On-policy Distillation with Verifiable Reward

Wenze Lin[1*†], **Jiale Zhao**[1,2*], **Xitai Jiang**[1*], **Songde Rao**[3], **Yining Li**[1], **Shenzhi Wang**[1], **Bingxiang He**[4], **and Gao Huang**[1✉]

[1] LeapLab, Tsinghua University [2] Beihang University [3] SMS, Peking University [4] NLPLab, Tsinghua University

* Equal Contribution † Project Lead ✉ Corresponding Author

Reinforcement Learning with Verifiable Rewards (RLVR) and on-policy distillation (OPD) have become two widely adopted paradigms for post-training large language models. However, RLVR suffers from sparse task-level feedback, while OPD provides dense token-level guidance but ignores trajectory correctness, limiting its performance to that of the teacher. Combining them is a promising direction: OPD supplies dense supervisory signals, while RLVR provides task-level correctness. Nevertheless, existing integrations often rely on weighted combination or heuristic switching, introducing extra hyperparameters and trade-offs. We propose **On-policy Distillation with Verifiable Reward (OPDVR)**, a simple yet effective method that seamlessly combines OPD and RLVR without adding any hyperparameters. We first reformulate the implicit reward of sampled-token OPD based on trajectory correctness, then apply a ReLU gating mechanism to ensure that correct trajectories receive non-negative rewards and incorrect ones receive non-positive rewards—thereby aligning the distillation signal with task success while preserving the teacher's distributional guidance. Furthermore, our modification transforms sampled-token OPD into a proper RLVR method, making it readily combinable with any policy gradient algorithm, such as GRPO. Experiments on six reasoning benchmarks show that OPDVR consistently outperforms standard OPD. Our code is available at https://github.com/LeapLabTHU/OPDVR.

1. Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has become a prevalent and effective post-training paradigm for reasoning tasks (Guo et al., 2025; Shao et al., 2024; Jaech et al., 2024; Trinh et al., 2024; Yang et al., 2024). RLVR delivers explicit reward signals based on task outcomes, such as correct mathematical final answers or compilable code, which directly align model optimization with core task objectives. However, these rewards are typically sparse, providing little supervision for intermediate steps and making credit assignment challenging. Recently, On-policy Distillation (OPD) has emerged as a promising post-training paradigm (Xu et al., 2026; Xiao et al., 2026; Yang et al., 2025a; Zeng et al., 2026). Unlike RLVR, which relies on sparse outcome-level rewards, OPD leverages a teacher model to provide dense token-level guidance. However, OPD's objective is purely distributional—it drives the student to mimic the teacher's output distribution without considering whether the generated response is correct or incorrect, limiting the student's performance to that of the teacher.

The complementary limitations of RLVR and OPD make combining them a natural and promising direction, where OPD provides fine-grained dense guidance and RLVR offers reliable task-level correctness supervision. Existing approaches typically treat OPD and RLVR as two separate methods, either

Correspond to: linwz25@mails.tsinghua.edu.cn, gaohuang@tsinghua.edu.cn.

combining them explicitly or selectively applying one based on heuristic criteria. Despite their empirical effectiveness, such designs often introduce extra hyperparameters and heuristic trade-offs (Cai et al., 2026; Yang et al., 2026a; Li et al., 2026a; Wang et al., 2026).

In this work, we propose **On-policy Distillation with Verifiable Reward (OPDVR)**. We apply an extremely simple ReLU gating mechanism to sampled-token OPD, seamlessly combining OPD and RLVR without introducing any hyperparameters. We first revisit sampled-token OPD from an RLVR perspective and provide a mathematical reformulation of sample-token OPD's reward. From the RLVR view, if we separate by trajectory correctness, sampled-token OPD can be viewed as applying token-level supervision by weighting binary task correctness rewards with teacher-student distribution discrepancies. Specifically, for sampled tokens that constitute correct reasoning trajectories, the model is assigned a token-level reward $+1$ weighted by $\log(\pi_T/\pi_\theta)$; for tokens leading to incorrect trajectories, the model receives a reward -1 weighted by $\log(\pi_\theta/\pi_T)$. However, we identify a critical limitation of this inherent reward design: both $\log(\pi_T/\pi_\theta)$ and $\log(\pi_\theta/\pi_T)$ are unbounded and can be either positive or negative. However, a key empirical principle shared by mainstream RLVR algorithms (Schulman et al., 2017; Guo et al., 2025; Shao et al., 2024) is that all tokens leading to a correct final outcome should receive non-negative advantages, while tokens leading to an incorrect outcome should receive non-positive advantages. This principle ensures that every token in a correct trajectory is treated as a valid prediction and reinforced accordingly, while every token in an incorrect trajectory is treated as an erroneous prediction and suppressed accordingly. Sampled-token OPD violates this principle: on correct trajectories, a negative value of $\log(\pi_T/\pi_\theta)$ penalizes valid token behaviors, while on incorrect trajectories, a negative value of $\log(\pi_\theta/\pi_T)$ encourages erroneous token predictions.

Motivated by this gap, we use a ReLU gating mechanism to standardize the reward signs according to trajectory correctness: it enforces non-negative $\log(\pi_T/\pi_\theta)$ values for all verified correct trajectories and ensures non-negative $\log(\pi_\theta/\pi_T)$ values for all incorrect trajectories. This minimal correction directly converts conventional sampled-token OPD into a valid RLVR method while preserving the guidance from the teacher model. Specifically, for correct trajectories, larger $\log(\pi_T/\pi_\theta)$ values—indicating accurate student predictions where the teacher is more confident than the student—yield stronger rewards. For incorrect trajectories, larger $\log(\pi_\theta/\pi_T)$ values—corresponding to wrong student predictions where the student is more confident than the teacher—incur heavier penalties. This learning paradigm aligns well with human learning intuition, which **reinforces reliable correct behaviors and suppresses overconfident erroneous predictions**. Furthermore, by reformulating sampled-token OPD into an RLVR formulation, our OPDVR supports seamless integration with existing prominent RL algorithms, such as GRPO, DAPO and PPO. To instantiate this, we combine OPDVR with GRPO to obtain a new variant, which we call **Group Relative Policy Distillation (GRPD)**. Overall, our method endows the model with dual supervisory signals: the rigorous task-level verifiable correctness from RLVR and the dense token-level distributional guidance from teacher distillation, leading to more robust and effective post-training optimization for reasoning tasks. Our experiments show that both OPDVR and GRPD consistently outperform OPD across all benchmarks.

