Distill Where You Fail: Recovering Learning Signals of Negative RL-Groups from Adaptive Teacher Guidance

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a standard paradigm for post-training large language models (LLMs). While Group Relative Policy Optimization (GRPO) is widely adopted, it suffers from sparse reward signals and loses gradients entirely when all responses within a group receive identical rewards. On-policy distillation (OPD) offers a natural remedy by providing dense, token-level supervision from a teacher model. However, naively combining GRPO with OPD leads to degraded performance, due to three underlying causes: not all samples benefit from distillation; fitting too quickly to the teacher undermines the exploratory capacity of RL; and OPD's advantages are asymmetric, suppressing most tokens. To address these challenges, we propose **RSTG** (**R**ecovering Learning **S**ignals via Adaptive **T**eacher **G**uidance), which applies distillation selectively and precisely where it matters most. At the sample level, OPD is restricted to negative zero-variance prompts with each sample weighted by the teacher's confidence score. At the token level, distillation targets only tokens with high student entropy or large teacher-student divergence. We further augment training with SFT on correct trajectories generated by the teacher model, injecting positive gradient signals where RL yields none. Experiments demonstrate that RSTG substantially outperforms naive GRPO+OPD by +4.02% on math and +3.05% on code.

1 Introduction

Post-training large language models via reinforcement learning with verifiable rewards (RLVR) has emerged as a standard approach for improving reasoning capabilities (Team, 2025b; DeepSeek-AI, 2025; Team, 2025a). Among RLVR methods, Group Relative Policy Optimization (GRPO; Shao et al., 2024) is widely adopted for its simplicity and stability. GRPO normalizes outcome rewards across a group of rollouts to estimate a scalar advantage applied uniformly to every token, resulting in sparse rewards (Li et al., 2026c) and vanishing gradients when all rollouts are correct or incorrect; we refer to such prompts as *positive* and *negative zero-variance prompts*, respectively (Zheng et al., 2025; Feng et al., 2025).

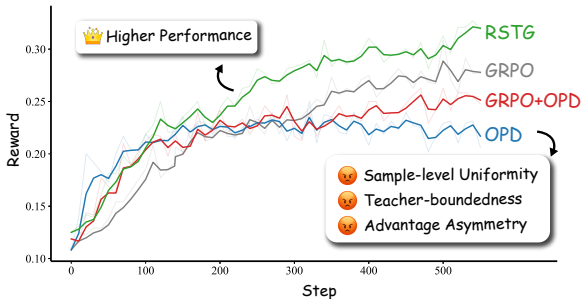

Figure 1: **Training Dynamics on MATH.** Naively GRPO + OPD proves ineffective, while ours RSTG achieves higher performance.

Recent work has shifted toward on-policy distillation (OPD; Agarwal et al., 2024; Gu et al., 2024), which has the student generate its own rollouts and leverages the teacher's per-token log-probabilities as a dense reward signal, naturally addressing the limitations of GRPO. A straightforward approach is to complement GRPO with OPD by summing their losses (Agarwal et al., 2024). However, as shown in Figure 1, this naive combination underperforms standard GRPO, and even with an annealing schedule on the OPD coefficient (Xu et al., 2025), the results remain unsatisfactory.

We identify three key reasons why this naive combination fails: **(1) Sample-level Uniformity:** Not all samples benefit from OPD; the quality of the teacher's guidance depends on its proficiency on the given sample. **(2) Teacher-boundedness:** OPD converges quickly with performance capped at the teacher's level. Premature convergence to the teacher severely undermines RL's exploratory

capacity. **(3) Advantage Asymmetry:** Student-generated tokens are typically assigned low probability by the teacher, causing most token-level advantages to be negative (Fu et al., 2026; Jia et al., 2026; Ko et al., 2026; Lu et al., 2026), suppressing the learning signal. Furthermore, OPD is inherently local, conditioning on a potentially erroneous student-generated prefix and thus producing unreliable gradients (Fu et al., 2026).

To address the aforementioned limitations, we propose RSTG (**R**ecovering Learning **S**ignals via Adaptive **T**eacher **G**uidance). **(1) Sample Selection.** Applying OPD exclusively on prompts where the student fails but the teacher succeeds outperforms standard OPD on all samples, using only 3.63% of the full data. We therefore apply OPD solely on *negative zero-variance prompts* (Le et al., 2025; Zheng et al., 2025) and weight the distillation signal by the teacher's degree of mastery over each prompt. **(2) Token Selection.** We restrict OPD gradient updates to tokens where the student exhibits high entropy or where the teacher-student divergence is large, slowing convergence and reducing gradient noise. **(3) Auxiliary SFT.** On negative zero-variance prompts, we perform SFT on correct trajectories pre-generated by the teacher, injecting positive gradient signals and providing a global perspective to alleviate the locality of OPD.

We validate RSTG across the Qwen2.5 and Qwen3 model families on mathematical and code benchmarks. RSTG achieves substantial improvements over naive GRPO+OPD ($+4.02\%$ on mathematics, $+3.05\%$ on code), while also mitigating advantage asymmetry, slowing convergence toward the teacher, and preventing abrupt response length inflation.

In summary, our contributions are as follows:

- We identify the applicable scope of OPD and demonstrate the impact of teacher proficiency on its effectiveness.
- We propose RSTG, enabling effective integration of GRPO and OPD.
- We analyze advantage asymmetry and reveal that it is less pronounced in the code domain.

2 Related Work

On-Policy Distillation (OPD). Knowledge distillation (Hinton et al., 2015) transfers capabilities from a teacher to a student model. OPD (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2026) samples trajectories from the student and aligns it with the teacher's token-level logit distribution, providing dense on-policy supervision that naturally complements GRPO. Two dominant paradigms exist: GKD-style OPD uses the token-level KL divergence directly as the training loss (Agarwal et al., 2024), while PG-style OPD treats the per-token reverse KL as a dense reward signal and updates the student via policy gradient, with the advantage defined as the negation of the reverse KL (Ko et al., 2026; Yang et al., 2026a). We adopt the latter, which unifies naturally with the RL framework (Yang et al., 2026b).

Combining RL and OPD. Prior work has explored integrating RL with knowledge distillation to jointly leverage reward signals and teacher supervision. GKD (Agarwal et al., 2024) first explores this unification for text summarization, and KDRL (Xu et al., 2025) further extends this direction. More recently, a growing body of work has investigated combining RL with on-policy self-distillation (OPSD) (Yang et al., 2026a; Li et al., 2026a; Lu et al., 2026). However, this paradigm still faces significant challenges, and existing methods tend to fail across a broader range of models and settings. RSTG aims to make this combination more robust and effective.

