}\right| + \left|A_t^{(\gamma)}\right|} + R(\boldsymbol{x}, \boldsymbol{y}). \tag{8}$$

When the denominator vanishes, equivalently when $A_k^{(\gamma)} = 0$ for all k, we define the normalized teacher term as zero. The first term is equivalent to applying softsign after response-wise mean-absolute normalization. It is monotonic in $A_t^{(\gamma)}$ and bounded within $(-1, 1)$, thereby retaining the relative token-level OPD credit while preventing the teacher-derived signal from overwhelming the task reward. Since $R(\boldsymbol{x}, \boldsymbol{y}) \in \{-1, 1\}$, it follows that $\mathrm{sign}(\widehat{A}_t^{(\gamma)}) = R(\boldsymbol{x}, \boldsymbol{y})$. Thus, RBM is reward-compatible by construction: the verifiable reward determines whether the sampled response is reinforced or suppressed, while the bounded γOPD credit modulates the token-level update strength. The resulting reward-compatible γOPD surrogate gradient is

$$\hat{g}_\gamma(\theta) \triangleq -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\widehat{A}_t^{(\gamma)}\nabla_\theta \log \pi_\theta(y_t \mid \boldsymbol{h}_t)\right]. \tag{9}$$

Therefore, RBM combines sparse verifiable rewards for task-level learning beyond teacher imitation with bounded γOPD for dense token-level credit assignment. The complete reward-enhanced training procedure is summarized in Algorithm 1. Unless otherwise specified, γOPD refers to its use within RBM throughout the remainder of this paper.

Table 1: **Mathematical reasoning results under vanilla and size-mismatch distillation.** We report accuracy and pass rate across four mathematical reasoning benchmarks.

Method	AIME24		AIME25		AMC23		MATH500		Average	
	Acc.	Pass	Acc.	Pass	Acc.	Pass	Acc.	Pass	Acc.	Pass
Qwen3-4B-Math → Qwen3-4B										
Student	22.60	60.00	20.83	33.33	60.16	92.50	67.40	72.60	42.75	64.61
Teacher	57.40	76.67	51.67	66.67	93.52	97.50	69.80	78.40	68.10	79.81
OPD	56.46	80.00	50.83	63.33	92.81	95.00	68.05	72.20	67.04	77.63
ExOPD	57.19	83.33	52.50	63.33	93.13	**97.50**	68.90	74.20	67.93	79.59
REOPOLD	56.98	83.33	51.67	63.33	**93.59**	**97.50**	68.70	72.60	67.74	79.19
AOPD	56.56	86.67	52.50	66.67	93.28	97.50	69.35	73.00	67.92	80.96
TOPD	54.79	80.00	51.67	63.33	93.44	97.50	69.95	72.40	67.46	78.31
γOPD	**60.94**	**86.67**	**54.17**	**66.67**	92.89	**97.50**	**69.95**	**78.80**	**69.49**	**82.41**
Qwen3-4B-Math → Qwen3-1.7B										
Student	11.77	46.67	8.33	20.00	39.84	90.00	59.20	72.80	29.79	57.37
Teacher	57.40	76.67	51.67	66.67	93.52	97.50	69.80	78.40	68.10	79.81
OPD	37.71	66.67	31.25	40.00	76.02	92.50	65.40	69.80	52.60	67.24
ExOPD	38.54	66.67	31.67	43.33	76.72	92.50	65.05	68.40	53.00	67.73
REOPOLD	37.50	66.67	30.83	43.33	77.73	92.50	66.70	69.00	53.19	67.88
AOPD	37.71	70.00	32.50	43.33	79.38	95.00	65.85	70.80	53.86	69.78
TOPD	39.38	70.00	30.83	36.67	79.14	95.00	65.60	70.00	53.74	67.92
γOPD	**42.08**	**70.00**	**35.00**	**46.67**	**80.86**	**95.00**	**68.10**	**73.20**	**56.51**	**71.22**

4 EXPERIMENTS

4.1 SETTINGS

Benchmarks. Our experiments evaluate both mathematical and code reasoning abilities. For mathematical reasoning, we train on DeepMath-103K (He et al., 2026b), retaining problems with difficulty level at least 6, and evaluate on AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), AMC23 (Math-AI, 2023), and MATH500 (Hendrycks et al., 2021; Lightman et al., 2024). For code reasoning, we train on the 25K-sample Eurus-RL-Code dataset (Cui et al., 2026) and evaluate on HumanEval+, MBPP+ (Liu et al., 2023), and the v6 split of LiveCodeBench (Jain et al., 2025), covering problems from February 2025 to May 2025. Detailed training and evaluation configurations are provided in Appendix E.

Models. We conduct experiments under three distillation settings: (1) *vanilla distillation*, where we distill a Qwen3-4B-Math (Yang et al., 2026a) teacher into a Qwen3-4B (Yang et al., 2025) student for mathematical reasoning; (2) *size-mismatch distillation*, where we distill the same Qwen3-4B-Math teacher into a smaller Qwen3-1.7B student for mathematical reasoning; and (3) *multi-teacher distillation*, where we distill Qwen3-4B-Math and Qwen3-4B-Code (Yang et al., 2026a) teachers into a single Qwen3-4B student for both math and code reasoning.

Baselines. We compare our proposed γOPD with several baselines, including vanilla OPD (Lu & Lab, 2025), ExOPD (Yang et al., 2026a), REOPOLD (Ko et al., 2026), AOPD (Jia et al., 2026), and TOPD (Zhang et al., 2026). For γOPD, we use $\gamma = 0.99$ by default unless otherwise specified. All methods are implemented using the `veRL` framework (Sheng et al., 2025).

4.2 MAIN RESULTS

Vanilla Distillation. As shown in Table 1, γOPD consistently outperforms existing OPD baselines in the vanilla distillation setting. Compared with the strongest baseline for each metric, γOPD achieves relative improvements of 2.30% in AvgAcc and 1.80% in AvgPass. Notably, it also surpasses the teacher on both averaged metrics. These results demonstrate the effectiveness of γOPD

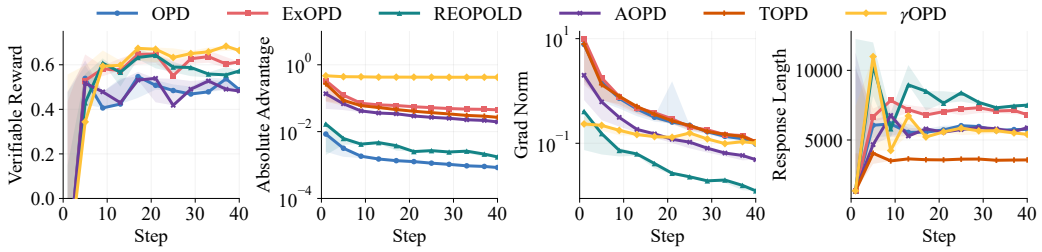


Figure 3: **Training dynamics under vanilla distillation.** Verifiable reward, absolute advantage, gradient norm, and response length throughout training are reported. Due to the truncation mechanism, TOPD's verifiable reward remains mostly below zero throughout training.