2. Related Work

Reinforcement Learning with Verifiable Rewards (RLVR) Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a highly effective paradigm for post-training large language models (Schulman et al., 2017; Jaech et al., 2024; Trinh et al., 2024; Yang et al., 2024; Guo et al., 2025; Shao et al., 2024). RLVR leverages rule-based verifiers to provide objective, deterministic signals such as answer correctness or code compilation. Various extensions have been proposed to improve RLVR, such as controlling length (Liu et al., 2025; Sui et al., 2025; Xiang et al., 2025; Hammoud et al., 2025; Hou et al., 2025; Huang et al., 2026) and stabilizing entropy (Petrenko et al., 2026; Yang et al., 2025b; Jiang et al., 2025; Su et al., 2025). However, a fundamental limitation persists: the reward is sparse, leading to a severe credit assignment problem.

On-policy Distillation On-policy Distillation (OPD) has recently emerged as an efficient post-training paradigm that provides dense, token-level supervisory signals by distilling a teacher model's output

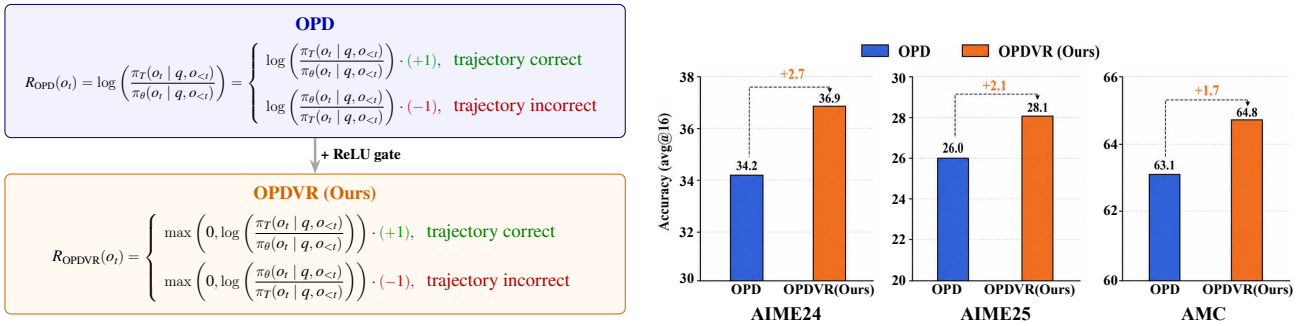


Figure 1: Left: Overview of OPDVR. The ReLU gating mechanism ensures correct trajectories receive non-negative rewards and incorrect ones receive non-positive rewards, while preserving the teacher's distributional guidance. Right: Results on AIME24, AIME25, and AMC under the same-architecture setting (Qwen3-4B ← Qwen3-4B-RL), reported as avg@16 accuracy.

distribution into the student (Xu et al., 2026; Xiao et al., 2026; Yang et al., 2025a; Zeng et al., 2026). Unlike RLVR's sparse outcome-based rewards, OPD offers fine-grained guidance at every generation step. A growing line of work has explored various aspects of OPD, including its failure modes (Fu et al., 2026; Li et al., 2026b), how to determine which tokens are worth training (Xing et al., 2026; Jin et al., 2026), and its training stability issues in practice (Oh et al., 2026). However, OPD's objective is purely distributional: it drives the student to mimic the teacher's output distribution without considering whether the generated response is correct or incorrect. This observation naturally raises the question of whether OPD can be combined with RLVR, where trajectory correctness provides the ultimate optimization signal. Combining the dense supervision of OPD with the task-level verifiability of RLVR thus represents a promising and increasingly relevant direction.

Combining OPD with RLVR Given the complementary strengths of OPD (dense token-level guidance) and RLVR (task-level correctness), several recent works have attempted to combine them. These include directly weighting the two objectives (Hubotter et al., 2026), applying OPD and RLVR separately depending on the sign of the advantage (Wang et al., 2026; Cai et al., 2026), aligning outcome rewards with process-level distillation signals (Hou et al., 2026), and leveraging teacher-student probability ratios for to weight the advantage in GRPO (Yang et al., 2026a). Despite their empirical gains, these approaches often treat OPD and RLVR as distinct objectives or losses that must be either explicitly combined or selectively applied based on heuristic criteria, often introducing additional hyperparameters for balancing the two terms, or relies on heuristic trade-offs. Instead, our method implicitly transforms sampled-token OPD into a proper RLVR method using a simple gated mechanism, while preserving the teacher's distributional guidance—achieving the benefits of both paradigms without explicit multi-objective balancing.

3. Preliminaries

3.1. Reinforcement Learning with Verifiable Rewards (RLVR)

Reinforcement Learning with Verifiable Rewards (RLVR) optimizes a policy using reward signals that can be automatically verified against ground truth. In mathematical reasoning tasks, the reward indicates whether the final answer is correct. A common choice for REINFORCE is $R = +1$ for a correct trajectory and $R = -1$ for an incorrect one, while many other RLVR methods (including GRPO) use $R = +1$ for correct and $R = 0$ for incorrect.

Consider a policy π_θ that generates a response $o = \{o_1, \ldots, o_{|o|}\}$ given a prompt q. The RLVR objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{o \sim \pi_\theta(\cdot|q)}\left[R\right],$$

where R is the trajectory-level reward.

To optimize this objective via gradient ascent, we compute the gradient:

$$\nabla_\theta \mathcal{J}(\theta) = \nabla_\theta \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[R\right].$$

Using the REINFORCE (log-derivative) trick, we rewrite this as:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[R \cdot \nabla_\theta \log \pi_\theta(o \mid q)\right],$$

where $\log \pi_\theta(o \mid q) = \sum_{t=1}^{|o|} \log \pi_\theta(o_t \mid q, o_{<t})$.

We sample a single response $o \sim \pi_\theta(\cdot \mid q)$ to obtain a Monte Carlo estimate:

$$\nabla_\theta \mathcal{J}(\theta) \approx R \cdot \sum_{t=1}^{|o|} \nabla_\theta \log \pi_\theta(o_t \mid q, o_{<t}).$$

This gradient can be implemented by minimizing the following loss over the sequence:

$$\mathcal{L}_{\text{RLVR}}(\theta) = -R \cdot \sum_{t=1}^{|o|} \log \pi_\theta(o_t \mid q, o_{<t}).$$

When $R = +1$, minimizing this loss increases the probability of the entire response (reward). When $R = -1$, it decreases the probability (penalty). This formulation embodies the core RL principle: correct actions are rewarded, incorrect ones are penalized.

The REINFORCE formulation above suffers from high variance, as it relies on a single trajectory-level reward. A more stable alternative is Group Relative Policy Optimization (GRPO) (Guo et al., 2025; Shao et al., 2024). Specifically, for a group of G responses $\{o_1, \ldots, o_G\}$ sampled from the policy, GRPO normalizes each response's reward against the group mean and standard deviation:

$$\hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}.$$

Note that GRPO typically uses $R \in \{0, 1\}$ (1 for correct, 0 for incorrect) rather than $\{-1, +1\}$, since the group-relative normalization naturally centers the advantages around zero.