3 Preliminaries

Let D denote the input distribution, and let $\pi_{\boldsymbol{\theta}}$ and π^* denote the student and teacher policies, respectively. We unify the following three methods under a common reinforcement learning framework.

3.1 Group Relative Policy Optimization

The RL objective can be formulated as

$$\mathcal{J}_{\mathrm{RL}}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim D, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|x)} r(\boldsymbol{x}, \boldsymbol{y}), \quad (1)$$

where the trajectories $\boldsymbol{y}$ are sampled from the current policy model $\pi_{\boldsymbol{\theta}}$, $r(\boldsymbol{x}, \boldsymbol{y})$ is the reward function that measures the quality of a response sequence $\boldsymbol{y} = (y_1, \cdots, y_T)$ to a query $\boldsymbol{x}$. A common approach to solving this objective is to apply policy gradient, updating the policy parameters using an estimated gradient of the form:

$$\nabla_{\boldsymbol{\theta}} \mathcal{J}_{\mathrm{RL}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \left[\sum_{t=1}^{T} A_t \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} \left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t} \right) \right], \quad (2)$$

where A_t is the relative advantage of token y_t over a baseline value. Traditional methods such as PPO estimate A_t via a learned critic, introducing substantial overhead. Group Relative Policy Optimization (GRPO) obviates the need for an additional value function approximator as required in PPO. Instead, it uses the average reward of multiple sampled outputs generated in response to the same question as the baseline. Formally, let $\pi_{\boldsymbol{\theta}}$ sample G responses $\{y_1, y_2, \cdots, y_G\}$ for each prompt x. The optimization objective of GRPO with token-level loss and without the KL penalty term is:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\boldsymbol{\theta}) =& \mathbb{E}_{\boldsymbol{x}\sim D, \{\boldsymbol{y}^{(i)}\}_{i=1}^{G}\sim\pi_{\boldsymbol{\theta}_{\text{old}}}(\cdot|\boldsymbol{x})} \\ & \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(r_{i,t}(\boldsymbol{\theta})A_{i,t},\right. \\ & \left.\text{clip}(r_{i,t}(\boldsymbol{\theta}), 1-\varepsilon, 1+\varepsilon)A_{i,t}\right)\Bigg],\end{aligned} \tag{3}$$

where $r_{i,t}(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(y_{i,t}|x, y_{i,<t})}{\pi_{\boldsymbol{\theta}_{\text{old}}}(y_{i,t}|x, y_{i,<t})}$, the advantage is given by $A_{i,t} = \frac{r(\boldsymbol{x},\boldsymbol{y}^{(j)}) - \text{mean}\left(\{r(\boldsymbol{x},\boldsymbol{y}^{(j)})\}_{j=1}^{G}\right)}{\text{std}\left(\{r(\boldsymbol{x},\boldsymbol{y}^{(j)})\}_{j=1}^{G}\right)}$, and ε is a clipping hyperparameter.

3.2 On-Policy Distillation

The main idea of OPD is to let the student generate its own trajectories, and then minimize the reverse KL divergence between the student and the teacher on those student-generated trajectories:

$$\begin{aligned}\mathcal{J}_{\text{OPD}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim D,\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \\ \left[\mathcal{D}_{\text{KL}}(\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi^*(\boldsymbol{y} \mid \boldsymbol{x}))\right].\end{aligned} \tag{4}$$

Then, we can get the gradient of OPD as:

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\text{OPD}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D,\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \\ &\Bigg[\sum_{t=1}^{T}\left(\log\pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right) - \log\pi^*\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right) \\ &\quad \cdot\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\Bigg].\end{aligned} \tag{5}$$

Eq. (17) takes the same form as Eq. (2), unifying OPD within the RL framework, where $A_t^{\text{OPD}} = -\left(\log\pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right) - \log\pi^*\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right)$ serves as the advantage in OPD, enabling token-level credit assignment. The detailed derivation is provided in Appendix A.1.

3.3 Supervised Fine-Tuning

Supervised Fine-Tuning (SFT) trains the student policy $\pi_{\boldsymbol{\theta}}$ to mimic the teacher π^* using demonstration data. From an information-theoretic perspective, SFT minimizes the forward KL divergence between the teacher and the student, which contrasts with the reverse KL divergence used in OPD:

$$\begin{aligned}\mathcal{J}_{\text{SFT}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim D} \\ \left[\mathcal{D}_{\text{KL}}(\pi^*(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}))\right].\end{aligned} \tag{6}$$

Ignoring the entropy of π^* (independent of $\boldsymbol{\theta}$), this is equivalent to maximizing the expected log-likelihood of teacher trajectories:

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim D, \boldsymbol{y}\sim\pi^*(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{T}\log\pi_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})\right]. \tag{7}$$

Taking the gradient with respect to $\boldsymbol{\theta}$ yields:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\text{SFT}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D,\, \boldsymbol{y}\sim\pi^*(\cdot|\boldsymbol{x})} \\ \left[\sum_{t=1}^{T} 1\cdot\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right].\end{aligned} \tag{8}$$

Comparing Eq. (8) with the general policy gradient formulation in Eq. (2), SFT can be elegantly unified within the RL framework as an *off-policy* algorithm (Zhu et al., 2026; Wu et al., 2025). Specifically, the trajectories $\boldsymbol{y}$ are sampled from the teacher policy π^* rather than the active student policy $\pi_{\boldsymbol{\theta}}$, and the advantage function is implicitly set to a constant $A_t^{\text{SFT}} = 1$. This implies that every token generated by the teacher is treated as a gold standard, receiving a uniform, positive credit.

4 Method

The framework of RSTG is illustrated in Figure 2. When a negative zero-variance prompt is encountered, the RSTG branch is activated; otherwise, the standard GRPO branch is followed. Within RSTG, we weight the OPD advantages according to the teacher's confidence on the given prompt (§4.1), restrict OPD gradient updates to high-quality tokens (§4.2), and perform SFT using reference answers generated by the teacher model (§4.3).