Table 2: **Accuracy on mathematical and code reasoning benchmarks under multi-teacher distillation.** TotalAvg is the equal-weighted average of the mean mathematical score and the mean code score.

Method	Mathematical Reasoning				Code Reasoning			TotalAvg
	AIME24	AIME25	AMC23	MATH500	HumanEval+	MBPP+	LCB	
Student	22.60	20.83	60.16	67.40	75.61	61.90	17.14	47.15
Teacher	57.40	51.67	93.52	69.80	82.32	70.63	25.71	63.82
OPD	55.21	52.50	93.13	68.65	84.15	67.99	26.00	63.37
ExOPD	57.50	52.50	92.34	68.25	85.37	**71.16**	26.71	64.36
REOPOLD	58.13	52.08	91.88	68.55	82.32	69.31	28.00	63.77
AOPD	55.52	**55.83**	93.05	68.70	84.76	69.05	26.71	64.22
TOPD	57.81	51.67	92.34	**69.50**	84.15	69.58	27.86	64.18
γOPD	**59.27**	55.00	**94.38**	69.00	**89.02**	70.63	**28.14**	**66.00**

and suggest that the proposed RBM can better leverage dense teacher supervision while enabling improvement beyond direct imitation.

Size-Mismatch Distillation. The size-mismatch setting (Qwen3-4B-Math → Qwen3-1.7B) is more challenging because the student has substantially lower capacity than the teacher. Consequently, as shown in Table 1, all student methods remain below the teacher. Nevertheless, γOPD substantially narrows the performance gap and achieves the best overall performance. Compared with the strongest baseline for each metric, γOPD achieves relative improvements of 4.92% in AvgAcc and 2.06% in AvgPass. These results highlight the benefit of temporal credit assignment when distilling a stronger teacher into a lower-capacity student. For completeness, we further compare with representative RLVR methods under the same experimental setting. As reported in Appendix F.2, γOPD also maintains a clear advantage in both the 4B and 1.7B student settings.

Multi-Teacher Distillation. We next evaluate whether γOPD remains effective in the multi-teacher distillation setting. As shown in Table 2, it consistently performs strongly across mathematical reasoning benchmarks and achieves the best results on HumanEval+ and LCB among the code reasoning benchmarks. Overall, γOPD achieves a relative improvement of 2.55% in TotalAvg over the strongest baseline, ExOPD. Notably, it also outperforms the corresponding domain-specific teachers on average in both mathematical and code reasoning. These results demonstrate that γOPD remains effective when jointly training a shared student with domain-specialized teachers.

Training Dynamics. We further visualize the training dynamics of different methods under the vanilla distillation setting in Figure 3. Throughout training, γOPD consistently achieves the highest verifiable reward, indicating the largest proportion of correctly solved problems. It also maintains a stable absolute OPD advantage and exhibits the smallest fluctuations in gradient norm among all methods, suggesting more consistent temporal credit assignment and optimization. Moreover, the response length of γOPD converges to a shorter and more stable range than those of most baselines, with the exception of TOPD, which explicitly truncates the distillation signal. Together, these results suggest that γOPD enables more stable optimization while achieving stronger mathematical reasoning performance.

Table 3: **Ablation study of the proposed components on AIME benchmarks.** γ, Mix, and Norm correspond to temporal discounting, reward mixing, and bounded normalization, respectively. Δ Avg. denotes the absolute improvement in average accuracy over the vanilla OPD baseline.

Components			Accuracy (%)			Δ Avg.
γ	Mix	Norm	AIME24	AIME25	Avg.	
Vanilla OPD			56.46	50.83	53.65	–
✓	–	–	58.29	53.08	55.69	+2.04
✓	✓	–	<u>58.81</u>	52.67	<u>55.74</u>	+2.09
–	✓	✓	57.55	<u>53.75</u>	55.65	+2.00
✓	✓	✓	**60.94**	**54.17**	**57.56**	**+3.91**

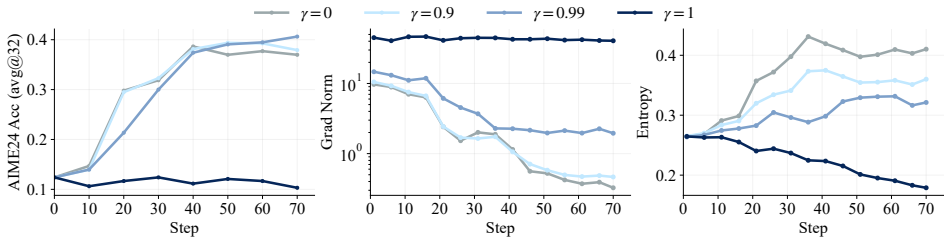


Figure 4: **Training dynamics under different γ.** OPD distills a Qwen3-4B-Math teacher into a Qwen3-1.7B student, with validation accuracy, gradient norm, and policy entropy reported.

4.3 Ablation Studies

We isolate the effects of temporal discounting, reward mixing, and bounded normalization on the AIME benchmarks in Table 3. Temporal discounting alone improves the average accuracy by 2.04 points over vanilla OPD. Removing temporal discounting from the full method while retaining reward mixing and normalization reduces the gain from 3.91 to 2.00 points, confirming its complementary contribution. Moreover, adding naive reward mixing to temporal discounting brings only a marginal 0.05-point improvement, whereas the full combination including bounded normalization yields the strongest overall performance. These results suggest that temporal credit assignment and reward-compatible normalization provide complementary benefits, with their combination yielding the strongest performance. We further analyze the computational and memory overhead of the individual components of γOPD. Compared with standard OPD, γOPD introduces only about 0.1% additional computation time with negligible memory overhead; detailed profiling results are provided in Appendix F.1.

4.4 Detailed Analysis

Sensitivity to γ. We further investigate the sensitivity of temporal credit assignment to the discount factor γ without RBM. Specifically, we distill a Qwen3-4B-Math teacher into a Qwen3-1.7B student using different values of γ, with the results shown in Figure 4. Here, $\gamma = 0$ reduces to local OPD, whereas $\gamma = 1$ corresponds to the undiscounted sequence-level return-to-go. Due to the long response horizon, the sequence-level variant produces substantially larger gradient norms and suffers from rapid entropy collapse, resulting in inferior reasoning performance. In contrast, local OPD and the discounted variants steadily improve during training. Among them, $\gamma = 0.99$ achieves the best performance, outperforming smaller values such as $\gamma = 0.9$, while exhibiting distinct gradient-norm and entropy dynamics. These results suggest that a large but sub-unity discount factor provides the best balance between long-range temporal credit assignment and optimization stability.