This group-relative advantage reduces variance and serves as a stable, computationally efficient alternative to direct binary rewards. For simplicity, we present the on-policy version of GRPO without clipping or KL penalty; the loss is:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \hat{A}_{i,t} \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}).$$

3.2. On-policy Distillation

On-policy Distillation (OPD) optimizes a student policy π_θ toward a teacher policy π_T by minimizing their reverse KL divergence:

$$\min_\theta \ \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[\sum_{t=1}^{|o|} D_{\text{KL}}(\pi_\theta(\cdot \mid q, o_{<t}) \,\|\, \pi_T(\cdot \mid q, o_{<t}))\right].$$

The **full-vocabulary OPD** computes this exact KL at every token position:

$$\mathcal{L}_{\text{OPD}}^{\text{full}}(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[\sum_{t=1}^{|o|}\sum_{v\in\mathcal{V}} \pi_\theta(v \mid q, o_{<t}) \log \frac{\pi_\theta(v \mid q, o_{<t})}{\pi_T(v \mid q, o_{<t})}\right].$$

This loss is directly minimized. In practice, the log-ratio $\log(\pi_\theta/\pi_T)$ is treated as a constant reward with stop-gradient when computing gradients, which is equivalent to the exact gradient of the reverse KL via the log-derivative trick. However, computing the reverse KL over the full vocabulary at every token position is computationally expensive, especially for large language models with very large vocabularies.

A practical compromise is **Top-k OPD**, which restricts the KL computation to the k tokens with the highest student probabilities:

$$\mathcal{L}_{\text{OPD}}^{\text{top-}k}(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[\sum_{t=1}^{|o|} D_{\text{KL}}\left(\bar{\pi}_\theta^{(S_t)} \,\|\, \bar{\pi}_T^{(S_t)}\right)\right],$$

where S_t is the set of top-k tokens.

The most widely adopted variant is **sampled-token OPD**. It uses a single Monte Carlo sample $o_t \sim \pi_\theta(\cdot \mid q, o_{<t})$ to approximate the reverse KL gradient:

$$\mathcal{L}_{\text{OPD}}^{\text{sample}}(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[\sum_{t=1}^{|o|} \log\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})}\right].$$

The log-ratio $\log(\pi_\theta/\pi_T)$ is treated as a constant reward with stop-gradient.

4. Method

4.1. Revisiting Sampled-Token OPD from an RLVR Perspective

We begin with the sampled-token OPD loss. Recall that sampled-token OPD approximates the reverse KL gradient using a single Monte Carlo sample. The loss over the entire response $o = \{o_1, \ldots, o_{|o|}\}$ is defined as:

$$\mathcal{L}_{\text{OPD}}^{\text{sample}}(\theta) = \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[\sum_{t=1}^{|o|} \log\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})}\right].$$

For a sampled response $o \sim \pi_\theta(\cdot \mid q)$, the per-token loss is:

$$\ell_t = \log\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})}.$$

The gradient of the sequence-level loss with respect to θ is:

$$\nabla_\theta \mathcal{L}_{\text{OPD}}^{\text{sample}}(\theta) = \sum_{t=1}^{|o|} \log\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})} \cdot \nabla_\theta \log \pi_\theta(o_t \mid q, o_{<t}).$$

We now compare this with the standard RLVR gradient. Recall from Section 3 that the RLVR loss for a trajectory with reward R is:

$$\mathcal{L}_{\text{RLVR}}(\theta) = -R \cdot \sum_{t=1}^{|o|} \log \pi_\theta(o_t \mid q, o_{<t}).$$

Its gradient is:

$$\nabla_\theta \mathcal{L}_{\text{RLVR}}(\theta) = -R \cdot \sum_{t=1}^{|o|} \nabla_\theta \log \pi_\theta(o_t \mid q, o_{<t}).$$

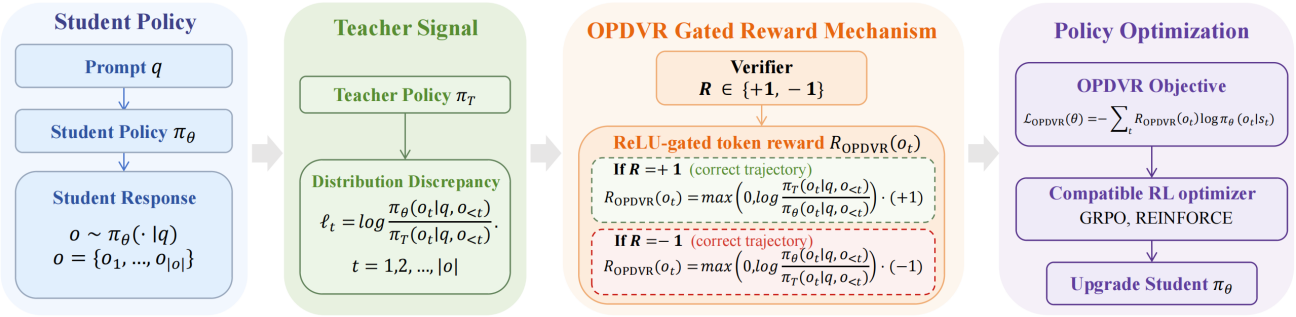


Figure 2: OPDVR method

We observe that the sampled-token OPD gradient shares the same form as the RLVR gradient, where the log-ratio term corresponds to the reward coefficient. By matching the coefficients of $\nabla_\theta \log \pi_\theta(o_t|q, o_{<t})$, we obtain:

$$-R = \log \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})}.$$

this observation allows us to reinterpret the log-ratio term as an implicit token-level reward $R_{\text{OPD}}(o_t)$

$$R_{\text{OPD}}(o_t) = -\log \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})} = \log \frac{\pi_T(o_t \mid q, o_{<t})}{\pi_\theta(o_t \mid q, o_{<t})}.$$

However, unlike RLVR, where the reward sign is determined by the verifier outcome, the sign of Ropd is solely determined by the teacher-student probability ratio. To analyze their alignment, we consider two cases based on trajectory correctness:

$$R_{\text{OPD}}(o_t) = \begin{cases} \log \dfrac{\pi_T(o_t \mid q, o_{<t})}{\pi_\theta(o_t \mid q, o_{<t})} \cdot (+1), & \text{trajectory correct,} \\ \log \dfrac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})} \cdot (-1), & \text{trajectory incorrect.} \end{cases}$$

In other words, on correct trajectories, the gradient is weighted by $\log(\pi_T/\pi_\theta)$; on incorrect trajectories, it is weighted by $\log(\pi_\theta/\pi_T)$ with a negative sign.

The empirical RLVR principle requires that correct trajectories receive positive rewards and incorrect trajectories receive negative rewards. However, the sign of $\log(\pi_T/\pi_\theta)$ is determined by whether $\pi_T > \pi_\theta$ or $\pi_T < \pi_\theta$, which is independent of trajectory correctness. This creates two failure cases: correct trajectories can receive negative rewards when $\pi_T < \pi_\theta$, and incorrect trajectories can receive positive rewards when $\pi_T > \pi_\theta$. Both violate the RL principle.

