Teacher-Guided Policy Optimization for On-Policy Reasoning Distillation under Large Policy Divergence

Anonymous ACL submission

Abstract

On-policy distillation (OPD) has become a promising paradigm for reasoning-oriented post-training of large language models (LLMs), especially when combined with reinforcement learning from verifiable rewards (RLVR). Existing OPD methods rely on reverse KL (RKL)-based teacher supervision over trajectories sampled from the student policy. However, we identify a critical limitation: under large teacher–student policy divergence, RL-driven exploration often produces trajectories outside the teacher distribution, resulting in uninformative negative feedback. To address this, we propose Teacher-Guided Policy Optimization (TGPO), an on-policy reasoning distillation method that remains effective under large policy divergence settings. Rather than relying solely on evaluative supervision, TGPO uses teacher to directly guide token level generation conditioning on student-generated contexts; together with RLVR-style trajectory level rewards, TGPO steers exploration toward improved continuations. Experiments on reasoning benchmarks show that TGPO consistently outperforms existing RKL-based OPD methods and remains robust across different teacher models.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) (Team et al., 2025; Guo et al., 2025) and knowledge distillation (Team, 2025; Xiao et al., 2026) are two widely used approaches for improving the reasoning abilities of LLMs. RLVR enables scalable optimization from verifiable outcomes, but its reward signals are sparse and uniformly applied across all generated tokens, providing limited fine-grained feedback. In contrast, knowledge distillation offers dense token-level supervision from a teacher model but relies on off-policy data. Recently, **on-policy distillation (OPD)** (Agarwal et al., 2024; Lu and Lab, 2025; Xu et al., 2025) has emerged as a promising paradigm that combines the advantages of both approaches. Unlike conventional teacher-forced distillation, OPD trains the student on trajectories sampled from its own policy while leveraging teacher supervision signals. By aligning training with the student-induced distribution, OPD alleviates the mismatch between training and inference and naturally complements RLVR-based reasoning optimization.

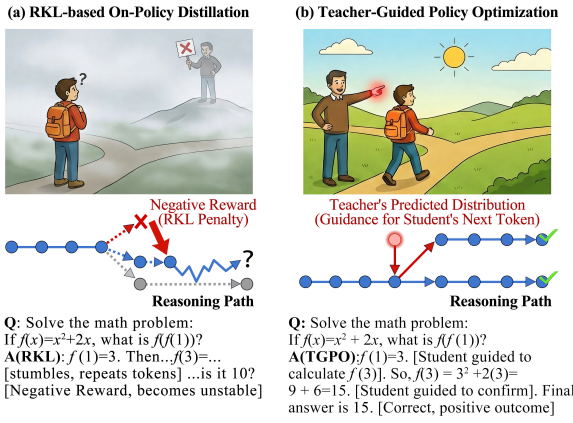

Figure 1: **RKL vs. TGPO. (a)** RKL relies on scalar rewards to penalize deviation. When the policy gap is significant, these penalties fail to provide directional information. **(b)** TGPO utilizes the teacher's predicted distribution as **guidance**, explicitly informing the student *what* to generate next rather than *what not* to generate.

Most existing OPD methods formulate teacher supervision through reverse KL (RKL)-based objectives. As detailed in Section 2, such objectives mainly provide *evaluative* supervision, rewarding trajectories preferred by the teacher while penalizing unlikely ones. In practice, existing OPD methods often reduce teacher–student policy divergence before applying on-policy distillation. For example, prior work constructs teacher–student pairs within the same model family (Agarwal et al., 2024), employs self-teaching strategies (Hübotter et al., 2026; Zhao et al., 2026), or introduces additional intermediate training stages to increase distribution overlap (Lu and Lab, 2025; Xu et al., 2025). These

design choices suggest that RKL-based OPD methods implicitly rely on sufficient overlap between teacher and student trajectory distributions for effective supervision.

However, we argue that this dependence on distribution overlap reflects a fundamental limitation of RKL-based supervision: *the teacher primarily evaluates sampled trajectories and does not provide explicit guidance toward better continuations*. As a result, the student must rely on exploration to discover teacher-preferred trajectories. This limitation becomes more severe when the student policy drifts far from the teacher distribution, as we analyze in Section 2. In such cases, the teacher assigns near-zero probability to many student-generated tokens, causing optimization to be dominated by uninformative negative feedback rather than useful directional guidance. Such token-level penalties can further degrade the quality of sampled trajectories. When combined with RLVR optimization (e.g., GRPO) in reasoning-oriented post-training, this issue can destabilize optimization, as sampled groups become increasingly dominated by poor trajectories (Le et al., 2025).

To address these limitations, we propose Teacher-Guided Policy Optimization (TGPO), an on-policy reasoning distillation framework designed to provide informative supervision even under large teacher–student divergence. As illustrated in Figure 1, unlike RKL-based objectives, which evaluate the teacher's likelihood of the student's actions, TGPO queries the teacher for the optimal action conditioned on the student's generated context. By maximizing the likelihood of teacher-predicted tokens during RLVR, TGPO leverages the exploration benefits of on-policy sampling while retaining the constructive supervision of supervised learning. This mechanism enriches traditional on-policy RL with fine-grained token-level supervision, bridging the gap between sparse outcome rewards and dense teacher guidance. Based on this perspective, we make the following contributions:

- We analyze the limitations of RKL-based OPD and empirically show that its effectiveness depends on sufficient teacher–student distribution overlap.
- We propose TGPO, an on-policy reasoning distillation framework that provides token-level teacher guidance on student-generated trajectories, enabling effective supervision under large teacher–student divergence.
- Experiments on reasoning benchmarks show that TGPO improves the robustness of OPD under large teacher–student divergence, even outperforming the mixed-policy approach.

2 RKL Limitations in LLM Distillation

In this section, we analyze the limitations of prior RKL-based on-policy distillation methods. We first formulate the RKL objective in Section 2.1, then show why RKL-based supervision becomes unstable under large teacher–student distribution divergence in Section 2.2. Finally, Section 2.3 provides empirical evidence supporting the analysis.