4.1 Data Selection and Teacher-Guided Advantage Weighting

To study the effects of data difficulty and teacher proficiency on OPD, we partition the full 57k-sample dataset $\mathcal{D}$ into two nested subsets based on

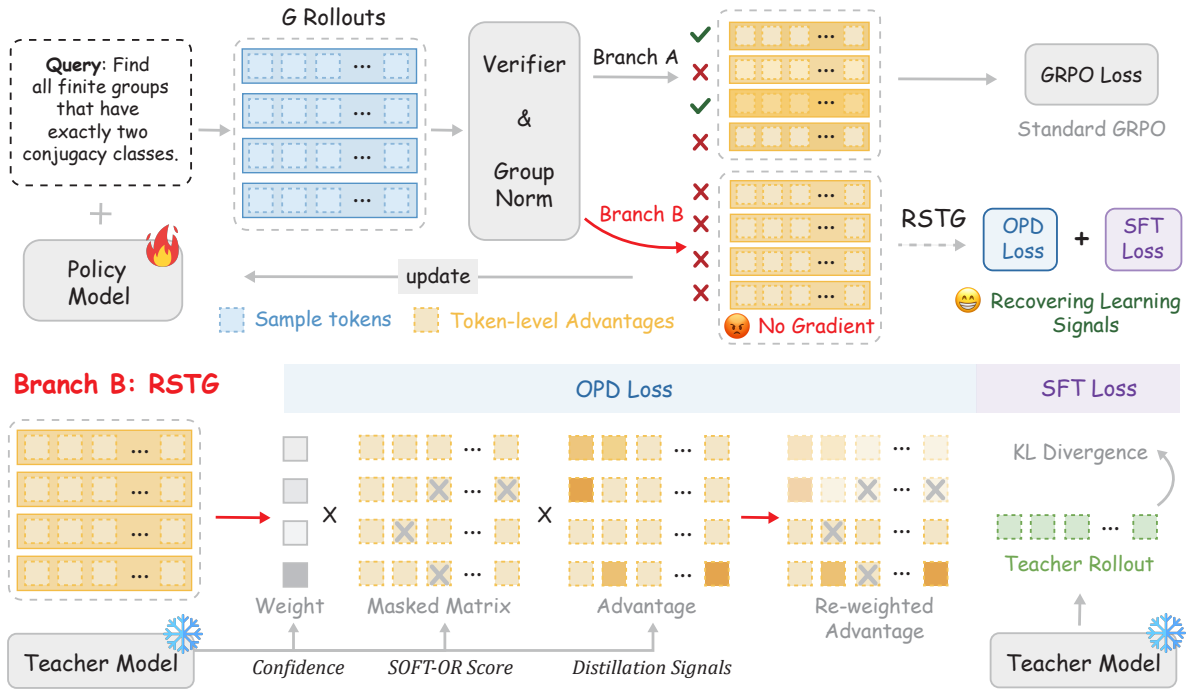

Figure 2: **Illustration of RSTG framework.** For negative zero-variance prompts, RSTG is applied; otherwise, standard GRPO is used. RSTG weights OPD advantages by the teacher's confidence, restricts gradient updates to high-quality tokens, and performs SFT on teacher-generated reference answers.

model performance: (1) $\mathcal{D}_{\text{sw}}$ (9k samples), where the student fails all 8 rollouts (mean@8 $= 0$), and (2) $\mathcal{D}_{\text{swtr}}$ (2k samples), a subset of $\mathcal{D}_{\text{sw}}$ where the teacher achieves a perfect success rate (mean@8 $= 1$). We evaluate OPD across these three data scales on mathematical benchmarks (Figure 3). The results yield two key insights:

Finding 1. Training on tasks challenging for the student ($\mathcal{D}_{\text{sw}}$) outperforms training on the full dataset ($\mathcal{D}$).

Finding 2. Restricting training to samples where the teacher excels ($\mathcal{D}_{\text{swtr}}$) yields the best performance using only 3.63% of $\mathcal{D}$.

This indicates that the complementary effect of OPD on RL should be channeled specifically into this category of samples, which naturally correspond to *negative zero-variance prompts* in GRPO, where all rollouts fail and the advantage collapses to zero, leaving no training signal. OPD effectively recovers these lost gradient signals by providing token-level advantages. Furthermore, we incorporate teacher confidence as a fine-grained weighting mechanism into this hybrid objective.

Formally, for sample i, let $\omega_i \in [0, 1]$ denote the teacher's mean@8 score, serving as a proxy

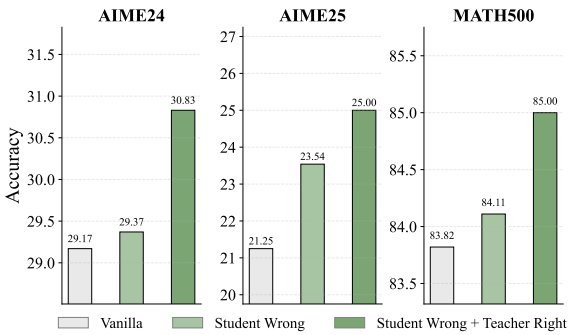

Figure 3: Performance on the mathmatical benchmark for Qwen3-1.7B-Instruct trained via OPD on different data partitions: Vanilla $\mathcal{D}$, Student Wrong $\mathcal{D}_{\text{sw}}$, Student Wrong and Teacher Right $\mathcal{D}_{\text{swtr}}$.

for teacher confidence to dynamically scale the token-level OPD advantage. The hybrid advantage function is defined as:

$$A_{i,t}^{\text{Hybrid}} = \begin{cases} \beta \cdot \omega_i \cdot A_{i,t}^{\text{OPD}}, & \text{if } r_j = 0,\ \forall j \in G \\ A_{i,t}^{\text{GRPO}}, & \text{otherwise} \end{cases} \tag{9}$$

where $A_{i,t}^{\text{GRPO}}$ denotes the standard GRPO advantage, $A_{i,t}^{\text{OPD}}$ the OPD advantage, and β a scaling hyperparameter controlling the magnitude of the OPD signal. The weighting coefficient ω_i assigns

greater optimization weight to tokens where the teacher is more confident, leading to more reliable policy optimization.

4.2 Mitigating Premature Convergence via Token Selection

We select high-quality tokens to slow the student's convergence toward the teacher and reduce gradient noise. Inspired by TIP (Xu et al., 2026), we identify two categories of valuable tokens: (1) **high-entropy tokens**, where the student is uncertain, typically representing critical reasoning junctures; and (2) **tokens with large teacher-student discrepancy**, which tend to carry richer information.