4.2. OPDVR: A Simple Gated Mechanism

We propose On-policy Distillation with Verifiable Reward (OPDVR) to resolve this issue. The solution is **extremely simple**: apply a ReLU gate to enforce RLVR compliance on the sampled token while preserving the teacher's distributional guidance.

The reward can be written as:

$$R_{\text{OPDVR}}(o_t) = \begin{cases} \max\left(0, \log \dfrac{\pi_T(o_t \mid q, o_{<t})}{\pi_\theta(o_t \mid q, o_{<t})}\right) \cdot (+1), & \text{trajectory correct,} \\ \max\left(0, \log \dfrac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_T(o_t \mid q, o_{<t})}\right) \cdot (-1), & \text{trajectory incorrect.} \end{cases}$$

The corresponding loss is:

$$\mathcal{L}_{\text{OPDVR}}(\theta) = -\sum_{t=1}^{|o|} R_{\text{OPDVR}}(o_t) \cdot \log \pi_\theta(o_t \mid q, o_{<t}).$$

This gated mechanism ensures that:

- Correct trajectories receive non-negative rewards, and incorrect trajectories receive non-positive rewards, aligning the reward sign with trajectory correctness.
- For tokens on correct trajectories, the reward magnitude is larger when the teacher is more confident than the student, i.e., $\log(\pi_T/\pi_\theta)$ is larger. This encourages the model to reinforce choices that are both correct and reliable according to the teacher.
- For tokens on incorrect trajectories, the penalty magnitude is larger when the student is more confident than the teacher, i.e., $\log(\pi_\theta/\pi_T)$ is larger. This forces the model to suppress overconfident mistakes.

4.3. Interpreting the ReLU Gating Mechanism as a Conditional Mask

In this subsection, we interpret the ReLU gating mechanism in OPDVR as a *conditional token mask*: it selectively zeroes out gradient updates on tokens whose learning direction conflicts with the verifier signal. Standard sampled-token OPD lacks this mask, so it inevitably updates all tokens based solely on the teacher-student confidence ratio, regardless of trajectory correctness.

To see this, recall the standard OPD gradient for a sampled token a:

$$g_{\text{OPD}} = \underbrace{\left[\log \frac{\pi_T}{\pi_\theta}\right]_+ \nabla_\theta \log \pi_\theta(a)}_{\text{Term A: push up (teacher more confident)}} - \underbrace{\left[\log \frac{\pi_\theta}{\pi_T}\right]_+ \nabla_\theta \log \pi_\theta(a)}_{\text{Term B: pull down (student more confident)}}. \tag{1}$$

The OPDVR gradient instead applies the ReLU gate conditioned on the verifier reward $R \in \{+1, -1\}$:

$$g_{\text{OPDVR}} = \begin{cases} \text{Term A}, & R = +1, \\ -\text{Term B}, & R = -1. \end{cases} \tag{2}$$

Thus, compared to standard OPD, OPDVR *removes* exactly two types of tokens whose updates conflict with the verifier:

- **Type I Conflicting Token (Correct Trajectory, $\pi_\theta > \pi_T$):** When the trajectory is correct ($R = +1$), the student has already produced the right answer. If the student is more confident than the teacher on a specific token ($\pi_\theta > \pi_T$), then Term B is activated. Standard OPD applies a *negative* gradient, pulling down this correct token's probability to match the teacher's lower confidence. This update works against the verifier signal in two ways:
  1. The student's higher confidence is *consistent* with the correct outcome.
  2. Reducing the student's confidence on a correct answer introduces a distributional distortion that is not driven by task performance.
- **Type II Conflicting Token (Incorrect Trajectory, $\pi_T > \pi_\theta$):** When the trajectory is incorrect ($R = -1$), the student has produced a wrong answer. If the teacher is more confident than the student on a specific token ($\pi_T > \pi_\theta$), then Term A is activated. Standard OPD applies a *positive* gradient, pushing up this incorrect token's probability to align with the teacher. This update works against the verifier signal in two ways:
  1. The teacher's high confidence does not accompany a correct trajectory here.
  2. Increasing the probability of tokens on an incorrect trajectory reinforces a reasoning pattern that the verifier has marked as wrong.

By masking out these two token types, OPDVR preserves the student's correct high-confidence predictions and withholds the teacher's guidance on wrong answers. The teacher still controls the *magnitude* of the update via the log-ratio, but the verifier now determines the *direction*—reinforce or suppress—so that the distillation process never works against the task reward. We provide a further analysis in Appendix A.

4.4. Group Relative Policy Distillation (GRPD)

Having established that OPDVR transforms sampled-token OPD into a proper RLVR method with a binary verifier reward, a natural extension is to replace this coarse binary signal with a more nuanced, group-relative advantage estimate like GRPO (Guo et al., 2025; Shao et al., 2024).

Recall from Section 3 that GRPO computes a group-relative advantage $\hat{A}_{i,t}$ for each token position t in response o_i:

$$\hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G})},$$

where $R_i \in \{0, 1\}$ is the verifier reward for response o_i, and G is the group size. The key property of $\hat{A}_{i,t}$ is that it is positive for responses that are better than the group average, and negative for those that are worse—capturing relative performance within the sampled batch.

We now apply the same ReLU gating logic, but with the binary correctness sign R replaced by the group-relative advantage $\hat{A}_{i,t}$. The GRPD reward becomes:

$$R_{\text{GRPD}}(o_{i,t}) = \begin{cases} \max\left(0, \log \dfrac{\pi_T(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}\right) \cdot (+1), & \hat{A}_{i,t} > 0, \\ \max\left(0, \log \dfrac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_T(o_{i,t} \mid q, o_{i,<t})}\right) \cdot (-1), & \hat{A}_{i,t} < 0. \end{cases}$$

Equivalently, this can be written compactly as:

$$R_{\text{GRPD}}(o_{i,t}) = \text{sign}(\hat{A}_{i,t}) \cdot \text{ReLU}\left(\text{sign}(\hat{A}_{i,t}) \cdot \log \frac{\pi_T(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}\right),$$

where $\text{sign}(\hat{A}_{i,t})$ ensures the direction of the update is governed by the advantage sign.

The corresponding loss is:

$$\mathcal{L}_{\text{GRPD}}(\theta) = -\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} R_{\text{GRPD}}(o_{i,t}) \cdot \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}).$$

5. Experiments

5.1. Main Experimental Settings

We conduct experiments on both same-architecture and cross-architecture distillation settings to evaluate the effectiveness of our method.

Same-architecture setting. We use Qwen3-4B-nonthinking as the student model and distill it from a teacher model of the same architecture, which is obtained by training Qwen3-4B with GRPO on the DeepMath dataset (He et al., 2026). Following Yang et al. (2026b), we use the filtered subset of the DeepMath dataset consisting of 57k samples with difficulty level ≥ 6.