2.1 RKL-Based On-Policy Distillation

Given a prompt dataset $\mathcal{D} = \{x\}$, we aim to train a student policy $\pi_\theta(\cdot|x)$ to approximate a fixed, superior teacher policy $\pi_T(\cdot|x)$. On-policy distillation (OPD) (Gu et al., 2023; Team et al., 2024; Agarwal et al., 2024) achieve this by minimizing the RKL divergence over student-generated responses y:

$$\begin{aligned}\mathcal{J}_{\mathrm{RKL}}(\theta) &= \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta||\pi_T) \\ &= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y|x)}{\pi_T(y|x)}\right]. \end{aligned} \quad (1)$$

Unlike Forward KL or supervised fine-tuning, which rely on teacher-generated samples, the RKL objective takes expectations over responses sampled from the student policy itself. This on-policy formulation shares the same expectation structure as RL objectives, where optimization is also performed over trajectories sampled from the current policy. As a result, RKL-based distillation can be naturally interpreted within an RL framework.

Let $r(y)$ denote the reward assigned to a sampled sequence y. Standard RL objectives can then be written as:

$$\mathcal{J}_{\mathrm{RL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[r(y)\right]. \quad (2)$$

Comparing Eq. 1 and Eq. 2, minimizing $\mathcal{J}_{\mathrm{RKL}}$ is equivalent to maximizing $\mathcal{J}_{\mathrm{RL}}$ with intrinsic reward $r(y) = -\log \frac{\pi_\theta(y|x)}{\pi_T(y|x)}$, enabling OPD to be naturally optimized within standard RL frameworks.

2.2 The Limitations of RKL-based Methods

Despite its simple formulation, the RKL objective introduces optimization challenges under large teacher–student distribution gaps. In this section, we analyze this issue by viewing RKL as an intrinsic reward within an RL framework, following

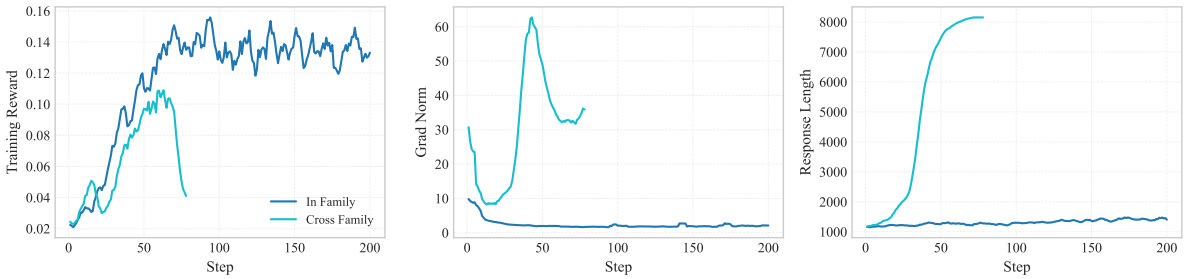

Figure 2: Comparison of RKL distillation dynamics. We distill a `Qwen2.5-Math-1.5B` student using either an In-Family teacher (`Qwen2.5-Math-7B`) or a Cross-Family teacher (`Qwen3-30B-A3B`). While the In-Family setting converges stably, the Cross-Family setting exhibits catastrophic instability, characterized by sharp training score degradation **(Left)**, gradient norm spikes **(Middle)**, and unbounded response length growth **(Right)**.

recent studies on RKL-based OPD (Xu et al., 2025; Lu and Lab, 2025).

Let $\rho(y) = \frac{\pi_\theta(y|x)}{\pi_T(y|x)}$ denote the density ratio. The intrinsic reward $-\log \rho(y)$ decreases monotonically with $\rho(y)$. Since trajectories y are sampled from the student policy $\pi_\theta(\cdot|x)$, they are concentrated in regions where the student already assigns high probability. Consequently, the optimization dynamics mainly fall into two regimes: [1]:

- $\rho(y) \approx 1$ (**Consensus**): The generated trajectories lie within the teacher's high-probability support. In this case, the density ratio is close to one, leading to a near-neutral intrinsic reward ($-\log \rho(y) \approx 0$).
- $\rho(y) \gg 1$ (**Rejection**): The student assigns high probability to trajectories that receive low probability under the teacher policy. This produces a large density ratio and a strong negative reward ($-\log \rho(y) \ll 0$).

In the Consensus regime, successful rollouts are naturally reinforced by the RL algorithm. However, in the Rejection regime, the teacher functions merely as a punitive critic, providing only negative scalar feedback without guidance toward better actions. As a result, the student must explore the large action space through inefficient trial-and-error, which often leads to optimization stagnation. As illustrated in Figure 1(a), the lack of directional correction makes escaping the low-reward region computationally intractable.

Beyond the lack of directional guidance, RKL-based objectives also exhibit asymmetry in reward scaling. While the density ratio $\rho(y)$ becomes unbounded from above when $\pi_T(y|x) \to 0$, it is lower-bounded by the student's own probability $\pi_\theta(y|x)$. Because trajectories are sampled from the student policy, the ratio rarely falls far below 1. As a result, negative penalties can dominate positive rewards by a large margin. This imbalance allows a single "bad" sample to produce gradients that overwhelm the accumulated positive signals from "good" samples, leading to unstable optimization. We provide a detailed analysis in Appendix A.

2.3 Empirical Validation

Based on our analysis, we conjecture that training stability and performance degrade when the student frequently generates trajectories with high density ratios ($\rho(y) \gg 1$). To validate this hypothesis, we train a `Qwen2.5-Math-1.5B` student under two configurations that induce different levels of teacher–student distributional shift[2]:

- **In-Family Distillation:** We use `Qwen2.5-Math-7B` as the teacher. Since the teacher and student belong to the same model family and share similar training distributions, the resulting distribution mismatch is relatively small.
- **Cross-Family Distillation:** We use `Qwen3-30B-A3B` as the teacher.[3] Compared to the student, this teacher exhibits different reasoning behaviors and output distributions, leading to a larger distribution mismatch.

Result. Figure 2 shows the training dynamics under the two settings. Although training uses only the intrinsic RKL reward, we report the average task reward on the training set to evaluate outcome correctness. The two settings exhibit markedly different behaviors. In the In-Family setting, the

[1] Because sampling is performed from π_θ, trajectories with $\pi_\theta(y|x) \ll \pi_T(y|x)$ (i.e., $\rho(y) \ll 1$) are rarely observed in practice. As a result, the student seldom receives strong positive rewards for trajectories favored by the teacher but not yet covered by the student policy.

[2] Detailed experimental settings and hyperparameters are provided in Appendix B.1.

[3] We use the reasoning-oriented "thinking" MoE model as a proxy for a strong general-purpose model that differs substantially from the specialized math student.