Visualization of Token Advantages. We further visualize the normalized token-level advantages $A_t^{(0)}$ and $A_t^{(0.99)}$ in Figure 5. In the correct response, $A_t^{(0)}$ provides sparse local supervision, assigning a noticeable negative signal to only a few tokens while exerting little influence on most of the reasoning process. In contrast, $A_t^{(0.99)}$ propagates information from subsequent steps, producing a smoother signal and assigning stronger positive credit to the key mathematical derivation while mildly penalizing redundant text. For the incorrect response, $A_t^{(0)}$ strongly penalizes several tokens

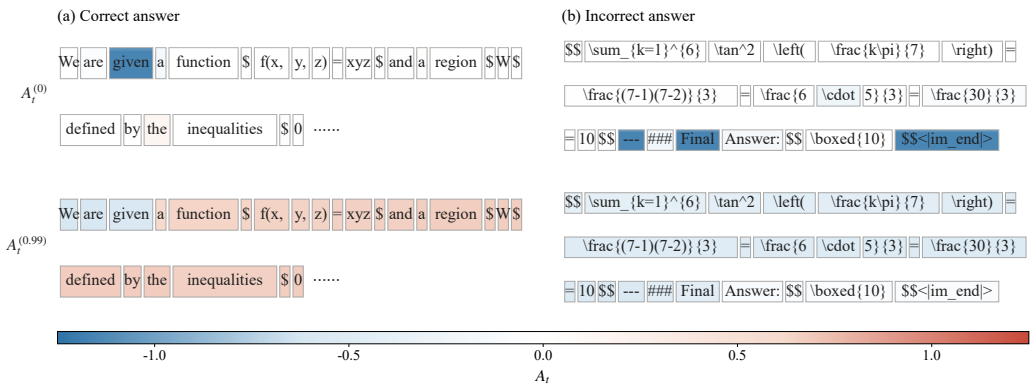


Figure 5: **Token-level advantage visualization.** (a) Correct response; (b) incorrect response. The top row shows normalized $A_t^{(0)}$, and the bottom row shows normalized $A_t^{(0.99)}$. Blue and red denote negative and positive advantages, respectively, with color intensity indicating magnitude. Complete response examples are provided in Appendix F.3.

that are only weakly related to the actual reasoning error, whereas $A_t^{(\gamma)}$ concentrates stronger negative credit on the erroneous formula derivation without excessively penalizing the final token. These examples illustrate that temporal credit assignment can redistribute supervision toward reasoning steps that are more relevant to the final outcome, yielding smoother and more semantically aligned token-level signals. Complete response-level visualizations are provided in Appendix F.3.

5 Related Work

Credit Assignment for OPD. Recent studies have explored credit assignment in OPD, motivated by the potentially unreliable reasoning trajectories generated by relatively weak student policies. Truncation-based methods (Zhou et al., 2026; Zhang et al., 2026) terminate rollouts early or mask the learning signals of tokens beyond selected positions, thereby reducing the influence of unreliable continuations. Entropy-aware OPD (Jin et al., 2026) uses token-level uncertainty to modulate the direction or magnitude of optimization signals, while ExOPD (Yang et al., 2026a) introduces a reference model to calibrate the OPD advantage. Fu et al. (2026) further distinguish token-level from sequence-level OPD and analyze their credit-assignment behaviors, yet a principled balance between sequence-level objective fidelity and token-level optimization stability remains unresolved.

Reward-Guided OPD. Recent studies incorporate verifiable rewards into OPD to complement teacher-derived supervision. REOPOLD (Ko et al., 2026) combines reward clipping, entropy-based sampling, and exploration-to-refinement scheduling, while SCOPE (Zheng et al., 2026) routes correct and incorrect trajectories to different OPD or KL-correction objectives. AOPD (Jia et al., 2026) derives trajectory-level signals from the signs of token-level advantages, whereas RWOPD (Zou et al., 2026) weights teacher KL gradients using verifier rewards. Other works (Ding et al., 2026) further stabilize optimization through advantage compression and warmup-then-anneal scheduling. Nevertheless, these methods often introduce additional complexity through group-based advantage estimation, log-ratio correction, or verifier-weighted teacher KL objectives, potentially incurring substantial computational overhead.

6 Conclusion

In this work, we presented a unified view of token-level and sequence-level OPD through the lens of temporal credit assignment and proposed γOPD to balance long-horizon supervision and optimization stability. We further introduced RBM to combine discounted teacher guidance with verifiable outcome rewards. Empirical results on mathematical and code reasoning demonstrate that γOPD remains effective across different teacher–student configurations and distillation scenarios.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and readability editing, debugging and minor editing of software code, and creating or modifying schematic figures. We did not use generative AI for any required disclosure tasks, including synthetic data generation, theoretical or conceptual development, mathematical claims or proofs, hypothesis development, methodology or experiment design, method implementation, translation, dataset processing, qualitative analysis, or result interpretation. No other required disclosure tasks apply.

All AI-assisted text, code, and figures were manually reviewed. The code was tested in the experimental pipeline, and the figures were checked against the corresponding methods and results. The authors determined and verified the research ideas, methods, experiments, analysis, and conclusions, and take responsibility for the final content of 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 *The Twelfth International Conference on Learning Representations*, 2024.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=9SkkifLopZ.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Yifan Ding, Xincheng Wei, Yoshua Y. Li, Ziheng Li, Yuquan Lu, Siyu Zhang, Dongsheng Ma, Rongxiang Weng, Xunliang Cai, and Yun Chen. Saf-opd: Stable advantage fusion for on-policy distillation, 2026.

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

GLM-5 Team. Glm-5: From vibe coding to agentic engineering, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You,

Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, Ning Ding, and Zhiyuan Liu. Justrl: Scaling a 1.5b llm with a simple rl recipe. ICLR 2026 Blog Track, 2026a. URL https://iclr.cc/virtual/2026/events/BlogTrack-2026.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/pdf/d9f7507b92be053067e42caf97b8b9d7c39d347e.pdf.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 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. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems 36*, 2023.

Shiqi Liu, Zeyu He, Guojian Zhan, Letian Tao, Zhilong Zheng, Jiang Wu, Yinuo Wang, Yang Guan, Kehua Sheng, Bo Zhang, Keqiang Li, Jingliang Duan, and Shengbo Eben Li. Stapo: Stabilizing reinforcement learning for llms by silencing rare spurious tokens, 2026. URL https://arxiv.org/abs/2602.15620.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab blog, 2025.