Cross-architecture setting. We use the DAPO-Math-17k dataset (Yu et al., 2026). The teacher model is Qwen3-4B-base, fine-tuned with GRPO for 3 epochs on the same dataset. The student model is Qwen3-1.7B-base. The distillation training runs for 3 epochs.

Table 1: Results on same-architecture distillation (Qwen3-4B ← Qwen3-4B-RL). All models are evaluated with avg@16 accuracy.

Method	AIME24	AIME25	AMC	MATH500	Minerva	OlympiadBench	Avg.
Student (Qwen3-4B)	24.0	15.8	60.8	80.9	27.6	42.9	42.0
Teacher (Qwen3-4B-RL)	36.0	29.0	65.9	87.0	35.4	49.3	50.4
Sampled-Token OPD	34.2	26.0	63.1	**85.5**	31.6	46.5	47.8
Top-64 OPD	34.6	23.5	62.0	85.0	32.2	46.8	47.4
OPDVR (Ours)	**36.9**	**28.1**	**64.8**	84.7	**33.2**	**47.0**	**49.1**

Table 2: Results on cross-architecture distillation (Qwen3-1.7B-Base ← Qwen3-4B-Base-RL). All models are evaluated with avg@16 accuracy.

Method	AIME24	AIME25	AMC	MATH500	Minerva	OlympiadBench	Avg.
Student (Qwen3-1.7B-Base)	4.1	1.7	23.2	48.9	8.9	17.1	17.3
Teacher (Qwen3-4B-Base-RL)	10.6	13.1	40.3	74.2	17.2	30.0	30.9
Sampled-Token OPD	6.5	2.1	24.8	59.1	11.5	21.6	20.9
Top-64 OPD	**8.5**	**3.3**	26.4	60.1	10.7	21.4	21.7
OPDVR (Ours)	**8.5**	**3.3**	**30.3**	**60.8**	**11.6**	**22.0**	**22.8**

Benchmarks. We evaluate all models on six reasoning benchmarks: AIME24, AIME25, AMC, MATH500, Minerva, and OlympiadBench.

5.2. Main Results

We compare our method OPDVR against standard sampled-token OPD and top-64 OPD on both same-architecture and cross-architecture distillation settings. Tables 1 and 2 report the performance on six reasoning benchmarks.

As shown in both tables, OPDVR consistently outperforms standard sampled-token OPD and top-64 OPD across all six benchmarks in both distillation settings. In the same-architecture setting, OPDVR achieves gains of 2.7 points on AIME24 and 2.1 points on AIME25 over sampled-token OPD, and even surpasses the teacher model on AIME24. In the cross-architecture setting, OPDVR delivers substantial improvements over the baselines, with gains of 5.5 points on AMC and 1.7 points on MATH500 over sampled-token OPD, highlighting its robustness to architectural differences.

5.3. Group Relative Policy Distillation

To further validate the effectiveness of replacing the binary verifier reward with group-relative advantages, we instantiate OPDVR with GRPO-style advantage estimation (describe in Section 4.4) and evaluate it under the same-architecture setting. The teacher model is the same Qwen3-4B-Nonthinking model trained with GRPO on DeepMath. The student model is Qwen3-4B-nonthinking. To more cleanly evaluate the effectiveness of replacing binary rewards with group-relative advantages, we conduct this experiment on DAPO-Math-17K, distinct from the teacher's DeepMath training data. The group size is set to $G = 8$. We compare GRPD against GRPO and standard sampled-token OPD.

Table 3 reports the results. GRPD consistently outperforms both GRPO and OPD across all six benchmarks, with notable gains of 6.5 points on AIME24 and 10.9 points on AIME25 over GRPO. It also surpasses OPD on five out of six benchmarks, achieving a 2.8-point improvement on AIME24. These results demonstrate that combining group-relative advantage estimation with the ReLU-gated distillation signal yields stronger and more stable improvements than either method alone.

Table 3: Results on Group Relative Policy Distillation (Qwen3-4B ← Qwen3-4B-RL). All models are evaluated with avg@16 accuracy.

Method	AIME24	AIME25	AMC	MATH500	Minerva	OlympiadBench	Avg.
Student (Qwen3-4B)	24.0	15.8	60.8	80.9	27.6	42.9	42.0
Teacher (Qwen3-4B-RL)	36.0	29.0	65.9	87.0	35.4	49.3	50.4
GRPO	28.3	20.8	62.3	83.9	28.9	44.6	44.8
OPD	32.0	**31.7**	65.6	85.4	28.9	46.6	48.4
GRPD (Ours)	**34.8**	**31.7**	**67.0**	**85.6**	**30.5**	**47.0**	**49.4**

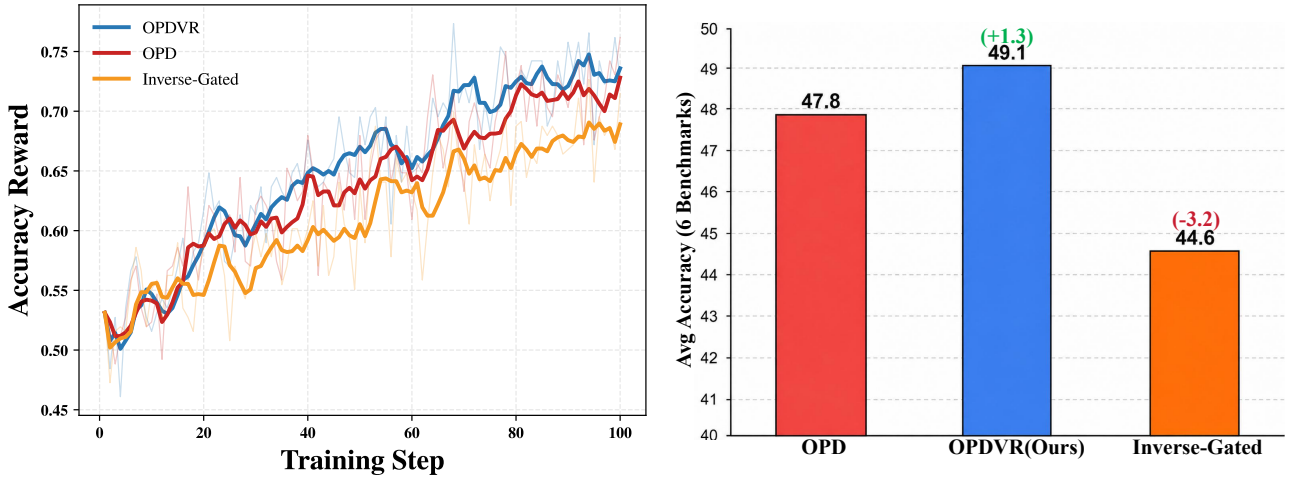


Figure 3: Ablation study on the gating mechanism. **Left**: training-time accuracy reward of OPD, OPDVR, and the inverse-gated variant. **Right**: average accuracy over six benchmarks.