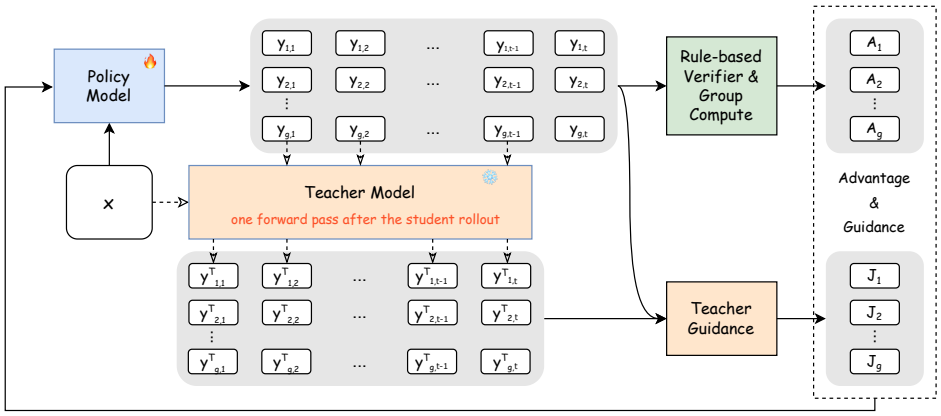

Figure 3: Overview of the TGPO. The Policy Model generates a group of rollouts $\{y_i\}_{i=1}^{g}$ conditioned on input x. At each step, the Teacher Model provides dynamic token-level guidance by predicting the optimal target token y^T based on the student's current prefix. This dense guidance signal (J) complements the outcome-based advantage (A) derived from the Rule-based Verifier to update the policy.

student converges steadily and achieves consistent improvements in task accuracy, indicating that RKL provides stable supervision when π_θ and π_T are initially well aligned. In contrast, training in the Cross-Family setting becomes highly unstable. Consistent with our analysis of the "Rejection" regime, we observe three failure modes: (1) *Performance Collapse*, where task rewards fail to improve consistently; (2) *Exploding Gradients*, where persistently large gradient norms suggest that unbounded penalties destabilize optimization; and (3) *Distributional Divergence*, where the student rapidly deviates from its initial distribution (e.g., pathological response length drift) after only $\sim$100 training steps. When combined with GRPO, these instabilities can further increase the likelihood of groups dominated by low-reward samples.

3 Teacher-Guided Policy Optimization

To address the limitations discussed in Section 2, we propose **Teacher-Guided Policy Optimization (TGPO)**, a new on-policy distillation algorithm for reasoning-oriented LLM training. Instead of using the teacher for evaluative supervision, TGPO reformulates teacher feedback as directional guidance for policy optimization. Combined with RLVR training, TGPO integrates teacher guidance more effectively while preserving the exploration benefits of RL-based optimization.

3.1 Guidance on Student Trajectories

Similar to RKL-based methods, our approach remains fully on-policy and relies only on trajectories sampled from the student policy π_θ. Given an input x, the student autoregressively generates a trajectory $y \sim \pi_\theta(\cdot \mid x)$, where each token y_t is sampled conditioned on the prefix $y_{<t}$.

To address the lack of corrective guidance in RKL, we introduce a teacher-guided objective defined on student-visited states. As illustrated in Figure 3, for each student prefix $y_{<t}$, we query the teacher policy and select its highest-probability next token: $y_t^T = \arg\max_{v \in \mathcal{V}} \pi_T(v \mid x, y_{<t})$, where $\mathcal{V}$ denotes the vocabulary. All teacher targets are computed from the generated trajectory in a single teacher forward pass, without iterative querying during decoding.

We then train the student to increase the likelihood of the teacher-preferred token at each visited state. The guidance objective $\mathcal{J}_{\text{G}}$ is defined as:

$$\mathcal{J}_{\text{G}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta}\left[-\sum_{t=1}^{|y|} \log \pi_\theta(y_t^T \mid y_{<t})\right].$$

Unlike the RKL objective, which only evaluates the student's sampled actions, our objective directly provides teacher-preferred continuations on the states visited by the student. This gives the student explicit guidance toward promising regions of the trajectory space, which may be difficult to discover through exploration alone.

Mechanistically, the objective resembles the teacher-forcing loss used in SFT. However, a key distinction lies in the trajectory distribution: our samples y are drawn from the student policy (π_θ) rather than the static ground truth. This ensures that the teacher's guidance is dynamic; it corrects the student based on the student's *actual* current

state, thereby mitigating the distribution shift and exposure bias issues associated with offline SFT.

3.2 Integrating Guidance into GRPO

Because the proposed guidance objective is fully on-policy, it can be naturally integrated into RLVR methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Given a query $x \sim \mathcal{D}$, the policy π_θ generates a group of G outputs $\{y_i\}_{i=1}^{G}$. Following recent work (Yu et al., 2025; He et al., 2025; Liu et al.), we omit the explicit KL regularization term with respect to a reference policy. The GRPO objective is defined as:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_\theta}\left[\frac{1}{Z}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\rho_{i,t}(\theta)A_i\right]$$

where $Z = \sum_i |y_i|$ normalizes by the total number of generated tokens, $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}$ denotes the importance sampling ratio, and $A_i = \frac{r_i-\mu}{\sigma}$ denotes the normalized group advantage.

Existing OPD methods typically combine distillation and RLVR signals through either reward shaping or differentiable teacher regularization. The main difference is whether teacher feedback is treated as a scalar reward on sampled trajectories or as a direct optimization target toward teacher-preferred continuations.

For TGPO, we adopt the latter formulation. Reward shaping is less suitable in our setting because GRPO updates the sampled student token $y_{i,t}$, while the guidance signal is defined on the teacher target token $y^T_{i,t}$. Using the guidance score as a scalar reward therefore introduces a mismatch between the optimized action and the supervised target. Instead, we directly optimize the likelihood of teacher targets conditioned on student-generated trajectories:

$$\mathcal{J}_{\text{TGPO}}(\theta) = \mathcal{J}_{\text{RL}}(\theta) + w\mathcal{J}_{\text{G}}(\theta), \qquad (3)$$

where w controls the strength of teacher guidance.