Math-AI. Amc23: American mathematics competitions 2023. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/math-ai/amc23.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. Kl for a kl: On-policy distillation with control variate baseline, 2026.

OpenAI. Introducing gpt-5. OpenAI, August 2025. URL https://openai.com/index/introducing-gpt-5/.

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

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 20th European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

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, 2025. URL https://arxiv.org/abs/2505.09388.

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

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation, 2026b. URL https://arxiv.org/abs/2603.19220.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems 38*, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Yaocheng Zhang, Jiajun Chai, Yuqian Fu, Songjun Tu, Xiaohan Wang, Wei Lin, Guojun Yin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Are full rollouts necessary for on-policy distillation?, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/math-ai/aime24.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

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

Ziheng Zhou, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation, 2026.

Qingyun Zou, Yingze Li, Tianen Liu, Bingsheng He, and Weng-Fai Wong. Reward-weighted on-policy distillation with an open property-equivalence verifier for nl-to-sva generation, 2026.

APPENDIX

A PROOF OF PROPOSITION 3.1

Starting from the token-level OPD objective in equation 4b, we have

$$J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{h}_t\sim d_{\bar{\theta}}(\cdot|\boldsymbol{x})}\left[\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\cdot\mid\boldsymbol{h}_t)\,\|\,\pi^*(\cdot\mid\boldsymbol{h}_t)\right)\right]\right].$$

Since the prefix distribution $d_{\bar{\theta}}(\boldsymbol{h}_t \mid \boldsymbol{x})$ is treated with stop-gradient, it is fixed when differentiating with respect to θ. Therefore, the gradient can be moved inside the expectation over prefixes:

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{h}_t\sim d_{\bar{\theta}}(\cdot|\boldsymbol{x})}\left[\nabla_\theta\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)\left(\log\pi_\theta(a\mid\boldsymbol{h}_t)-\log\pi^*(a\mid\boldsymbol{h}_t)\right)\right]\right]. \tag{10}$$

For a fixed prefix $\boldsymbol{h}_t$, define

$$\Delta(a;\boldsymbol{h}_t) \triangleq \log\pi_\theta(a\mid\boldsymbol{h}_t) - \log\pi^*(a\mid\boldsymbol{h}_t).$$

This is the full-vocabulary counterpart of the sampled token-level log-ratio Δ_t in equation 1a. Then the gradient of the inner next-token KL term is

$$\begin{aligned}&\nabla_\theta\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)\Delta(a;\boldsymbol{h}_t)\\ &=\sum_{a\in\mathcal{V}}\nabla_\theta\pi_\theta(a\mid\boldsymbol{h}_t)\Delta(a;\boldsymbol{h}_t)+\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)\nabla_\theta\log\pi_\theta(a\mid\boldsymbol{h}_t).\end{aligned} \tag{11}$$

The second term in equation 11 vanishes because

$$\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)\nabla_\theta\log\pi_\theta(a\mid\boldsymbol{h}_t)=\sum_{a\in\mathcal{V}}\nabla_\theta\pi_\theta(a\mid\boldsymbol{h}_t)=\nabla_\theta\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)=0.$$

Using the score-function identity

$$\nabla_\theta\pi_\theta(a\mid\boldsymbol{h}_t)=\pi_\theta(a\mid\boldsymbol{h}_t)\nabla_\theta\log\pi_\theta(a\mid\boldsymbol{h}_t),$$

we obtain

$$\begin{aligned}\nabla_\theta\mathbb{D}_{\mathrm{KL}}\left(\pi_\theta(\cdot\mid\boldsymbol{h}_t)\,\|\,\pi^*(\cdot\mid\boldsymbol{h}_t)\right)&=\sum_{a\in\mathcal{V}}\pi_\theta(a\mid\boldsymbol{h}_t)\Delta(a;\boldsymbol{h}_t)\nabla_\theta\log\pi_\theta(a\mid\boldsymbol{h}_t)\\ &=\mathbb{E}_{y_t\sim\pi_\theta(\cdot|\boldsymbol{h}_t)}\left[\Delta_t\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right],\end{aligned} \tag{12}$$

where $\Delta_t=\Delta(y_t;\boldsymbol{h}_t)$ follows from equation 1a.

Substituting equation 12 into equation 10 gives

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{h}_t\sim d_{\bar{\theta}}(\cdot|\boldsymbol{x}),\,y_t\sim\pi_\theta(\cdot|\boldsymbol{h}_t)}\left[\Delta_t\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right]\right]. \tag{13}$$

When the stop-gradient prefix distribution is evaluated at the current policy, the prefix $\boldsymbol{h}_t$ together with the next token y_t can be generated by an on-policy trajectory $\boldsymbol{y}\sim\pi_\theta(\cdot\mid\boldsymbol{x})$, while gradients are not propagated through the prefix-sampling distribution. Hence, equation 13 can be written as

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{token}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\,\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\Delta_t\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right],$$

which is exactly the token-level OPD gradient in equation 5. This proves Proposition 3.1.

B Proof of Proposition 3.2

Starting from the sequence-level OPD objective in equation 4a, we expand the reverse KL over complete response trajectories:

$$J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{\boldsymbol{y}\in\mathcal{V}^T} \pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\Delta(\boldsymbol{y})\right],$$

where $\Delta(\boldsymbol{y})$ is the sequence-level log-ratio defined in equation 1b. Taking the gradient with respect to θ gives

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\sum_{\boldsymbol{y}\in\mathcal{V}^T} \nabla_\theta\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\Delta(\boldsymbol{y}) + \sum_{\boldsymbol{y}\in\mathcal{V}^T} \pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\nabla_\theta\log\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\right]. \tag{14}$$

The second term in equation 14 vanishes because

$$\sum_{\boldsymbol{y}\in\mathcal{V}^T} \pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\nabla_\theta\log\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x}) = \nabla_\theta \sum_{\boldsymbol{y}\in\mathcal{V}^T} \pi_\theta(\boldsymbol{y}\mid\boldsymbol{x}) = \nabla_\theta 1 = 0.$$

Using the score-function identity,

$$\nabla_\theta\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x}) = \pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\nabla_\theta\log\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x}),$$

we obtain

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\,\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\Delta(\boldsymbol{y})\nabla_\theta\log\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x})\right]. \tag{15}$$

By the autoregressive factorization and the log-ratio decomposition in equation 1, we have

$$\Delta(\boldsymbol{y}) = \sum_{t'=1}^{T}\Delta_{t'}, \qquad \nabla_\theta\log\pi_\theta(\boldsymbol{y}\mid\boldsymbol{x}) = \sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t). \tag{16}$$

Substituting equation 16 into equation 15 yields

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\,\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\sum_{t'=1}^{T}\Delta_{t'}\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right]. \tag{17}$$