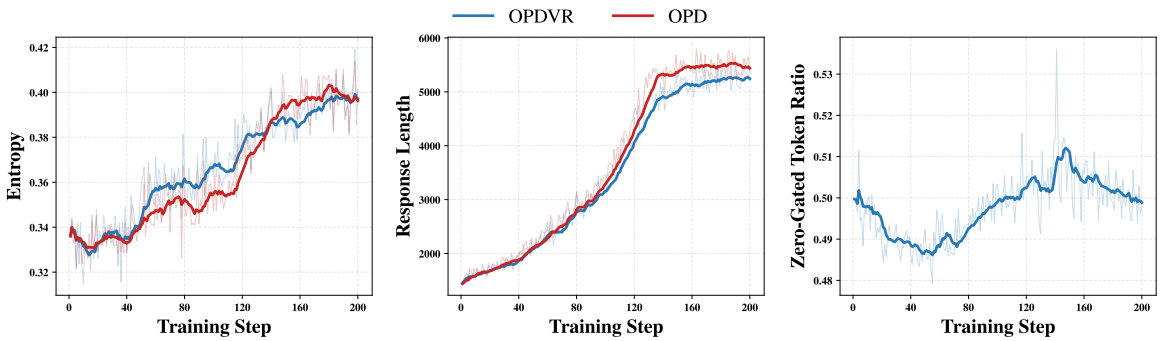
5.4. Ablation Study: Inverse-Gated Experiment

To examine the effectiveness of the gating mechanism in OPDVR, we conduct an inverse-gated ablation under the same-architecture setting (Qwen3-4B ← Qwen3-4B-RL) on DeepMath, following the same experimental setup as the main experiments.. Recall that OPDVR keeps a token only when its update direction agrees with the verifier signal: on a correct trajectory ($R{=}+1$), tokens where the teacher is more confident than the student ($\pi_T > \pi_\theta$) are kept and tokens where the student is already more confident ($\pi_\theta > \pi_T$) are gated out; on an incorrect trajectory ($R{=}-1$), the roles are reversed – tokens where the student is more confident than the teacher are kept, and tokens where the teacher is more confident are gated out. The inverse-gated variant swaps these two sets exactly: it keeps the tokens that OPDVR gates out (the student-more-confident tokens on correct trajectories and the teacher-more-confident tokens on incorrect trajectories), and gates out the tokens that OPDVR keeps, while keeping the masking ratio and everything else unchanged.

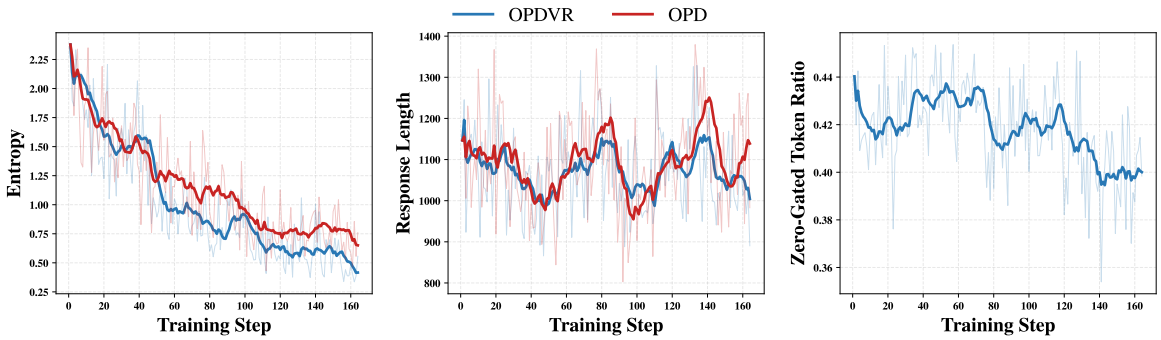
Table 4 shows the final benchmark results, and Figure 3 tracks the accuracy reward on the training set. The three variants start from the same point and separate monotonically throughout training, yielding a consistent ordering OPDVR > OPD > Inverse-Gated in both the training curves and the final benchmarks. Reversing the gate falls below vanilla OPD on all six benchmarks, confirming the effectiveness of the gating mechanism. Notably, even the inverse-gated variant still improves over the initial model, indicating that the teacher's distributional guidance remains useful—though its benefits are substantially hindered when the sign is misaligned with trajectory correctness

Table 4: Ablation study on the gating mechanism. We compare OPD, OPDVR, and an inverse-gated variant where the ReLU gate is applied in the opposite direction.

Method	AIME24	AIME25	AMC	MATH500	Minerva	OlympiadBench	Avg.
Student (Qwen3-4B)	24.0	15.8	60.8	80.9	27.6	42.9	42.0
Teacher (Qwen3-4B-RL)	36.0	29.0	65.9	87.0	35.4	49.3	50.4
OPD	34.2	26.0	63.1	85.5	31.6	46.5	47.8
OPDVR (Ours)	36.9	28.1	64.8	84.7	33.2	47.0	49.1
Inverse-Gated	30.3	21.2	62.3	83.6	27.7	42.8	44.6



(a) Training dynamic of same-architecture distillation (Qwen3-4B ← Qwen3-4B-RL).



(b) Training dynamic of cross-architecture distillation (Qwen3-1.7B-Base ← Qwen3-4B-Base-RL).

Figure 4: Training dynamics

5.5. Training Dynamics

To understand how token-level gating shapes the distillation process, we track three quantities throughout training: the student policy entropy, the average response length, and the zero-gated token ratio, i.e., the fraction of tokens whose distillation loss is masked by the gate. Figure 4 shows the dynamics of the same-architecture setting (Qwen3-4B ← Qwen3-4B-RL) and the cross-architecture setting (Qwen3-1.7B-Base ← Qwen3-4B-Base-RL), respectively.

The entropy and response-length trajectories of the sampled-token OPD baseline show no consistent pattern across settings; instead, they depend strongly on the teacher–student pair. In the same-architecture setting, the entropy of both methods drifts mildly upward (from ∼0.33 to ∼0.40) while the response length inflates by more than four times, from roughly 1.6k to over 6.7k tokens. In the cross-architecture setting, the picture inverts: the student entropy collapses rapidly from ∼2.0 at the start of training and the response length stays nearly flat throughout. These quantities therefore do not follow a setting-independent trend – their evolution is dictated by the specific teacher and student models rather than by the distillation objective itself.