Strong guidance is useful in the early stage of training, but overly rigid supervision may later restrict exploration. To balance imitation and exploration, we linearly decay the guidance weight during training: $w_t = \max(w_{\text{init}} - \delta \cdot t, 0)$, where w_{init} is the initial guidance weight, t is the current training step, and δ is the decay rate. This schedule gradually shifts training from teacher-guided optimization toward pure reward-driven optimization. We use the annealed formulation as the default TGPO setting, and refer to the variant with a fixed guidance weight as **TGPO w/o annealing**.

4 Experimental Setup

Model and Dataset Construction. Following previous work (Yan et al., 2025; Liu et al.; Zeng et al., 2025), we adopt `Qwen2.5-Math-7B` (Yang et al., 2024) as our default base model. We adopt `Qwen3-30B-A3B` (Team, 2025) as the teacher model, aligning with the Cross-Family setting described in Section 2.3. We use `OpenR1-Math-46k-8192` (Yan et al., 2025), a subset of `OpenR1-Math-220k` (Hugging Face, 2025), as the training prompt set. To enable direct comparison with off-policy and mixed-policy methods, we sample teacher responses for `OpenR1-Math-46k-8192` and filter incorrect outputs using Math-Verify[4]. This process yields 35k prompts with corresponding off-policy reasoning traces.[5] We further evaluate TGPO with `Qwen2.5-Math-1.5B` as the student model to study performance under a larger teacher–student gap. Additional details are provided in Appendix C.

Benchmarks and Metrics. We assess performance across six widely-adopted mathematical reasoning benchmarks: AIME24, AIME25, AMC (Li et al., 2024), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and MATH500 (Hendrycks et al., 2021). For AIME24, AIME25, and AMC, we report **avg@32** due to their relatively small evaluation sets; for the remaining benchmarks, we use standard **pass@1**. To evaluate out-of-distribution generalization beyond mathematics, we additionally report results on ARC-c (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024) (denoted as GPQA*), and MMLU-Pro (Wang et al., 2024). During inference, we use a sampling temperature of 0.6. We also shuffle multiple-choice options to reduce position bias and mitigate potential data contamination.

Baseline Methods. We compare TGPO against both on-policy reasoning baselines and off-policy/mixed-policy methods. The on-policy baselines fall into two categories: RKL-based methods and pure RLVR methods. For RKL-based approaches, we include OP Distill (Lu and Lab, 2025), which uses the RKL log-ratio as the advantage signal, and KDRL (Xu et al., 2025), which

[4] https://github.com/huggingface/Math-Verify

[5] These traces are used only for off-policy and mixed-policy methods, while TGPO requires prompts only.

Model	In-Distribution Performance						Out-of-Distribution Performance			
	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	Avg.	ARC-c	GPQA*	MMLU-Pro	Avg.
Original Models										
Qwen2.5-Math-7B	11.5/4.9	31.3	43.6	7.4	15.6	19.0	18.2	11.1	16.9	15.4
Qwen3	59.5/49.8	85.3	96.0	52.9	68.0	68.6	94.1	65.2	80.0	79.8
Qwen3-8192	25.1/17.4	52.2	86.2	47.4	47.7	46.0	93.8	49.0	76.5	73.1
Off-Policy and Mixed-Policy Methods										
SFT	12.9/15.1	45.3	80.4	**42.3**	41.0	39.5	73.1	20.2	44.9	46.1
LUFFY	19.6/14.9	57.6	83.6	38.6	**51.9**	44.4	80.1	**38.9**	**50.1**	56.4
On-Policy Methods										
SimpleRL-Zero	27.0/6.8	54.9	76.0	25.0	34.7	37.4	30.2	23.2	34.5	29.3
PRIME-Zero	17.0/12.8	54.0	81.4	39.0	40.3	40.7	73.3	18.2	32.7	41.4
Oat-Zero	**33.4**/11.9	61.2	78.0	34.6	43.4	43.7	70.1	23.7	41.7	45.2
GRPO++	19.5/15.8	58.3	82.2	37.5	47.3	43.4	77.4	32.3	46.9	52.1
KDRL	17.2/14.4	55.8	83.6	36.0	43.4	41.7	78.4	35.4	46.9	53.6
OP Distill	5.7/4.5	29.9	64.0	23.2	27.1	25.7	26.1	6.1	23.0	18.4
TGPO w/o annealing	20.1/16.0	58.6	83.6	37.9	48.1	44.1	81.2	37.9	48.9	56.0
TGPO	21.1/**17.9**	**60.2**	**84.4**	40.4	49.8	**45.6**	**82.8**	37.4	**50.1**	**56.8**

Table 1: In-distribution and out-of-distribution performance based on Qwen2.5-Math-7B. We primarily benchmark against on-policy reasoning baselines, while also including off-policy and mixed-policy methods for comparison. The teacher model employed is Qwen3-30B-A3B (Qwen3); we additionally report its performance with a maximum generation length of 8192 tokens (Qwen3-8192). All models are evaluated under a unified setting. Bold indicates the best result, and underline indicates the second best (excluding the teacher model).

augments the GRPO objective with RKL regularization. We further compare against four pure RLVR variants: (1) SimpleRL-Zero, trained with standard rule-based rewards; (2) Oat-Zero (Liu et al.), which adopts Dr.GRPO for simplified advantage computation and loss normalization; (3) PRIME-Zero (Cui et al., 2025), which derives implicit process rewards from policy rollouts and outcome labels; and (4) GRPO++, which removes the explicit KL penalty and introduces token-level supervision. For completeness, we also report results for two off-policy or mixed-policy methods: (1) SFT, fine-tuned on teacher-sampled responses, and (2) LUFFY (Yan et al., 2025), which incorporates teacher-sampled trajectories as auxiliary supervision during RLVR training. Detailed training configurations are provided in Appendix B.1.

5 Experimental Results

5.1 Main Results

Table 1 reports results on both in-distribution (ID) math tasks and out-of-distribution (OOD) reasoning benchmarks. TGPO w/o annealing achieves better performance than on-policy methods, while remaining competitive with the strong mixed-policy baseline LUFFY. TGPO further improves over the no-annealing variant and achieves the best average performance on both ID and OOD benchmarks, highlighting the benefit of gradually annealing teacher guidance strength.