It remains to remove the terms with $t' < t$. For $t' < t$, $\Delta_{t'}$ is determined by earlier tokens and is therefore measurable with respect to $\boldsymbol{h}_t$. Hence, conditioning on $\boldsymbol{h}_t$,

$$\begin{aligned}&\mathbb{E}_{y_t\sim\pi_\theta(\cdot|\boldsymbol{h}_t)}\left[\Delta_{t'}\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right]\\ &\quad= \Delta_{t'}\mathbb{E}_{y_t\sim\pi_\theta(\cdot|\boldsymbol{h}_t)}\left[\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right] = 0,\end{aligned}$$

where the last equality follows from the token-level score identity

$$\mathbb{E}_{y_t\sim\pi_\theta(\cdot|\boldsymbol{h}_t)}\left[\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right] = \nabla_\theta\sum_{y_t\in\mathcal{V}}\pi_\theta(y_t\mid\boldsymbol{h}_t) = 0.$$

Therefore, all terms with $t' < t$ in equation 17 vanish in expectation. Keeping only the terms with $t' \geq t$, we obtain

$$\nabla_\theta J_{\mathrm{OPD}}^{\mathrm{seq}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\,\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\left(\sum_{t'=t}^{T}\Delta_{t'}\right)\nabla_\theta\log\pi_\theta(y_t\mid\boldsymbol{h}_t)\right],$$

which is exactly the sequence-level gradient in equation 6. This proves Proposition 3.2.

C Proof of Theorem 3.3

Proof. Let $N_t = T - t + 1$ and define

$$\widetilde{A}_\tau^{(0)} = A_\tau^{(0)} - \mathbb{E}\Big[A_\tau^{(0)}\Big].$$

Since

$$\left\|\widetilde{A}_\tau^{(0)}\right\|_2^2 = \mathrm{Var}\Big(A_\tau^{(0)}\Big) \le \mathbb{E}\left[\left(A_\tau^{(0)}\right)^2\right] \le \sigma_\Delta^2,$$

we have $\left\|\widetilde{A}_\tau^{(0)}\right\|_2 \le \sigma_\Delta$.

For the sequence-level OPD credit, the triangle inequality in L_2 gives

$$\begin{aligned}\sqrt{\mathrm{Var}\Big(A_t^{(1)}\Big)} &= \left\|\sum_{\tau=t}^{T} \widetilde{A}_\tau^{(0)}\right\|_2 \\ &\le \sum_{\tau=t}^{T} \left\|\widetilde{A}_\tau^{(0)}\right\|_2 \le N_t \sigma_\Delta.\end{aligned}$$

Therefore,

$$\mathrm{Var}\Big(A_t^{(1)}\Big) \le N_t^2 \sigma_\Delta^2 = (T-t+1)^2 \sigma_\Delta^2.$$

This quadratic dependence is attainable in the worst case. In particular, if $A_\tau^{(0)} = Z$ for all $\tau \in \{t, \dots, T\}$, where $\mathbb{E}[Z] = 0$ and $\mathrm{Var}(Z) = \sigma_\Delta^2$, then

$$\mathrm{Var}\Big(A_t^{(1)}\Big) = \mathrm{Var}(N_t Z) = N_t^2 \sigma_\Delta^2.$$

For the γOPD credit, similarly,

$$\begin{aligned}\sqrt{\mathrm{Var}\Big(A_t^{(\gamma)}\Big)} &= \left\|\sum_{\tau=t}^{T} \gamma^{\tau-t} \widetilde{A}_\tau^{(0)}\right\|_2 \\ &\le \sigma_\Delta \sum_{\tau=t}^{T} \gamma^{\tau-t} \\ &= \sigma_\Delta \frac{1-\gamma^{N_t}}{1-\gamma} \le \frac{\sigma_\Delta}{1-\gamma}, \qquad \forall \gamma \in [0,1).\end{aligned}$$

Squaring both sides yields

$$\mathrm{Var}\Big(A_t^{(\gamma)}\Big) \le \frac{\sigma_\Delta^2}{(1-\gamma)^2}, \qquad \forall \gamma \in [0,1).$$

Thus, sequence-level OPD can exhibit variance that grows quadratically with the remaining horizon, whereas γOPD admits a horizon-independent variance bound for every fixed $\gamma < 1$. □

D Algorithm

The complete reward-enhanced γOPD procedure is summarized in Algorithm 1.

Table 4: Training hyperparameters for the OPD-based methods. The RLVR baselines use a rollout group size of 8.

Hyperparameter	Value
Training framework	`veRL`
Rollout backend	vLLM
Math reward function	DAPO boxed verifier
Code reward function	Execution-based verifier
Train batch size	1024
Responses per prompt	1
PPO mini batch size	1024
PPO micro batch size/GPU	1
Max prompt length	2048
Max response length	16384
Optimizer	AdamW
Weight decay	0.01
Learning rate	1×10^{-5}
Training temperature / top-p	1.0 / 1.0
Validation temperature / top-p	1.0 / 1.0
Rollout tensor parallel size	4

Algorithm 1 Reward-Enhanced Temporal-Credit On-Policy Distillation (γOPD)

Require: Dataset $\mathcal{D}$, student policy π_θ, teacher policy π^*, discount factor γ, batch size B

for each training iteration **do**
  Sample prompts $\{\boldsymbol{x}_i\}_{i=1}^{B} \sim \mathcal{D}$ and responses $\{\boldsymbol{y}_i\}_{i=1}^{B} \sim \pi_\theta(\cdot \mid \boldsymbol{x}_i)$
  Evaluate the verifier to obtain $\{R_i\}_{i=1}^{B}$, where $R_i \leftarrow R(\boldsymbol{x}_i, \boldsymbol{y}_i)$
  for each mini-batch **do**
    for each response $(\boldsymbol{x}, \boldsymbol{y}, R)$ in the mini-batch **do**
      for $t = 1, \ldots, T$ **do**
        $\Delta_t \leftarrow \log \pi_\theta(y_t \mid \boldsymbol{h}_t) - \log \pi^*(y_t \mid \boldsymbol{h}_t)$
      end for
      Compute $\{\hat{A}_t^{(\gamma)}\}_{t=1}^{T}$ according to equation 7 and equation 8
    end for
    Update θ using equation 9
  end for
end for
return final student policy π_θ

E Experiment Details

All methods, including γOPD, are implemented with `veRL` (Sheng et al., 2025), using vLLM (Kwon et al., 2023) for rollouts and FSDP for actor training. For mathematical reasoning, we train on DeepMath-103K (He et al., 2026b), retaining examples with difficulty level at least 6. Each prompt is formatted in chat style and appended with "Please output the final answer within \boxed{}." For code reasoning, we train on Eurus-RL-Code (Cui et al., 2026). For both domains, verifier outcomes are mapped to binary rewards in $\{+1, -1\}$. Math responses are evaluated with a DAPO-style boxed-answer verifier (Yu et al., 2025), while code responses are verified through execution-based unit tests. In the multi-teacher setting, each sample is routed to its domain-specific teacher and verifier while sharing the same student policy.