We define the student's entropy at position t as:

$$h_t = -\sum_{v} p_{t,v} \log p_{t,v}, \tag{10}$$

where $p_{t,v}$ is the student's predicted probability for vocabulary token v at position t. The teacher-student discrepancy is defined as:

$$d_t = |A_{i,t}^{\text{OPD}}|. \tag{11}$$

Tokens with large h_t or d_t are considered high-value. We combine these two criteria via the Soft-OR formulation (see Appendix A.2 for details). Given min-max normalized $\hat{h}_t, \hat{d}_t \in [0, 1]$, the selection score is:

$$s_t = \hat{h}_t + \hat{d}_t - \hat{h}_t \cdot \hat{d}_t. \tag{12}$$

The token-masked OPD advantage is then defined as:

$$\hat{A}_{i,t}^{\text{OPD}} = \begin{cases} A_{i,t}^{\text{OPD}}, & \text{if } t \in \mathcal{S}_k \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\mathcal{S}_k$ denotes the top-$k\%$ tokens ranked by s_t, enabling gradual distillation of the teacher's capabilities while reducing gradient noise.

4.3 Complementary Supervision on Negative Zero-Variance Prompts

This advantage asymmetry is further exacerbated on negative zero-variance prompts. As illustrated in Figure 4, the advantage values on negative zero-variance prompts are consistently more negative than those on positive zero-variance prompts throughout training, reflecting a severely suppressive optimization signal that penalizes the student without providing any constructive guidance toward correct solutions.

Figure 4: OPD advantage curves during training on positive and negative zero-variance prompts.

To mitigate this advantage asymmetry of OPD, we augment the training objective for negative zero-variance prompts with an auxiliary SFT loss computed on correct trajectories pre-generated out by the teacher π^*. Formally, the final training objective is defined as:

$$\mathcal{J}_{\text{Final}}(\boldsymbol{\theta}) = \begin{cases} \mathcal{J}_{\text{GRPO}}\left(\boldsymbol{\theta}; \beta \cdot \omega_i \cdot \hat{A}_{i,t}^{\text{OPD}}\right) \\ +\beta \cdot \mathcal{J}_{\text{SFT}}(\boldsymbol{\theta}), \quad \text{if } r_j = 0, \ \forall j \in G \\ \mathcal{J}_{\text{GRPO}}\left(\boldsymbol{\theta}; A_{i,t}^{\text{GRPO}}\right), \quad \text{otherwise} \end{cases} \tag{14}$$

where $\mathcal{J}_{\text{SFT}}$ is computed exclusively on negative zero-variance prompts, and β is shared with the OPD term in Eq. (9), ensuring consistent regularization across both components.

As established in Section 3.3, this objective is equivalent to assigning a uniform positive advantage $A_t^{\text{SFT}} = 1$ to every token in the teacher trajectory, which injects purely positive gradient signals into the optimization on prompts where RL yields no learning signal, thereby shifting the OPD advantage space toward the positive direction, alleviating the advantage symmetry issue, while simultaneously introducing a global perspective signal and guiding the student toward correct solution trajectories demonstrated by the teacher.

5 Experiments

5.1 Setup

Models We employ three teacher-student model pairs across two model families:

- **Qwen3** (Team, 2025b): Qwen3-1.7B-Instruct → Qwen3-4B-Instruct-2507 (Pair ❶); Qwen3-4B-Instruct → Qwen3-4B-Instruct-2507 (Pair ❷).
- **Qwen2.5** (Yang et al., 2024): Qwen2.5-3B-Instruct → Qwen2.5-14B-Instruct (Pair ❸).

Table 1: Main results on MATH and CODE benchmarks across three Student-Teacher model pairs. **Best** and second-best are highlighted.

Method	MATH					CODE		
	AIME24	AIME25	MATH500	OLMPIAD	AVG	APPS	MBPP+	AVG
Student: Qwen3-1.7B-Instruct Teacher: Qwen3-4B-Instruct-2507								
Vanilla	12.08	10.83	72.66	40.26	39.96	30.80	49.22	40.01
OPD	32.29	20.62	84.32	53.46	47.67	43.65	57.98	50.82
GRPO	34.79	27.71	88.64	55.15	51.57	50.39	69.66	60.03
GRPO+OPD	35.21	26.88	88.27	55.13	51.37	57.03	72.06	64.55
ReLIFT	35.21	30.42	88.65	55.79	52.52	60.17	60.98	60.58
RL-ZVP	31.87	25.62	86.99	53.46	49.49	49.73	70.12	59.93
RSTG	**42.98**	**31.87**	**89.36**	**57.36**	**55.39**	**61.81**	**72.41**	**67.11**
Student: Qwen3-4B-Instruct Teacher: Qwen3-4B-Instruct-2507								
Vanilla	24.38	18.33	83.84	52.20	44.69	43.93	74.22	59.08
OPD	50.83	40.63	91.93	63.71	61.78	52.70	75.39	64.05
GRPO	54.37	45.00	94.29	66.08	64.94	65.29	77.82	71.56
GRPO+OPD	50.42	41.87	93.95	**67.25**	63.37	73.46	84.53	79.00
ReLIFT	**58.13**	42.71	94.74	64.61	65.05	**76.12**	85.41	80.77
RL-ZVP	50.42	42.71	93.54	63.41	62.52	68.16	82.98	75.57
RSTG	57.08	**47.92**	**95.30**	67.24	**66.89**	75.68	**88.42**	**82.05**
Student: Qwen2.5-3B-Instruct Teacher: Qwen2.5-14B-Instruct								
Vanilla	4.58	1.25	62.14	27.04	23.75	16.60	62.84	39.72
OPD	6.67	1.46	63.56	28.98	25.17	25.83	65.05	45.44
GRPO	6.04	3.33	**68.16**	32.09	27.41	37.18	66.63	51.91
GRPO+OPD	6.88	4.58	67.68	31.95	27.77	49.72	73.37	61.55
ReLIFT	6.46	2.71	67.44	32.05	27.17	52.60	71.50	62.05
RL-ZVP	6.46	3.33	67.24	31.72	27.19	42.03	71.01	57.52
RSTG	**8.33**	**5.42**	68.14	**32.74**	**28.66**	**53.01**	**74.03**	**63.52**

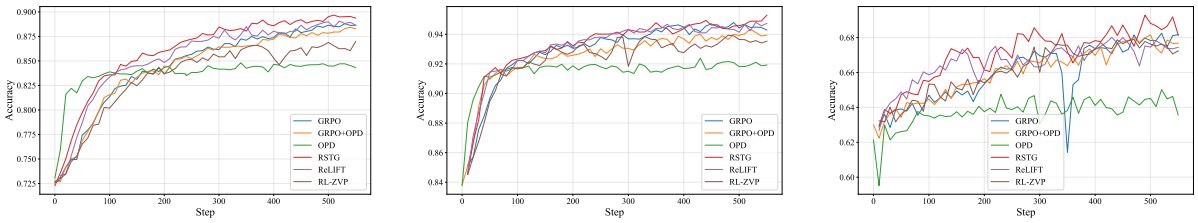

Figure 5: Performance on MATH500 for Qwen3-1.7B-Instruct → Qwen3-4B-Instruct-2507, Qwen3-4B-Instruct → Qwen3-4B-Instruct-2507, and Qwen2.5-3B-Instruct → Qwen2.5-14B-Instruct from left to right, respectively.