On ID benchmarks, TGPO improves over KDRL by 3.9 points (45.6 vs. 41.7). It also avoids the training collapse observed in OP Distill, indicating more stable optimization under on-policy exploration. Beyond on-policy distillation methods, TGPO also outperforms strong baselines from other training paradigms, including LUFFY (44.4) and the RLVR baseline GRPO++ (43.4). On OOD benchmarks, TGPO achieves the highest average score of 56.8, improving over SFT by 10.7 points (56.8 vs. 46.1). It also exceeds LUFFY on challenging reasoning tasks such as ARC-c (82.8 vs. 80.1), suggesting that teacher-guided on-policy training improves generalization to unseen reasoning tasks.

5.2 Training Dynamics and Stability Analysis

We analyze the training dynamics of TGPO and several baselines (GRPO++, KDRL, OP Distill, and LUFFY) using three metrics: training reward, response length, and gradient norm.

Figure 4 shows that TGPO maintains stable training dynamics compared to RKL-based methods (OP Distill and KDRL). Specifically, OP Distill exhibits early reward collapse (Left), severe response length explosion (Middle), and large gradient fluctuations (Right). When RKL-based supervision is combined with the RLVR framework, as in KDRL, the instability in response length and gradient norm is partially mitigated. However, its training reward remains lower than the pure RLVR baseline

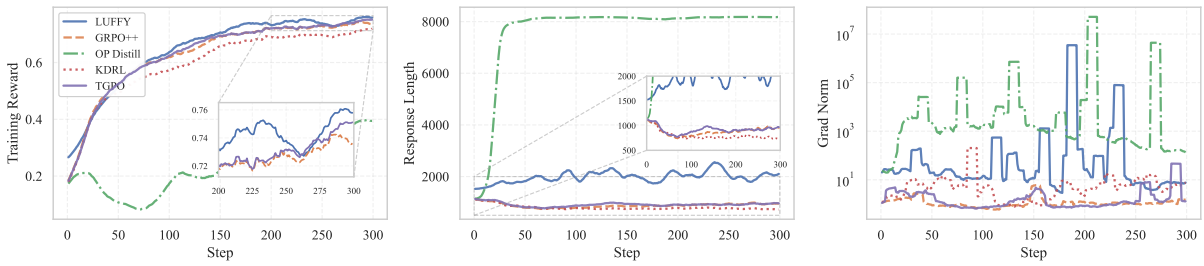

Figure 4: Training Dynamics Analysis. **(Left)** Training reward. TGPO demonstrates robust growth and convergence compared to RKL-based methods (i.e., KDRL, OP Distill). **(Middle)** Response length. TGPO avoids OP Distill's length explosion and aligns with GRPO++'s stability. **(Right)** Gradient norm. TGPO shows stable optimization compared to the high variance in OP Distill, KDRL and LUFFY.

Teacher Model	AMC	MATH	Olympiad	GPQA*	Avg.
No Teacher	58.3	82.2	47.3	32.3	55.0
R1-Distill-Qwen-32B	57.8	83.4	47.4	**40.9**	57.4
Qwen3-30B-A3B	**60.2**	**84.4**	**49.8**	37.4	**58.0**

Table 2: Ablation study on different teacher models. We compare the performance of TGPO when guided by different teacher policies with a no-teacher baseline.

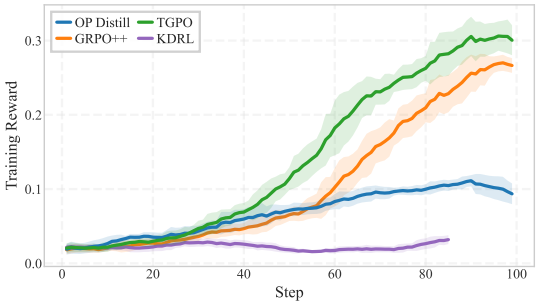

Figure 5: Training reward curves in the in-family setting. TGPO consistently achieves higher rewards than OP Distill, KDRL and GRPO++ throughout training.

GRPO++, suggesting that RKL-based OPD still negatively affects RLVR optimization. In contrast, TGPO converges stably with controlled response lengths while achieving stronger final benchmark performance than GRPO++, as shown in Table 1. These results suggest that TGPO effectively combines on-policy exploration with teacher supervision under large teacher–student divergence.

Finally, although LUFFY appears to achieve the highest training reward, this value is likely inflated by its strategy of including a ground-truth sample in each training group, which may also lead to response length instability and large gradient norm fluctuations observed in Figure 4.

5.3 TGPO with Different Teachers

To evaluate whether TGPO generalizes across different teacher models, we compare a baseline trained without teacher guidance (No Teacher) against TGPO variants guided by `R1-Distill-Qwen-32B` and `Qwen3-30B-A3B`. As shown in Table 2, incorporating teacher guidance consistently improves performance over the pure RLVR baseline. TGPO with `Qwen3-30B-A3B` achieves the best average accuracy (58.0%) and performs particularly well on mathematical benchmarks, including AMC, MATH, and Olympiad. In contrast, TGPO with `R1-Distill-Qwen-32B` obtains the strongest result on GPQA (40.9%). These results suggest that TGPO can effectively transfer the strengths of different teacher models and does not rely on a specific teacher architecture. We leave the exploration of a broader range of teacher models to future work.

5.4 Comparison in the In-family Setting

We further evaluate whether TGPO maintains its advantages in the in-family setting. Following the setup in Section 2, we use `Qwen2.5-Math-7B` to supervise `Qwen2.5-Math-1.5B` for 100 training steps. We compare the on-policy distillation methods OP Distill, KDRL, and TGPO with the pure RLVR baseline GRPO++ by tracking the training reward. As shown in Figure 5, OP Distill achieves stable reward improvements, indicating that RKL-based OPD remains effective when the teacher and student distributions are relatively aligned. GRPO++ obtains higher rewards, likely because outcome-based rewards better align with mathematical reasoning tasks. Although both OP Distill and GRPO++ improve steadily on their own, KDRL fails to converge in the in-family setting. In contrast, TGPO exhibits the fastest reward growth and consistently outperforms the other methods. These results show that TGPO remains effective beyond large-divergence regimes and generalizes well to in-family distillation scenarios.

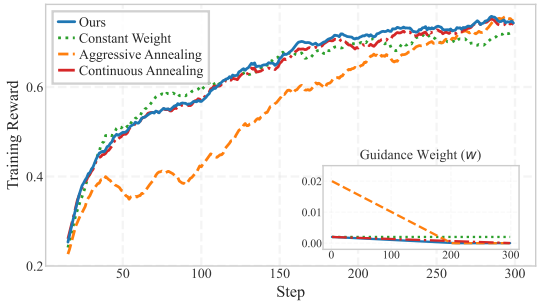

Figure 6: Ablation of annealing schedules. The inset details the guidance weight (w) schedule for each setting. Our method yields the best convergence.