In contrast, the zero-gated token ratio is consistent across both settings. It remains in a band around fifty percent during the entire course of training (≈0.48–0.50 for the 4B student and ≈0.40–0.44 for the 1.7B student), never degenerating toward the trivial extremes of gating all or no tokens. This means that roughly half of the sampled tokens are zeroed by the ReLU gate at any point of training. Recalling the failure cases identified in Section 4, these are exactly the tokens whose implicit OPD reward pushes against the RLVR direction: on correct trajectories where the student already assigns higher probability than the teacher ($\pi_\theta > \pi_T$), and on incorrect trajectories where the teacher still outranks the student ($\pi_T > \pi_\theta$). The stable ratio indicates that such RLVR-violating tokens constitute a persistent fraction of the data throughout training – the gate consistently filters them out while distilling the remaining tokens with their full teacher–student log-ratio magnitude.

6. Conclusion

We revisited sampled-token OPD from an RLVR perspective and observed that its implicit reward is governed by the teacher-student probability ratio rather than trajectory correctness, which can lead to updates that can penalize tokens in correct trajectories or reward tokens in incorrect ones. To address this, we proposed OPDVR, which adds a simple ReLU gate on the sampled token to enforce RLVR compliance. This minimal modification combines OPD and RLVR without any hyperparameter or heuristic trade-off. Experiments across mathematical reasoning benchmarks demonstrate that OPDVR consistently outperforms standard OPD.

More importantly, by reformulating OPD as an RLVR method, OPDVR becomes a general framework that can be readily integrated with any policy gradient algorithm, including REINFORCE, GRPO, and DAPO. We instantiate this with GRPO and present Group Relative Policy Distillation (GRPD), which further demonstrates the versatility and strength of our approach. The empirical success of both OPDVR and GRPD across multiple benchmarks confirms that aligning distillation signals with trajectory correctness is not only effective but also broadly applicable, offering a simple yet powerful recipe for future post-training methods.

References

Cai, Q., Ma, Y., Li, L., Li, P., Chen, Y., Guo, Q., Zou, Y., Gui, T., Feng, X., and Qin, B. H2 sd: Hybrid hindsight self-distillation. *arXiv preprint arXiv:2607.18955*, 2026.

Fu, Y., Huang, H., Jiang, K., Liu, J., Jiang, Z., Zhu, Y., and Zhao, D. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hammoud, H. A. A. K., Alhamoud, K., Hammoud, A., Bou-Zeid, E., Ghassemi, M., and Ghanem, B. Train long, think short: Curriculum learning for efficient reasoning. *arXiv preprint arXiv:2508.08940*, 2025.

He, Z., Liang, T., Xu, J., Liu, Q., Chen, X., Wang, Y., Song, L., Yu, D., Liang, Z., Wang, W., et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026.

Hou, B., Zhang, Y., Ji, J., Liu, Y., Qian, K., Andreas, J., and Chang, S. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Hou, W., Peng, S., Wang, W., Ruan, Z., Zhang, Y., Zhou, Z., Gao, M., Chen, Y., Wang, K., Yang, H., et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Huang, C., Zhang, Z., and Cardie, C. Hapo: Training language models to reason concisely via history-aware policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31122–31130, 2026.

Hubotter, J., Lubeck, F., Behric, L. D., Baumann, A., Bagatella, M., Marta, D., Hakimi, I., Shenfeld, I., Buening, T. K., Guestrin, C., and Krause, A. Reinforcement learning via self-distillation. *ArXiv*, abs/2601.20802, 2026. URL https://api.semanticscholar.org/CorpusID:285102353.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jiang, Y., Li, Y., Chen, G., Liu, D., Cheng, Y., and Shao, J. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

Jin, W., Min, T., Yang, Y., Wei, D., Zhou, Y., Kadhe, S. R., Baracaldo, N., and Lee, K. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Li, G., Yang, T., Fang, J., Song, M., Zheng, M., Guo, H., Zhang, D., Wang, J., and Chua, T.-S. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Li, Y., Zuo, Y., He, B., Zhang, J., Xiao, C., Qian, C., Yu, T., Gao, H.-a., Yang, W., Liu, Z., et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Liu, W., Zhou, R., Deng, Y., Huang, Y., Liu, J., Deng, Y., Zhang, Y., and He, J. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025.

Oh, M., Song, S., Choi, G., Choi, Y., and Jo, Y. Kl for a kl: On-policy distillation with control variate baseline. *arXiv preprint arXiv:2605.07865*, 2026.

Petrenko, A., Lipkin, B., Chen, K., Wijmans, E., Cusumano-Towner, M., Giryes, R., and Krähenbühl, P. Entropy-preserving reinforcement learning. *arXiv preprint arXiv:2603.11682*, 2026.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Su, Z., Pan, L., Lv, M., Li, Y., Hu, W., Zhang, F., Gai, K., and Zhou, G. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *ArXiv*, abs/2509.20712, 2025. URL https://api.semanticscholar.org/CorpusID:281526201.

Sui, Y., Chuang, Y.-N., Wang, G., Zhang, J., Zhang, T., Yuan, J., Liu, H., Wen, A., Zhong, S., Zou, N., et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Wang, C., Li, Z., Bai, J., Zhang, Y., Deng, H., Lan, G., and Wang, Y. Distilled reinforcement learning for llm post-training. *arXiv preprint arXiv:2607.17247*, 2026.

Xiang, V., Blagden, C., Rafailov, R., Lile, N., Truong, S., Finn, C., and Haber, N. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

Xiao, B., Xia, B., Yang, B., Gao, B., Shen, B., Zhang, C., He, C., Lou, C., Luo, F., Wang, G., et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Xing, X., Wang, H., Gao, B., Li, Z., and Tang, Y. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Xu, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., Ling, C., et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yang, C., Qin, C., Si, Q., Chen, M., Gu, N., Yao, D., Lin, Z., Wang, W., Wang, J., and Duan, N. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026a.

Yang, K., Xu, X., Chen, Y., Liu, W., Lyu, J., Lin, Z., Ye, D., and Yang, S. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control. *arXiv preprint arXiv:2511.15248*, 2025b.

Yang, W., Liu, W., Xie, R., Yang, K., Yang, S., and Lin, Y. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026b.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Zeng, A., Lv, X., Hou, Z., Du, Z., Zheng, Q., Chen, B., Yin, D., Ge, C., Huang, C., Xie, C., et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

A. Theoretical Analysis: Why OPDVR Improves over Sampled-Token OPD

We now provide a formal characterization of the advantage of OPDVR over the sampled-token OPD objective. Throughout this section, let $s_t = (q, o_{<t})$ denote the state at step t, $a_t = o_t$ the sampled action, and define the token-level teacher-student log-ratio as

$$r_t = \log \frac{\pi_T(a_t \mid s_t)}{\pi_\theta(a_t \mid s_t)}. \tag{3}$$

As before, $R \in \{+1, -1\}$ denotes the trajectory-level verifier reward.