Dataset and Metrics. For training, we filter the DeepMath (He et al., 2025) dataset to select 57K samples with difficulty $\geq$ 6 as math RL data, and use Eurus-RL-Code (Cui et al., 2025) (25K samples) as code RL data. For evaluation, we use AIME 2024 (MAA, 2024), AIME 2025 (MAA, 2025), MATH-500 (Gao et al., 2025), and OLMPIAD Bench (He et al., 2024) for mathematical reasoning, and APPS (Hendrycks et al., 2021) and MBPP+ (Liu et al., 2023) for code generation. We set temperature to 1.0, top-p to 1.0, and maximum generation length to 8,192, sampling 16 solutions per math problem and 4 per code problem. Further details are in Appendix A.3.

Implementation Details. We set the batch size to 256, maximum response length to 8192, number of rollouts to 8, and learning rate to 1×10^{-6}, with thinking mode disabled for all models. For mathematics, training runs for 550 steps (222 steps/epoch); for code, 400 steps (98 steps/epoch), both reaching convergence. The coefficient β in Eq. 14 is linearly annealed with $\beta_{\text{init}} = 5 \times 10^{-3}$, $\delta = 5 \times 10^{-5}$, and $\beta_{\text{min}} = 1 \times 10^{-3}$. To construct the SFT dataset, we pre-sample $n = 8$ responses per prompt from the teacher, retaining the shortest correct response as reference. More details and cost analysis are provided in Appendix A.5.

Baselines We compare our method against five baselines: GRPO, OPD, **GRPO+OPD**, which directly combines the GRPO and OPD losses following the same configuration as KDRL (Xu et al., 2025). See Appendix A.4 for more details. Since our method specifically targets negative zero-variance prompts, we also include two representative baselines from the line of work on learning from negatives: **ReLIFT** (Ma et al., 2025), which applies SFT using ground-truth answers on negative zero-variance prompts, and **RL-ZVP** (Le et al., 2025), which designs an asymmetric advantage formulation based on token-level entropy for both positive and negative zero-variance prompts to extract effective learning signals.

5.2 Main Results

Table 1 presents the performance of all methods across four mathematical reasoning benchmarks and two code generation benchmarks.

For mathematics, naive GRPO+OPD tends to slightly underperform standard GRPO, with changes of -0.2%, -1.57%, and $+0.36\%$ across the three model pairs. In contrast, RSTG achieves substantial improvements over naive GRPO+OPD, with gains of $+4.02\%$, $+3.52\%$, and $+0.89\%$ respectively, surpassing nearly all baselines.

For code generation, naive GRPO+OPD yields notable improvements over standard GRPO, with gains of $+4.52\%$, $+7.44\%$, and $+9.64\%$, potentially attributable to the absence of advantage asymmetry in the code domain (5.3). Nevertheless, RSTG achieves consistent further improvements over naive GRPO+OPD, with gains of $+2.56\%$, $+3.05\%$, and $+1.97\%$ across the three model pairs, surpassing almost all baselines.

Figure 5 illustrates the training dynamics of all methods on MATH500 as a representative benchmark, where RSTG outperforms nearly all baselines at every training step. Notably, for Model Pair 2, where both the teacher and student are 4B models with a relatively small capability gap, the performance gains are less pronounced compared to the other two model pairs. More complete training curves are provided in the Appendix A.6.

5.3 The Symmetry of Advantage

The asymmetry of advantages is alleviated. We compared the evolution of advantages during training between RSTG and naive GRPO+OPD. As shown in Figure 6, RSTG yields higher advantage values and alleviates the advantage asymmetry introduced by OPD, thereby allowing more tokens to receive positive learning signals.

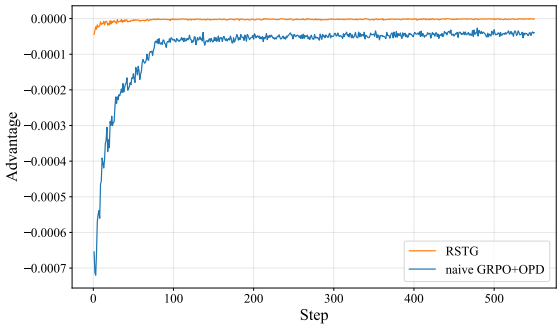

Figure 6: Comparison of advantage values during training between RSTG and naive GRPO+OPD, using model pair 2 as a representative example.

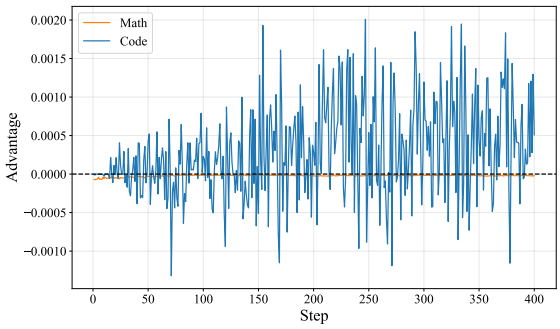

Figure 7: Comparison of advantage values during RSTG training on the mathematics and code benchmarks, using model pair 3 as a representative example.