5.5 Impact of Guidance Scheduling

To evaluate the guidance weight decay schedule introduced in Section 3.2, we compare our strategy with three alternatives that use different initial guidance weights w_{init} and decay rates δ: (1) Constant Weight ($w_{\text{init}} = 2e-3$, $\delta = 0$); (2) Aggressive Annealing ($w_{\text{init}} = 2e-2$, $\delta = 1e-4$); (3) Continuous Annealing ($w_{\text{init}} = 2e-3$, $\delta \approx 6.7e-6$, decaying to zero at the final training step); and (4) Ours ($w_{\text{init}} = 2e-3$, $\delta = 1e-5$, decaying to zero at step 200).

Figure 6 shows that our schedule achieves the best overall performance. Aggressive Annealing suppresses rewards early in training, indicating that overly strong guidance limits exploration. Constant Weight performs competitively at first but plateaus early, suggesting that persistent imitation constraints hinder further reward optimization. Our method also outperforms Continuous Annealing, indicating that entering a pure RL phase before the end of training is important for effective policy optimization. By removing teacher guidance at step 200, our method achieves the highest final reward.

6 Related Work

On-Policy Distillation. MiniLLM (Gu et al., 2023) first introduced on-policy distillation (OPD) by sampling directly from the student distribution using RKL supervision (Eq. 1). Concurrently, GKD (Agarwal et al., 2024) unified forward KL- and reverse KL-based distillation within a single framework and showed that OPD can be jointly optimized with RL objectives. Building on this line of work, KDRL (Xu et al., 2025) explored two ways to integrate RKL-based supervision into RLVR, including reward shaping and differentiable teacher regularization. The *On-Policy Distillation* blog by Thinking Machines (Lu and Lab, 2025) further compared the training cost of OPD and SFT+RL pipelines, highlighting the potential of OPD as a post-training approach. More recently, OPD has been extended to self-distillation settings such as OPSD (Zhao et al., 2026) and SDPO (Hübotter et al., 2026), which use previous trajectories with error feedback to guide exploration. Most existing OPD methods rely on RKL-based supervision and are studied in settings where the student and teacher policies remain relatively close. However, as we analyze in Section 2, the effectiveness of RKL supervision depends on the overlap between the student and teacher distributions, which limits its applicability under large policy divergence. To address this limitation, we propose TGPO, a reasoning-oriented post-training method that more effectively incorporates teacher supervision into RLVR under large teacher–student divergence.

Discussion over Mixed-Policy. In the context of LLM distillation, mixed-policy approaches (Yan et al., 2025; Zhang et al., 2025), which leverage samples from the teacher distribution, have achieved competitive results. However, our work remains strictly focused on the on-policy setting. We posit that on-policy learning, by optimizing the student's generation trajectory, offers greater robustness against distribution mismatch and ensures theoretical consistency with standard RL algorithms. By adhering to a strict on-policy setting, our insights are designed to not only advance LLM distillation but also generalize to fundamental RL research.

7 Conclusion

We present TGPO, an on-policy distillation framework designed to overcome Reverse KL limitations. Compared to the sparse and uninformative signals provided by Reverse KL based algorithms, TGPO incorporates dense and explicit teacher guidance based on the student's rollout, while maintaining the robustness of on-policy learning. Empirical results on mathematical reasoning benchmarks demonstrate that TGPO not only outperforms baselines but also exhibits adaptability to various teacher models. Moreover, we demonstrate that applying guidance via differentiable regularization, coupled with a linear decay schedule, is essential for stable convergence and continued self-improvement. We hope our findings provide a theoretically grounded and practically effective direction for future advancements in LLM alignment.

Limitation

Although TGPO demonstrates strong performance and improved training stability under large teacher–student policy divergence, the current framework is designed around the combination of token-level teacher guidance and trajectory-level verifiable rewards. As a result, TGPO is primarily suited to RLVR-style settings where reliable automatic verification signals are available. Its applicability to open-ended or subjective generation tasks remains less clear, particularly in scenarios where high-quality outcome reward models or rule-based verifiers are unavailable. In addition, like most distillation-based methods, TGPO currently assumes access to a capable teacher model that can provide informative token-level supervision during training. A promising direction for future work is to extend TGPO beyond verifiable reasoning tasks by incorporating stronger learned reward models or LLM-based judges, enabling more reliable supervision in domains with ambiguous or subjective evaluation criteria.

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*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 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)*, pages 3828–3850.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and 1 others. 2026. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

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. *arXiv preprint arXiv:2509.21880*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models, 2022. *URL https://arxiv. org/abs/2206.14858*, 1.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. *URL https://arxiv. org/abs/2503.20783*.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/on-policy-distillation.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, and 1 others. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.

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. *arXiv preprint arXiv:2506.02208*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance, 2025. *URL https://arxiv. org/abs/2504.14945*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 2025. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason. Notion Blog.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *Preprint*, arXiv:2508.11408.

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

A Theoretical Analysis of RKL Instability

In this appendix, we provide the formal derivations referenced in Section 2.2. We analyze the gradient behavior of the Reverse KL (RKL) objective and show why optimization becomes unstable when the student policy π_θ assigns probability mass to regions where the teacher policy π_T has low probability, i.e., in the **Rejection** regime defined in Section 2.2.

A.1 Gradient of RKL Objective

Recall the RKL objective in Eq. 1:

$$\begin{aligned}\mathcal{J}_{\text{RKL}}(\theta) &= \mathbb{E}_{x\sim\mathcal{D}} D_{\text{KL}}(\pi_\theta||\pi_T) \\ &= \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}\left[\log\frac{\pi_\theta(y|x)}{\pi_T(y|x)}\right].\end{aligned}$$

For a fixed prompt x, let $J(\theta)=\mathbb{E}_{y\sim\pi_\theta}[\log\rho(y)]$, where $\rho(y)=\frac{\pi_\theta(y|x)}{\pi_T(y|x)}$ denotes the density ratio. Using the log-derivative trick, the gradient with respect to θ is:

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{y\sim\pi_\theta}\left[\log\rho(y)\right] \\ &= \mathbb{E}_{y\sim\pi_\theta}\Big[\nabla_\theta\log\pi_\theta(y|x)\cdot\log\rho(y) \\ &\qquad + \nabla_\theta\log\rho(y)\Big] \\ &= \mathbb{E}_{y\sim\pi_\theta}\Big[\nabla_\theta\log\pi_\theta(y|x)\cdot\log\rho(y) \\ &\qquad + \nabla_\theta\log\pi_\theta(y|x)\Big] \\ &= \mathbb{E}_{y\sim\pi_\theta}\left[\nabla_\theta\log\pi_\theta(y|x)\cdot(\log\rho(y)+1)\right].\end{aligned}$$

Note that the term resulting from $\mathbb{E}[\nabla_\theta\log\pi_\theta]=0$ is often omitted, but strictly speaking, the gradient is weighted by the term $(\log\rho(y)+1)$.