OPD uses reverse-KL token advantages without an additional KL reward penalty. Unless otherwise specified, all models are trained with AdamW and a constant learning rate on 32 NVIDIA H20 GPUs, with each training run taking approximately 1–3 days; full hyperparameters are provided in Table 4.

Table 5: Computational and memory overhead of the additional γOPD operations.

Operation	**Time (ms/step)**	**Memory (MB)**
Discounted temporal credit	609.85	0.063
Bounded advantage shaping	0.42	0.125
Verifier-reward mixing	0.04	0.188
Total additional overhead	**610.32**	**0.375**
Vanilla OPD update (reference)	573,669.00	81,254.40
Relative overhead	**0.1064%**	**0.00046%**

Table 6: **Complete mathematical reasoning comparison with RL baselines.** We report accuracy and pass rate on four mathematical reasoning benchmarks under vanilla and size-mismatch distillation. JustRL and STAPO are pure RLVR baselines without teacher log-probabilities.

Method	AIME24		AIME25		AMC23		MATH500		Average	
	Acc.	Pass	Acc.	Pass	Acc.	Pass	Acc.	Pass	Acc.	Pass
Qwen3-4B-Math → Qwen3-4B										
Student	22.60	60.00	20.83	33.33	60.16	92.50	67.40	72.60	42.75	64.61
Teacher	57.40	76.67	51.67	66.67	93.52	97.50	69.80	78.40	68.10	79.81
JustRL	53.54	83.33	42.50	56.67	88.75	97.50	68.00	70.40	63.20	76.97
STAPO	55.31	83.33	50.00	66.67	90.16	97.50	68.35	69.80	65.95	79.33
OPD	56.46	80.00	50.83	63.33	92.81	95.00	68.05	72.20	67.04	77.63
ExOPD	57.19	83.33	52.50	63.33	93.13	**97.50**	68.90	74.20	67.93	79.59
REOPOLD	56.98	83.33	51.67	63.33	**93.59**	**97.50**	68.70	72.60	67.74	79.19
AOPD	56.56	86.67	52.50	66.67	93.28	97.50	69.35	73.00	67.92	80.96
TOPD	54.79	80.00	51.67	63.33	93.44	97.50	69.95	72.40	67.46	78.31
γOPD	**60.94**	**86.67**	**54.17**	**66.67**	92.89	**97.50**	**69.95**	**78.80**	**69.49**	**82.41**
Qwen3-4B-Math → Qwen3-1.7B										
Student	11.77	46.67	8.33	20.00	39.84	90.00	59.20	72.80	29.79	57.37
Teacher	57.40	76.67	51.67	66.67	93.52	97.50	69.80	78.40	68.10	79.81
JustRL	37.60	66.67	36.67	50.00	78.67	95.00	65.10	68.80	54.51	70.12
STAPO	35.52	63.33	30.83	46.67	77.73	95.00	64.95	68.80	52.26	68.45
OPD	37.71	66.67	31.25	40.00	76.02	92.50	65.40	69.80	52.60	67.24
ExOPD	38.54	66.67	31.67	43.33	76.72	92.50	65.05	68.40	53.00	67.73
REOPOLD	37.50	66.67	30.83	43.33	77.73	92.50	66.70	69.00	53.19	67.88
AOPD	37.71	70.00	32.50	43.33	79.38	95.00	65.85	70.80	53.86	69.78
TOPD	39.38	70.00	30.83	36.67	79.14	95.00	65.60	70.00	53.74	67.92
γOPD	**42.08**	**70.00**	**35.00**	**46.67**	**80.86**	**95.00**	**68.10**	**73.20**	**56.51**	**71.22**

We use temperature 1.0 and top-p 1.0 for all evaluations. For mathematical reasoning, we evaluate on AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), AMC23 (Math-AI, 2023), and MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), with a maximum prompt length of 2048 and response length of 16384. Following the repeated-sampling convention, AIME24, AIME25, and AMC23 are evaluated with 32 samples per problem, while MATH500 uses 4 samples per problem. Accuracy denotes sample-level correctness averaged over all generated responses. Pass rate denotes the fraction of problems for which at least one sampled response is correct; hence, Pass corresponds to Pass@32 for AIME24, AIME25, and AMC23 and Pass@4 for MATH500. All predictions are scored using the same boxed-answer verifier as in training. For code reasoning,

we evaluate HumanEval+ and MBPP+ (Liu et al., 2023) with EvalPlus, and the v6 split of LiveCodeBench (Jain et al., 2025). HumanEval+ and MBPP+ use one sample per problem, whereas LiveCodeBench uses four samples with a maximum generation length of 16384 tokens.

F Supplementary Experimental Results

F.1 Computational Time and Memory Analysis

We profile the additional computation introduced by γOPD using the same 4B student/teacher models and 32-GPU setup as the main experiments. Statistics are averaged over training steps 2–100 after warm-up. The extra operations include discounted temporal-credit computation, bounded advantage shaping, and verifier-reward mixing. Overall, γOPD increases computation time by only 0.1064% and memory usage by 0.00046% relative to a full OPD update, indicating negligible training overhead.

F.2 Additional RL Baseline Results

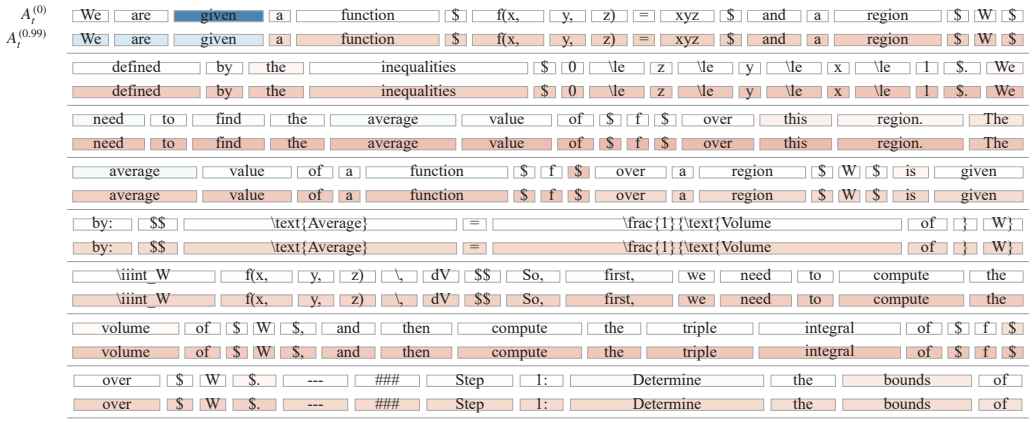
We additionally compare against JustRL (He et al., 2026a) and STAPO (Liu et al., 2026), two RLVR baselines that optimize the verifier reward without teacher log-probabilities. For a fair comparison, we use the same mathematical training data, student initialization, training setup, and evaluation protocol as in the corresponding distillation experiments, with the main implementation difference being that the RL baselines use a group size of 8.