Advantage asymmetry is less pronounced in the code domain. As shown in Figure 7, advantages of RSTG in the code domain fluctuate around zero and become increasingly positive over training, whereas in mathematics, despite mitigating the asymmetry, the average advantage remains negative. This suggests that advantage asymmetry is inherently less prevalent in the code domain. We attribute this to the low-entropy nature of code: a large proportion of tokens are deterministic, including keywords (`for`, `if`, `return`), syntactic symbols, and standard library calls, leaving a far smaller effective token choice space than natural language. This constrains both teacher and student to similar distributions at most positions, such that student-generated tokens retain high probability under the teacher, keeping advantages roughly balanced. This also explains why naive GRPO+OPD outperforms standard GRPO in the code domain (Table 1): with weaker asymmetry, the OPD distillation signal takes effect without being over-

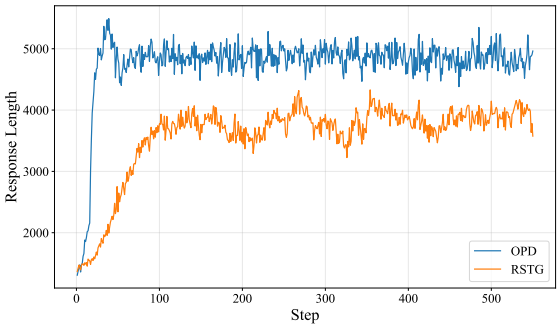

Figure 8: Comparison of response length during training between RSTG and OPD.

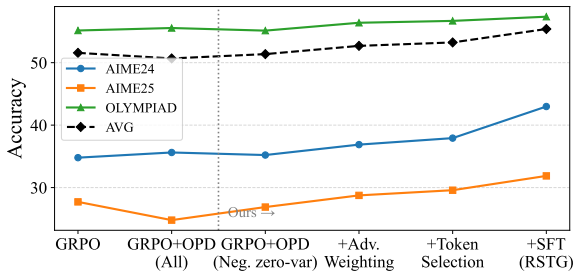

Figure 9: Performance on mathematical benchmarks as components of RSTG are progressively added. MATH500 is excluded from the figure due to its larger score range, but is included in the AVG computation.

whelmed by negative advantages.

5.4 Convergence Speed and Response Length

Convergence Speed. We use the overlap M_{overlap} between the top-k tokens ($k = 16$) of the teacher and the student (Li et al., 2026b) to quantify the alignment between their candidate spaces, where a higher value indicates that the student more closely fits the teacher. We monitor this metric to track the rate at which the student converges to the teacher. At step 200, standard OPD reaches 69.7%, naive GRPO+OPD reaches 67.6%, while RSTG achieves only 65.81%, demonstrating that RSTG slows convergence toward the teacher and preserves a larger exploration space for RL.

Response Length. OPD suffers from abrupt length inflation (Luo et al., 2026; Fu et al., 2026), as illustrated in Figure 8. Our RSTG effectively mitigates this issue, maintaining response length comparable to that of GRPO.

5.5 Ablation Studies

Each component plays a crucial role. Taking model pair1 as an example, we start from a baseline that naively combines the GRPO and OPD losses over all samples, and progressively incorporate each component of our RSTG. As shown in Figure 9, each component contributes meaningfully to the overall performance. Detailed results are provided in the Appendix A.7.

Table 2: Average accuracy on mathematical benchmarks under different β strategies. Best values are **bolded**.

Method	AIME24	AIME25	MATH500	OLMPIAD	AVG
$\beta_{\text{init}} = 5 \times 10^{-3}$	35.21	**26.88**	**88.27**	**55.13**	**51.37**
$\beta_{\text{init}} = 1 \times 10^{-2}$	**36.04**	26.46	87.35	54.01	50.97
$\beta = 5 \times 10^{-3}$	35.00	25.21	87.40	53.32	50.23
$\beta = 5 \times 10^{-2}$	34.79	24.58	86.55	52.18	49.53
$\beta = 1 \times 10^{-1}$	32.08	24.38	85.81	51.51	48.45

The coefficient β of OPD and SFT We investigate two strategies for β: constant and linear annealing. For the constant strategy, we evaluate several fixed values of β; for the linear annealing strategy, we evaluate several different initial values β_{init}. In both cases, OPD is applied solely on negative zero-variance prompts without any additional components. Results are reported in Table 2.

A constant β consistently degrades performance, as the performance upper bound of OPD is lower than that of GRPO. A large β causes the model to overfit to the teacher and converge prematurely to this lower bound. The linear annealing strategy suffers from the same issue when the initial value is too large, while too small an initial value results in a distillation signal too weak to be effective. We therefore adopted $\beta_{\text{init}} = 5 \times 10^{-3}$ as our final choice. Since the SFT intensity is aligned with that of OPD, both terms share the same coefficient β.

6 Conclusion

To address the performance degradation caused by the naive combination of GRPO and OPD, we have propose RSTG, which selectively applies OPD on difficult samples that the student has not yet mastered, weighted by the teacher's degree of proficiency. Furthermore, RSTG employs fine-grained token selection to slow down convergence and reduce gradient noise, and incorporates an auxiliary SFT objective to inject positive gradient signals and provide a global perspective on the solution space. Experiments across benchmarks in two domains and three model pairs confirm consistent improvements over both standard GRPO and naive GRPO+OPD baselines.

Limitations

We conduct experiments across two domains, math and code, which demonstrate the generalizability of our method. Nevertheless, we look forward to exploring its applicability in other settings, such as agentic tasks. Due to computational constraints and the strict requirement of OPD for a meaningful capability gap between the teacher and student, our experiments are conducted at the largest scale feasible within our resources. We expect RSTG to remain effective at larger model scales, and leave this for future work.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, 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, and 4 others. 2025. Process Reinforcement through Implicit Rewards. *CoRR*, abs/2502.01456.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948.

Yunzhen Feng, Parag Jain, Anthony Hartshorn, Yaqi Duan, and Julia Kempe. 2025. Don't Waste Mistakes: Leveraging Negative RL-Groups via Confidence Reweighting. *CoRR*, abs/2510.08696.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. 2026. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2025. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark for Large Language Models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3828–3850. Association for Computational Linguistics.

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. 2025. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. *CoRR*, abs/2504.11456.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021. Measuring Coding Challenge Competence With APPS. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. 2026. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. 2026. Scaling Reasoning Efficiently via Relaxed On-Policy Distillation. *CoRR*, abs/2603.11137.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. 2025. No Prompt Left Behind: Exploiting Zero-Variance Prompts in LLM Reinforcement Learning via Entropy-Guided Advantage Shaping. *CoRR*, abs/2509.21880.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. 2026a. Unifying Group-Relative and Self-Distillation Policy Optimization via Sample Routing. *CoRR*, abs/2604.02288.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-Ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. 2026b. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *CoRR*, abs/2604.13016.