In the context of RL with intrinsic rewards (as discussed in Section 2.2), the RKL term serves as a negative reward, $-\log\rho(y)$. Under the policy gradient framework, the resulting stochastic gradient estimator $\hat{g}(y)$ for a sampled trajectory y is proportional to:

$$\hat{g}(y)\propto\nabla_\theta\log\pi_\theta(y|x)\cdot\big(-\log\rho(y)\big).$$

A.2 Instability in the Rejection Regime

We now analyze the gradient behavior in the **Rejection** regime. Although language models generate tokens autoregressively (i.e., $\pi(y|x)=\prod_{t=1}^{L}\pi(y_t|y_{<t},x)$), our analysis operates at the complete trajectory level y. This perspective is crucial because the density ratio accumulates over the sequence length, amplifying the variance.

Unbounded Gradient Scaling. Consider a "bad" sample y_{bad} in the **Rejection** regime, where $\rho(y)\gg 1$, i.e., $\pi_\theta(y|x)\gg\pi_T(y|x)$. This corresponds to trajectories that receive non-negligible probability under the student policy but are assigned near-zero probability by the teacher. Formally, assume $\pi_\theta(y_{\text{bad}}|x)\geq\delta$ for some constant $\delta>0$, while $\pi_T(y_{\text{bad}}|x)\leq\epsilon$ with $\epsilon\to 0$.

The log-density ratio is then lower bounded by:

$$\begin{aligned}\log\rho(y_{\text{bad}}) &= \log\pi_\theta(y_{\text{bad}}|x)-\log\pi_T(y_{\text{bad}}|x) \\ &\geq \log\delta-\log\epsilon=\log\left(\frac{\delta}{\epsilon}\right).\end{aligned}$$

As $\epsilon\to 0$, the term $\log(\delta/\epsilon)$ diverges to infinity. Consequently, the gradient scaling factor $|\log\rho(y_{\text{bad}})|$ can become arbitrarily large, inducing extremely high-variance gradient estimates and unstable optimization.

This issue is particularly severe in cross-family distillation, where architectural and reasoning-style discrepancies often cause the teacher to assign near-zero probability to otherwise plausible student trajectories. This analysis directly explains the sharp gradient spikes observed in Figure 2 (Middle).

Variance Explosion. Optimization stability is closely related to the variance of the stochastic gradient estimator. A standard proxy for this variance is the second moment of the gradient norm:

$$\mathbb{E}_{y\sim\pi_\theta}\left[\|\nabla_\theta\log\pi_\theta(y|x)\|^2\cdot(\log\rho(y))^2\right].$$

In the **Rejection** regime, where $\pi_T(y|x)\to 0$, the log-density ratio $\log\rho(y)$ can become arbitrarily large. Consequently, the weighting term $(\log\rho(y))^2$ grows without bound, substantially increasing the second moment of the gradient estimator and inducing extremely high gradient variance.

Such variance can severely destabilize optimization, particularly for adaptive optimizers such as Adam, resulting in gradient spikes, noisy parameter updates, and degradation of previously learned capabilities.

A.3 Asymmetry of Reward Scaling

In Section 2.2, we argued that RKL induces an asymmetric optimization landscape. We formalize this observation by analyzing the intrinsic reward

$$r_{\text{int}}(y)=-\log\rho(y)=\log\frac{\pi_T(y|x)}{\pi_\theta(y|x)}.$$

- **Positive Rewards are Probabilistically Suppressed (Consensus Regime):** Large positive rewards arise when $\pi_T(y|x) \gg \pi_\theta(y|x)$, meaning the teacher assigns substantially higher probability to a trajectory than the student. However, trajectories are sampled from the student policy π_θ. Thus, large positive rewards are associated with trajectories that are already unlikely to be sampled. As $\pi_\theta(y|x) \to 0$, the probability of observing such trajectories vanishes, making strong positive reinforcement events exceedingly rare in practice.
- **Negative Penalties are Frequent and Unbounded (Rejection Regime):** Conversely, large negative rewards arise when $\pi_\theta(y|x) \gg \pi_T(y|x)$. Since trajectories are sampled from the student policy, such rejection trajectories are likely to be observed during optimization. At the same time, the intrinsic reward $r_{\text{int}}(y)$ becomes unbounded below as $\pi_T(y|x) \to 0$. Consequently, optimization is repeatedly dominated by large-magnitude negative updates. This asymmetry—where positive rewards are rarely observed while negative penalties are both frequent and unbounded—drives the variance explosion and instability discussed above.

B Experimental Details

In this appendix, we provide detailed experimental settings, hyperparameter configurations, and additional empirical results.

B.1 Detailed Setup

Training Dataset. All experiments use a unified dataset derived from a subset of OpenR1-Math-46k-8192. We retain the original prompts from NuminaMath 1.5 but reconstruct the reasoning traces to enable a controlled comparison across different training paradigms. Specifically, TGPO is an on-policy method that requires only prompts and generates rollouts during training. In contrast, the off-policy and mixed-policy baselines require static reasoning traces. To support these baselines under the same prompt distribution, we construct a shared set of teacher-generated trajectories using Qwen3-30B-A3B. We then validate the generated traces with Math-verify and retain only valid samples. The final curated dataset contains 34,975 prompts paired with verified teacher-generated reasoning traces for off-policy and mixed-policy training.