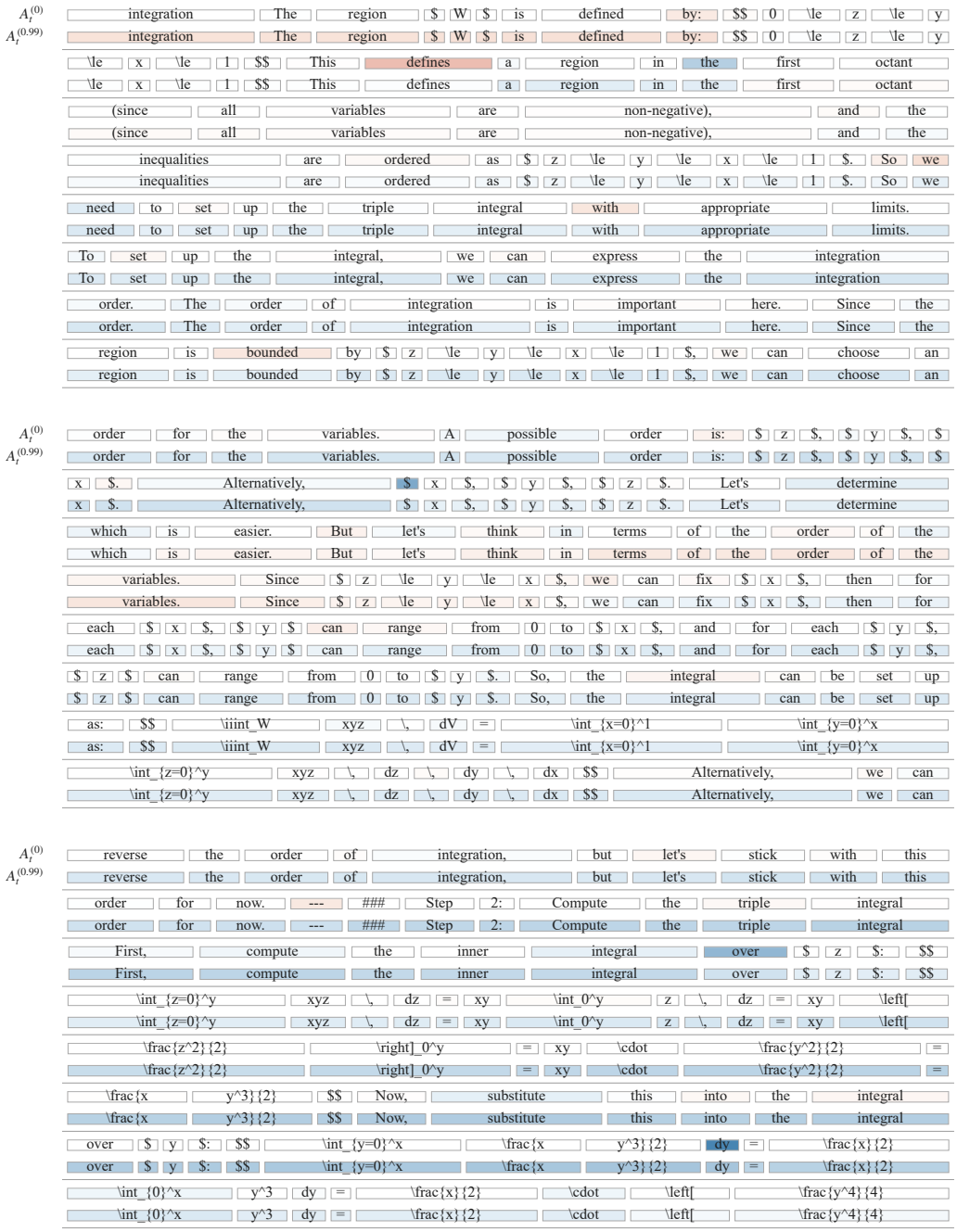
The complete results are reported in Table 6. Overall, γOPD consistently outperforms the RL baselines. The relatively limited gains of pure RL may stem from inefficient exploration under sparse verifier rewards and a small rollout group. In contrast, γOPD combines direct teacher guidance with verifier feedback, enabling more effective optimization without relying on group-based rollout exploration.

F.3 Complete Token-Level Advantage Visualization

This appendix provides the complete token-level visualizations behind Figure 5. Each response is shown as a sequence of consecutive full-width panels rather than as a collection of small subfigures, so that individual tokens and advantage magnitudes remain readable. The upper row in every panel shows the local advantage $A_t^{(0)}$, while the lower row shows the normalized temporally propagated advantage $A_t^{(0.99)}$.

As shown in Figure 6, for the correct response, $A_t^{(0)}$ gives a sparse local signal, whereas $A_t^{(0.99)}$ propagates information from later reasoning steps and assigns stronger positive credit to the key mathematical derivation. As shown in Figure 7, for the incorrect response, $A_t^{(0)}$ penalizes tokens largely unrelated to the actual error, while $A_t^{(0.99)}$ assigns stronger negative credit to the erroneous formula derivation without excessively penalizing the final end-of-sequence token.