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. 2026c. Outcome-Grounded Advantage Reshaping for Fine-Grained Credit Assignment in Mathematical Reasoning. *CoRR*, abs/2601.07408.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. 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: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and 1 others. 2026. Self-Distilled Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.15155*.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. 2026. Demystifying OPD: Length Inflation and Stabilization Strategies for Large Language Models. *CoRR*, abs/2604.08527.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, and Wentao Zhang. 2025. Learning What Reinforcement Learning Can't: Interleaved Online Fine-Tuning for Hardest Questions. *CoRR*, abs/2506.07527.

MAA. 2024. American invitational mathematics examination - aime 2024.

MAA. 2025. American invitational mathematics examination - aime 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *CoRR*, abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Gemini Team. 2025a. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *CoRR*, abs/2507.06261.

Qwen Team. 2025b. Qwen3 Technical Report. *CoRR*, abs/2505.09388.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2025. On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification. *CoRR*, abs/2508.05629.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. 2025. KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning. *CoRR*, abs/2506.02208.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. 2026a. Self-Distilled RLVR. *CoRR*, abs/2604.03128.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026b. Learning beyond Teacher: Generalized On-Policy Distillation with Reward Extrapolation. *CoRR*, abs/2602.12125.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2025. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via Selective Rollouts. *CoRR*, abs/2506.02177.

Wenhong Zhu, Ruobing Xie, Rui Wang, and Pengfei Liu. 2026. Hybrid Policy Distillation for LLMs. *CoRR*, abs/2604.20244.

A Appendix

A.1 Preliminaries

The main idea of OPD is to let the student generate its own trajectories, and then minimize the reverse KL divergence between the student and the teacher on those student-generated trajectories:

$$\mathcal{J}_{\mathrm{OPD}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim D,\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \left[\mathcal{D}_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi^*(\boldsymbol{y} \mid \boldsymbol{x}))\right]. \tag{15}$$

Expanding the KL divergence and applying the chain rule of probability, we have:

$$\begin{aligned}\mathcal{J}_{\mathrm{OPD}}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x},\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\left[\log \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})}{\pi^*(\boldsymbol{y} \mid \boldsymbol{x})}\right] \\ &= \mathbb{E}_{\boldsymbol{x},\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=1}^{T} \log \frac{\pi_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})}{\pi^*(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})}\right].\end{aligned} \tag{16}$$

Taking the gradient with respect to $\boldsymbol{\theta}$ and applying the log-derivative trick $\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}[f(\boldsymbol{y})] = \mathbb{E}_{\boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}}[f(\boldsymbol{y})\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y})]$:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{OPD}}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x}\sim D,\, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \\ &\left[\sum_{t=1}^{T} \left(\log \pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right) - \log \pi^*\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right)\right. \\ &\left.\cdot \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right].\end{aligned} \tag{17}$$

Eq. (17) takes the same form as Eq. (2), unifying OPD within the RL framework, where $A_t^{\mathrm{OPD}} = -\left(\log \pi_{\boldsymbol{\theta}}\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right) - \log \pi^*\left(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}\right)\right)$ serves as the advantage in OPD, enabling token-level credit assignment.

A.2 Method

Mitigating Premature Convergence via Token Selection Tokens with large h_t or d_t are considered high-value. We combine these two criteria via the Soft-OR formulation, which approximates the logical OR operation in a differentiable manner: if either $\hat{h}_t$ or $\hat{d}_t$ is large, s_t will be large, while avoiding double-counting when both are large. Given min-max normalized $\hat{h}_t, \hat{d}_t \in [0, 1]$, the selection score is:

$$s_t = \hat{h}_t + \hat{d}_t - \hat{h}_t \cdot \hat{d}_t. \tag{18}$$

A.3 Experiment Details

The APPS dataset contains 10,000 samples in total (Hendrycks et al., 2021), spanning three difficulty levels: Introductory, Interview, and Competition. To accelerate evaluation, we uniformly sample 500 instances as our test set, with an equal number of samples drawn from each difficulty level.

A.4 Baselines

For the configuration of naive GRPO+OPD, we follow KDRL (Xu et al., 2025), which considers three variants: (1) applying OPD on all samples, (2) applying OPD only on samples where the student fails, and (3) applying OPD only on negative zero-variance prompts. For each model pair, we evaluate all three variants and select the best-performing one as the naive GRPO+OPD baseline. The OPD coefficient is kept consistent with our method.

A.5 Implementation Details.

We implement RSTG based on the VeRL framework (Sheng et al., 2025). For mathematics, we train for 550 steps on 8 A100 GPUs; the wall-clock time for Qwen3-1.7B-Instruct → Qwen3-4B-Instruct-2507, Qwen3-4B-Instruct → Qwen3-4B-Instruct-2507, and Qwen2.5-3B-Instruct → Qwen2.5-14B-Instruct is approximately 3, 4, and 2.5 days, respectively. For code, we train for 400 steps, taking approximately 4, 5, and 3 days for the three pairs, respectively.

Cost Analysis During GRPO training, taking Qwen2.5-3B-Instruct → Qwen2.5-14B-Instruct as an example, negative zero-variance prompts account for approximately 15.6% of steps initially, and this proportion decreases as the model improves. Since RSTG applies OPD and SFT exclusively on negative zero-variance prompts, the additional overhead over standard GRPO is minimal. On 8 A100 GPUs for 550 training steps, the extra cost amounts to approximately 12 hours, which is acceptable.

Furthermore, the SFT data is pre-generated offline and incurs no cost during training. This pre-generation process is efficient: for the 57K training set with a single rollout per prompt, it takes only about 2 hours on 2 A100 GPUs. Overall, the computational cost of RSTG is fully manageable.

A.6 Training Dynamics

We select one model pair from each of the Qwen3 and Qwen2.5 families and present their perfor-

Table 3: Ablation study results on mathematical reasoning benchmarks.

Method	AIME24	AIME25	MATH500	OLYMPIAD	AVG
GRPO	34.79	27.71	88.64	55.15	51.57
GRPO+OPD (All)	35.62	24.79	86.76	55.55	50.68
GRPO+OPD (Negative zero-var)	35.21	26.88	88.27	55.13	51.37
+Advantage Weighting	36.88	28.75	88.74	56.39	52.69
+Token Selection	37.92	29.58	88.75	56.69	53.24
+SFT (Ours)	**42.98**	**31.87**	**89.36**	**57.36**	**55.39**

mance on mathematical benchmarks throughout training in Figures 10 and 11.