A.1. Directional Alignment with the Verifier Gradient

Let $u_t = \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ denote the policy gradient direction of the sampled token. The token-level update direction of sampled-token OPD is

$$\Delta_{\mathrm{OPD},t} = r_t u_t, \tag{4}$$

whereas the OPDVR update direction is

$$\Delta_{\mathrm{OPDVR},t} = R\,\mathrm{ReLU}(Rr_t)\,u_t, \tag{5}$$

with $\mathrm{ReLU}(x) = \max(0, x)$.

Recall that the pure verifier-driven RLVR update direction is $\Delta_{\mathrm{RLVR},t} = Ru_t$. The following proposition shows that OPDVR is always aligned with this verifier direction, while OPD may be anti-aligned.

Proposition A.1 (Verifier alignment)**.** *For any token with $u_t \neq 0$, the OPDVR update is never in the opposite direction of the verifier gradient. Specifically,*

$$\langle \Delta_{\mathrm{OPDVR},t}, \Delta_{\mathrm{RLVR},t} \rangle \geq 0. \tag{6}$$

In contrast, sampled-token OPD satisfies

$$\langle \Delta_{\mathrm{OPD},t}, \Delta_{\mathrm{RLVR},t} \rangle = r_t R \|u_t\|^2, \tag{7}$$

which becomes negative whenever r_t and R have opposite signs.

Proof. Using (5) and $\Delta_{\mathrm{RLVR},t} = Ru_t$, we have

$$\begin{aligned} \langle \Delta_{\mathrm{OPDVR},t}, \Delta_{\mathrm{RLVR},t} \rangle &= \langle R\,\mathrm{ReLU}(Rr_t)u_t, Ru_t \rangle \\ &= R^2\,\mathrm{ReLU}(Rr_t)\,\|u_t\|^2 \\ &= \mathrm{ReLU}(Rr_t)\,\|u_t\|^2 \geq 0. \end{aligned} \tag{8}$$

For sampled-token OPD, using (4) gives

$$\langle \Delta_{\mathrm{OPD},t}, \Delta_{\mathrm{RLVR},t} \rangle = \langle r_t u_t, Ru_t \rangle = r_t R \|u_t\|^2, \tag{9}$$

which is negative when $r_t R < 0$. □

The condition $r_t R < 0$ corresponds exactly to the two conflict cases identified earlier: (i) a correct trajectory with $\pi_\theta > \pi_T$, and (ii) an incorrect trajectory with $\pi_T > \pi_\theta$. In both cases, OPD pushes the policy in a direction that reduces the verifier reward.

A.2. Decomposition of the OPD Gradient

The gated mechanism can be interpreted as explicitly removing the verifier-conflicting component from the OPD gradient.

Proposition A.2 (Conflict removal)**.** *The sampled-token OPD update can be decomposed as*

$$\Delta_{\mathrm{OPD},t} = \Delta_{\mathrm{OPDVR},t} + \Delta_{\mathrm{conflict},t}, \tag{10}$$

where

$$\Delta_{\mathrm{conflict},t} = r_t\,\mathbf{1}(r_t R < 0)\,u_t. \tag{11}$$

Moreover, the conflict term always has a non-positive projection onto the verifier gradient:

$$\langle \Delta_{\mathrm{conflict},t}, \Delta_{\mathrm{RLVR},t} \rangle = -|r_t|\,\mathbf{1}(r_t R < 0)\,\|u_t\|^2 \leq 0. \tag{12}$$

Proof. When $r_t R \geq 0$, we have $R\,\mathrm{ReLU}(Rr_t) = r_t$, so $\Delta_{\mathrm{OPDVR},t} = r_t u_t$ and the conflict term vanishes. When $r_t R < 0$, we have $\mathrm{ReLU}(Rr_t) = 0$, so $\Delta_{\mathrm{OPDVR},t} = 0$ and the conflict term equals $\Delta_{\mathrm{OPD},t}$. The projection onto $\Delta_{\mathrm{RLVR},t}$ follows directly. □

Thus OPDVR is not an arbitrary modification of OPD; it is precisely OPD with the harmful verifier-opposing component removed.

A.3. A Simplified Token-Level Analysis: OPDVR Can Strictly Outperform the Teacher

To illustrate the mechanism while retaining the token-level structure of LLM generation, we consider a simplified single-token decision problem. Let s be a fixed prefix. At this prefix, the model must choose between two candidate next tokens: a_1 denotes a correct key token that leads to a correct final answer, and a_2 denotes an incorrect key token that leads to a wrong final answer. The trajectory-level verifier reward is therefore $R(a_1) = +1$ and $R(a_2) = -1$.

Let $\pi_\theta(a_1 \mid s) = q$ and $\pi_T(a_1 \mid s) = p$. Assume the teacher is suboptimal, i.e., $p < \frac{1}{2}$, and that the initial student policy is already better than the teacher, i.e., $q_0 > p$.

Under sampled-token OPD, the expected loss reduces to the reverse KL divergence

$$\mathcal{L}_{\mathrm{OPD}} = D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s)), \tag{13}$$

whose global minimizer is $\pi_\theta(\cdot \mid s) = \pi_T(\cdot \mid s)$. Hence the OPD optimum satisfies $q = p$, with expected verifier reward

$$J_{\mathrm{OPD}} = 2p - 1. \tag{14}$$

Under OPDVR, the expected update for token a_1 is

$$R(a_1)\,\mathrm{ReLU}\left(R(a_1)\log\frac{p}{q_0}\right) = \mathrm{ReLU}\left(\log\frac{p}{q_0}\right) = 0, \tag{15}$$

because $\log(p/q_0) < 0$ since $q_0 > p$. Similarly, for token a_2 we have $R(a_2) = -1$ and

$$\log\frac{1-p}{1-q_0} > 0 \tag{16}$$

since $q_0 > p$, so the OPDVR update for a_2 is

$$R(a_2)\,\mathrm{ReLU}\left(R(a_2)\log\frac{1-p}{1-q_0}\right) = -\mathrm{ReLU}\left(-\log\frac{1-p}{1-q_0}\right) = 0. \tag{17}$$

Thus the expected OPDVR gradient vanishes, and the policy remains at q_0. Consequently,

$$J_{\mathrm{OPDVR}} = 2q_0 - 1 > 2p - 1 = J_{\mathrm{OPD}} = J_{\mathrm{teacher}}. \tag{18}$$

Therefore, OPDVR strictly outperforms both standard sampled-token OPD and the teacher policy.

B. Hyperparameters

We provide the detailed hyperparameter configurations used in our experiments in Table 5. All models are trained using the **Verl** (Sheng et al., 2025) framework with the settings specified below.

Table 5: Hyperparameter settings.

Hyperparameter	Value
Learning Rate	1e-6
Train Batch Size	256
PPO Mini-Batch Size	256
Max Response Length	8192
Max Prompt Length	1024
Rollout Temperature	1.0
Evaluation Temperature	0.7
Evaluation Top-p	0.95

C. Hardware Setup

All experiments in this paper are conducted on NVIDIA GeForce RTX 5090 GPUs.