$A_t^{(0)}$
integration The region $ W $ is defined by: $$ 0 \le z \le y
$A_t^{(0.99)}$
integration The region $ W $ is defined by: $$ 0 \le z \le y
\le x \le 1 $$ This defines a region in the first octant
\le x \le 1 $$ This defines a region in the first octant
(since all variables are non-negative), and the
(since all variables are non-negative), and the
inequalities are ordered as $ z \le y \le x \le 1 $. So we
inequalities are ordered as $ z \le y \le x \le 1 $. So we
need to set up the triple integral with appropriate limits.
need to set up the triple integral with appropriate limits.
To set up the integral, we can express the integration
To set up the integral, we can express the integration
order. The order of integration is important here. Since the
order. The order of integration is important here. Since the
region is bounded by $ z \le y \le x \le 1 $, we can choose an
region is bounded by $ z \le y \le x \le 1 $, we can choose an
$A_t^{(0)}$
order for the variables. A possible order is: $ z $, $ y $, $
$A_t^{(0.99)}$
order for the variables. A possible order is: $ z $, $ y $, $
x $. Alternatively, $ x $, $ y $, $ z $. Let's determine
x $. Alternatively, $ x $, $ y $, $ z $. Let's determine
which is easier. But let's think in terms of the order of the
which is easier. But let's think in terms of the order of the
variables. Since $ z \le y \le x $, we can fix $ x $, then for
variables. Since $ z \le y \le x $, we can fix $ x $, then for
each $ x $, $ y $ can range from 0 to $ x $, and for each $ y $,
each $ x $, $ y $ can range from 0 to $ x $, and for each $ y $,
$ z $ can range from 0 to $ y $. So, the integral can be set up
$ z $ can range from 0 to $ y $. So, the integral can be set up
as: $$ \iiint_W xyz \, dV = \int_{x=0}^1 \int_{y=0}^x
as: $$ \iiint_W xyz \, dV = \int_{x=0}^1 \int_{y=0}^x
\int_{z=0}^y xyz \, dz \, dy \, dx $$ Alternatively, we can
\int_{z=0}^y xyz \, dz \, dy \, dx $$ Alternatively, we can
$A_t^{(0)}$
reverse the order of integration, but let's stick with this
$A_t^{(0.99)}$
reverse the order of integration, but let's stick with this
order for now. --- ### Step 2: Compute the triple integral
order for now. --- ### Step 2: Compute the triple integral
First, compute the inner integral over $ z $: $$
First, compute the inner integral over $ z $: $$
\int_{z=0}^y xyz \, dz = xy \int_0^y z \, dz = xy \left[
\int_{z=0}^y xyz \, dz = xy \int_0^y z \, dz = xy \left[
\frac{z^2}{2} \right]_0^y = xy \cdot \frac{y^2}{2} =
\frac{z^2}{2} \right]_0^y = xy \cdot \frac{y^2}{2} =
\frac{x y^3}{2} $$ Now, substitute this into the integral
\frac{x y^3}{2} $$ Now, substitute this into the integral
over $ y $: $$ \int_{y=0}^x \frac{x y^3}{2} dy = \frac{x}{2}
over $ y $: $$ \int_{y=0}^x \frac{x y^3}{2} dy = \frac{x}{2}
\int_0^x y^3 dy = \frac{x}{2} \cdot \left[\frac{y^4}{4}
\int_0^x y^3 dy = \frac{x}{2} \cdot \left[\frac{y^4}{4}

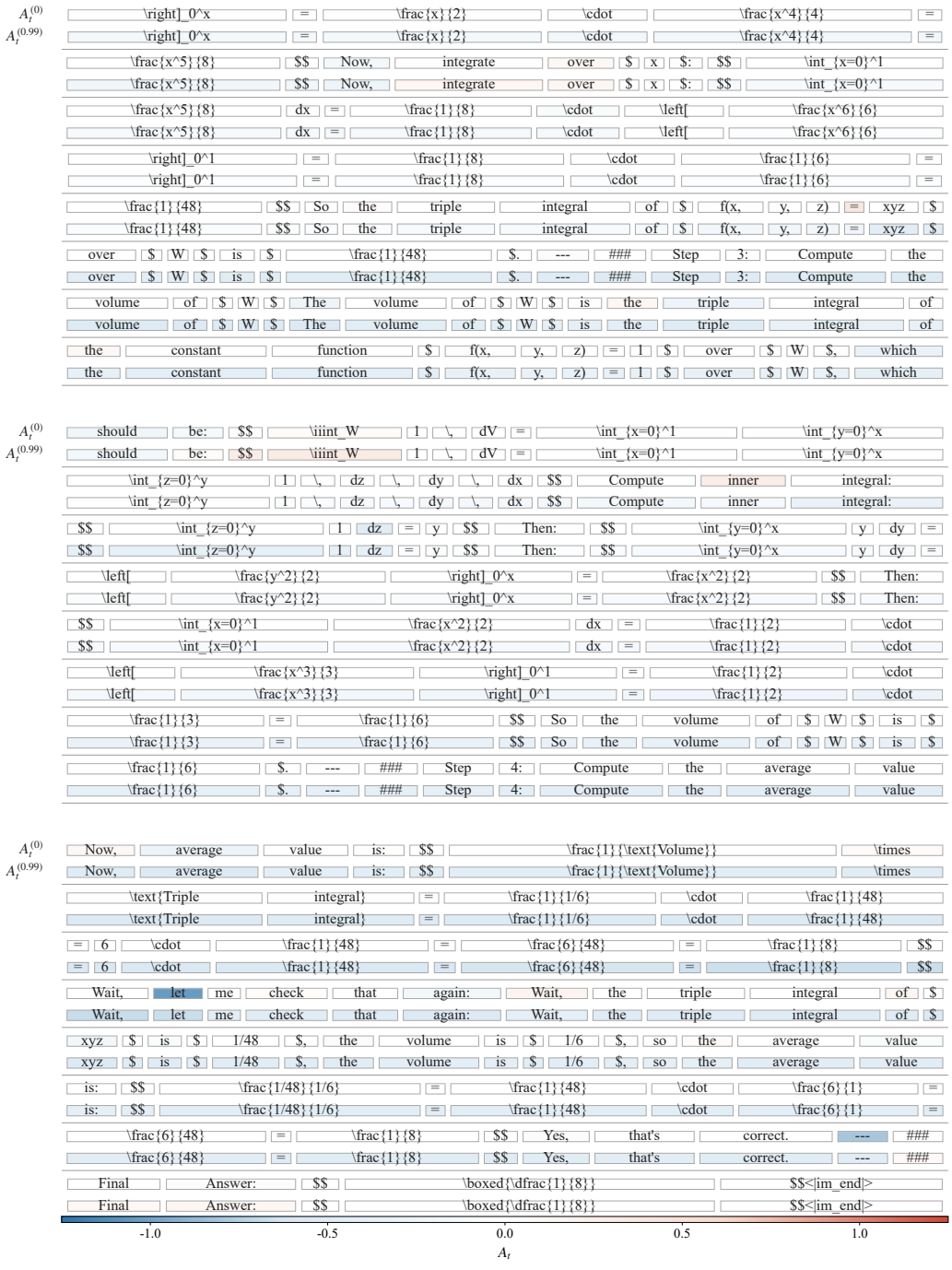


Figure 6. Complete token-level advantage visualization for the correct response. The response is shown in token order across seven consecutive panels, with $A_t^{(0)}$ in the upper row and $A_t^{(0.99)}$ in the lower row. Blue and red indicate negative and positive advantages, respectively.



Figure 7. Complete token-level advantage visualization for the incorrect response. The two consecutive panels show how $A_t^{(0)}$ can penalize tokens unrelated to the actual error, whereas $A_t^{(0.99)}$ assigns stronger negative credit to the erroneous formula derivation.