A.7 Ablation Studies

Taking Qwen3-1.7B-Instruct→Qwen3-4B-Instruct as an example, we start from a baseline that naively combines the GRPO and OPD losses over all samples, and progressively incorporate each component of our RSTG. Detailed results are provided in Table 3, where each component is shown to contribute meaningfully to the overall performance.

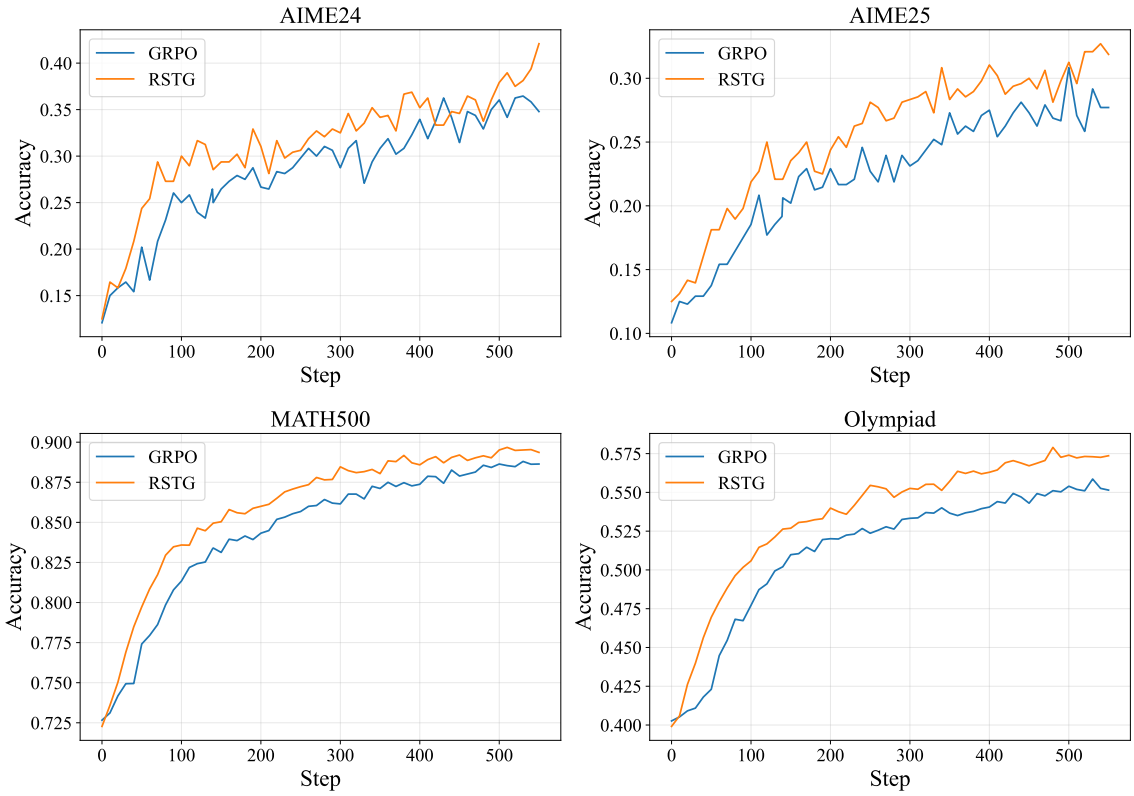

Figure 10: Performance of Qwen3-1.7B-Instruct → Qwen3-4B-Instruct-2507 on mathematical benchmarks throughout training, compared against standard GRPO.

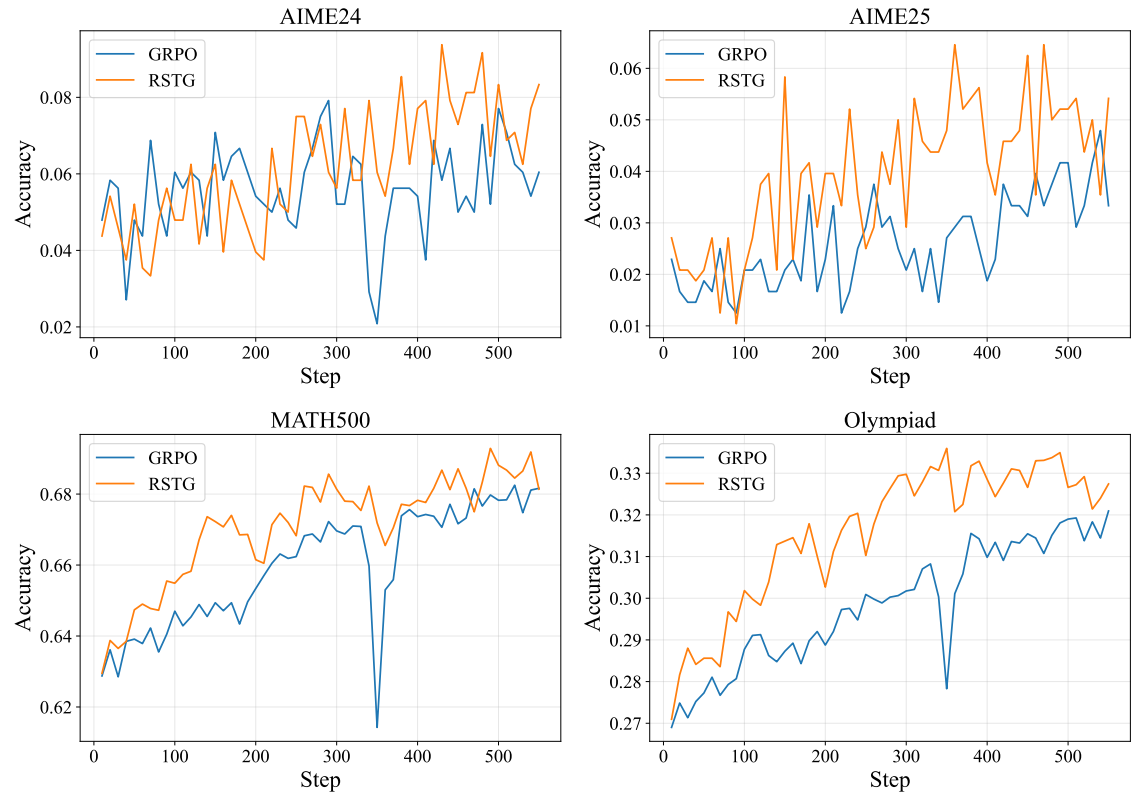

Figure 11: Performance of Qwen2.5-3B-Instruct → Qwen2.5-14B-Instruct-2507 on mathematical benchmarks throughout training, compared against standard GRPO.