Training Configuration. In addition to Qwen2.5-Math-7B, we also evaluate our method on the smaller Qwen2.5-Math-1.5B model. For the main experiments, we use Qwen3-30B-A3B as the teacher model. This setting introduces a large capability gap between the teacher and the student, corresponding to the **Rejection** regime analyzed in our paper. For the in-family experiments in Section 2.3, we replace the teacher model with Qwen2.5-Math-7B while keeping all other settings unchanged. To ensure a fair comparison, all RL-based methods use a fixed sampling budget of $K = 8$ rollouts per prompt. We use a constant learning rate of 1×10^{-6} and train all RL models for 300 steps. All experiments are conducted on a cluster of 8 NVIDIA A100 GPUs. Our implementation is based on the verl framework[6] and uses vLLM[7] for efficient rollout generation.

Model Configuration. The native context window of Qwen2.5-Math-7B and Qwen2.5-Math-1.5B (4,096 tokens) is insufficient to accommodate the long reasoning traces in the off-policy data. To address this issue, we modify the model configuration by increasing the RoPE base frequency (θ) from 10,000 to 40,000 and extending the context window to 16,384 tokens. In contrast, Qwen3-30B-A3B already supports a sufficiently large context window, so we keep its RoPE configuration unchanged. In addition, we resize the vocabulary dimensions of the student and teacher models to the same size to ensure tokenizer compatibility during training.

SFT Implementation. For all SFT baselines, we use the same dataset of prompts and Qwen3-30B-A3B-generated reasoning traces described above. We follow the training protocol of OpenR1 (Hugging Face, 2025), which reproduces the performance of the distilled DeepSeek-R1 models. Specifically, we train each model for 3 epochs with a global batch size of 64 and a learning rate of 5×10^{-5}. We use a warmup ratio of 0.1 and set the maximum sequence length to 16,384 tokens.

[6] https://github.com/verl-project/verl
[7] https://github.com/vllm-project/vllm

Model	AIME 24	AIME 25	AMC	MATH-500	Minerva	Olympiad	Avg.
Qwen2.5-Math-1.5B	7.2	3.6	26.4	28.0	9.6	21.2	16.0
LUFFY	5.8	4.9	32.6	64.0	22.4	24.7	25.7
GRPO++	10.1	7.4	41.8	69.4	28.3	35.9	32.2
KDRL	0.9	0.3	5.9	11.4	5.1	4.7	4.7
OP Distill	1.7	0.4	18.8	45.2	16.2	14.7	16.2
TGPO	**11.8**	7.8	43.0	**71.4**	**30.5**	**36.6**	**33.5**

Table 3: Performance evaluation based on Qwen2.5-Math-1.5B. The teacher model employed is Qwen3-30B-A3B. All models are evaluated under a unified setting. Bold indicates the best result.

B.2 System Prompt

> Your task is to follow a systematic, thorough reasoning process before providing the final solution. This involves analyzing, summarizing, exploring, reassessing, and refining your thought process through multiple iterations. Structure your response into two sections: Thought and Solution. In the Thought section, present your reasoning using the format: "<think>\n thoughts </think>\n". Each thought should include detailed analysis, brainstorming, verification, and refinement of ideas. After "</think>\n" in the Solution section, provide the final, logical, and accurate answer, clearly derived from the exploration in the Thought section. If applicable, include the answer in \boxed{} for closed-form results like multiple choices or mathematical solutions.
>
> **User:** {QUESTION}
> **Assistant:** <think>

C Results on the 1.5B Model

C.1 Overall Performance

To further evaluate TGPO under a larger teacher–student capability gap, we conduct experiments using `Qwen2.5-Math-1.5B` as the student model and `Qwen3-30B-A3B` as the teacher model. We compare TGPO against GRPO++, RKL, KDRL, and LUFFY. Results are summarized in Table 3.

TGPO achieves the best overall performance, reaching an average score of 33.5% and outperforming both on-policy and off-policy baselines across most benchmarks. GRPO++ remains competitive with 32.2% average accuracy but still falls short of TGPO, while LUFFY performs noticeably worse than its 7B counterpart.

In contrast, RKL-based on-policy distillation methods become highly unstable in this setting. KDRL achieves only 4.7% average accuracy, and both KDRL and RKL exhibit rapid training collapse, with generation lengths frequently saturating the maximum context window. We therefore report results from the best-performing checkpoints.

C.2 Training Dynamics

To better understand the performance differences across methods, we analyze the training dynamics of the 1.5B student model. Figure 7 shows the training reward, response length, and gradient norm during the first 300 optimization steps.

TGPO achieves stable reward improvement throughout training and converges to higher reward values than RKL-based methods, while KDRL shows almost no reward improvement from the beginning of training. In terms of response length, both RKL and KDRL quickly exhibit length explosion, with generation lengths saturating the 8,192-token rollout limit. TGPO avoids this behavior and maintains response lengths comparable to GRPO++. TGPO also maintains relatively stable gradient norms throughout training, whereas RKL and KDRL exhibit substantially higher variance and LUFFY shows several large gradient spikes. Overall, these results further support the instability of RKL-based supervision under large teacher–student capability gaps and show that TGPO maintains stable optimization behavior in this setting.

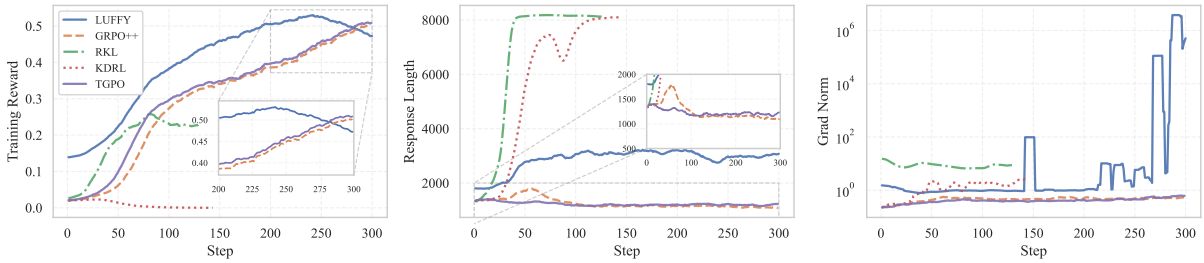

Figure 7: Training Dynamics Analysis. **(Left)** Training reward. TGPO demonstrates robust growth and convergence compared to RKL and KDRL. **(Middle)** Response length. TGPO avoids RKL's length explosion and aligns with GRPO++'s stability. **(Right)** Gradient norm. TGPO shows stable optimization compared to the high variance in RKL, KDRL and LUFFY.