TRUST REGION ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-Policy Distillation (OPD) is a fundamental technique for efficient post-training of large language models (LLMs), with broad applications in agent learning, multi-task enhancement, and model compression. However, OPD training becomes unstable when the teacher and student distributions differ substantially: teacher supervision on student-generated tokens may then yield unreliable policy gradients and even cause optimization failure. To make token-level on-policy supervision reliable, we propose Trust Region On-Policy Distillation (TrOPD), which consists of three components: **1) Trust-Region On-Policy Learning:** TrOPD performs OPD only in regions where the teacher provides reliable supervision, mitigating the optimization difficulty of the K_1 reverse-KL estimator under distribution mismatch. **2) Outlier Estimation:** For outliers outside the trust region, we compare clipping, masking, and forward-KL estimation, and adopt top-k forward KL to retain informative supervision without unreliable gradients. **3) Off-Policy Guidance:** The student imitates teacher-generated prefixes via forward KL and then continues generation on-policy, steering exploration toward the trust region. Experiments show that TrOPD consistently outperforms SoTA OPD baselines, including OPD, EOPD, and REOPOLD, across mathematical reasoning, code generation, instruction following, and STEM benchmarks. Our code is publicly available to further benchmark advanced OPD methods in community.

1 INTRODUCTION

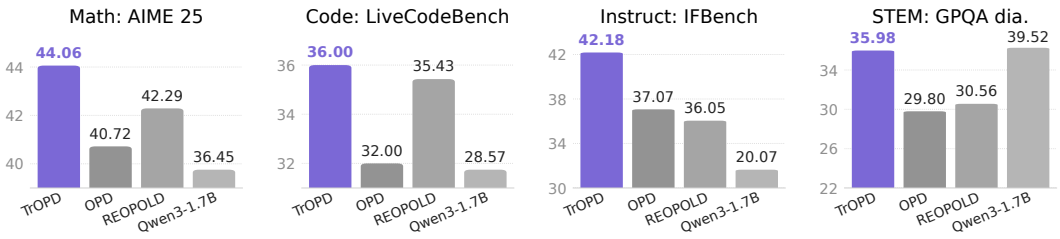


Figure 1: Performance comparison of TrOPD and baselines. All OPD methods, including TrOPD, OPD, and REOPOLD (Ko et al., 2026), are trained from Qwen3-SFT-1.7B, which is obtained by supervised finetuning of Qwen3-1.7B-Base. Qwen3-1.7B is the official Qwen model distilled with substantially more enclosed data.

Recent Large Reasoning Models (LRMs) (Zhang et al., 2025; Chen et al., 2024) scale test-time reasoning to achieve expert-level performance in mathematics (Ren et al., 2025), code generation (Anthropic, 2025), and agent tasks (Ghareeb et al., 2025). However, their substantial inference costs motivate the development of Small Reasoning Models (SRMs) (Zhao et al., 2025) for resource-efficient deployment, typically via distillation from LRMs. Conventional off-policy distillation (Kim & Rush, 2016) trains the student to imitate teacher-generated outputs. Since training relies on teacher-generated trajectories while inference follows student-generated ones, this paradigm suffers from exposure bias, especially in long chain-of-thought reasoning. On-Policy Distillation (OPD) (Lu et al., 2025; Agarwal et al., 2024) mitigates this issue by training directly on student-generated trajectories, making it an effective approach for building SRMs.

Despite this advantage, existing OPD methods often suffer from training instability caused by unreliable supervision. When the teacher and student distributions diverge substantially, student-

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

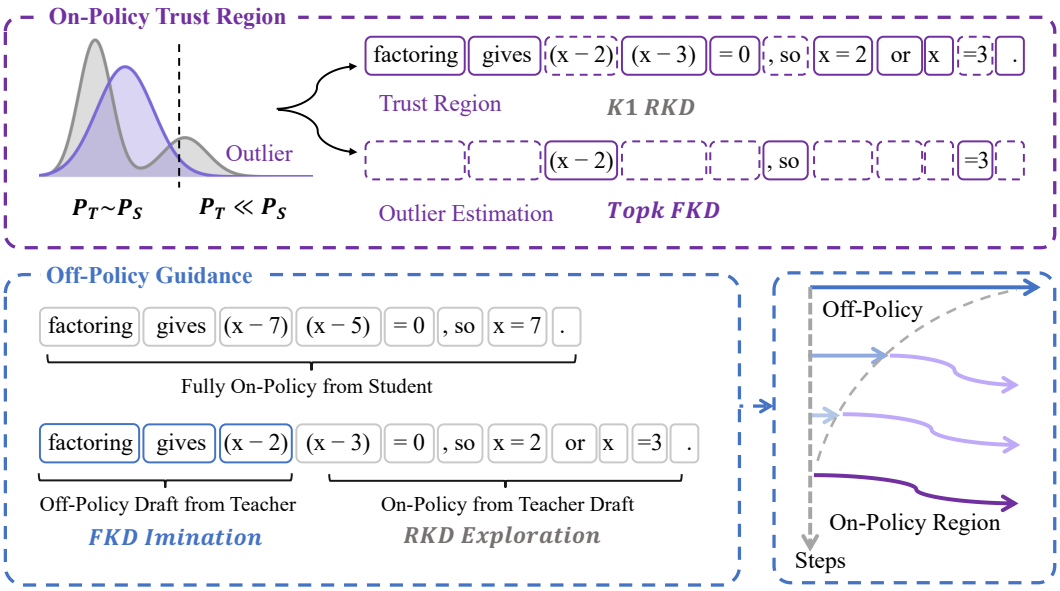


Figure 2: Overview of Trust Region On-Policy Distillation. For the on-policy component, student-generated tokens are divided into the trust region and outliers. The student model is further guided by teacher-generated responses.

generated trajectories may fall outside the teacher's reliable supervision region, yielding erroneous policy gradients and potentially causing training collapse. This problem is aggravated in reasoning-oriented OPD: to avoid the prohibitive memory overhead of full-vocabulary supervision on long responses (Agarwal et al., 2024), it relies on sampled KL estimators (Lu et al., 2025), which further reduce supervision reliability.

Yet it remains unclear how well existing methods handle this issue, as they are evaluated under inconsistent settings. To this end, we first establish a unified benchmark for reasoning-oriented OPD from three perspectives: (1) multi-domain evaluation, covering mathematics, code generation, instruction following, and STEM reasoning; (2) diverse OPD strategies, comparing representative conventional and recent methods under unified training settings; and (3) memory-efficient KL estimation, implementing the K_1 and top-k estimators for long reasoning distillation under practical memory budgets. Our evaluation reveals that existing methods only partially suppress erroneous policy gradients, yielding limited performance gains over vanilla OPD. Furthermore, the naive token masking methods or reward clipping in REOPOLD (Ko et al., 2026) discard informative supervision along with outlier gradients, which limits performance.

To address this, we propose Trust Region On-Policy Distillation (TrOPD), which partitions student-generated tokens according to their supervision reliability. As shown in Figure 2, TrOPD assigns a token to the teacher-verifiable trust region according to the teacher–student decoding agreement ratio, and applies K_1 reverse-KL supervision only within this region. For outliers, we employ a top-k forward-KL estimator to retain informative supervision without unreliable policy gradients. To further steer the student toward the trust region, we introduce off-policy guidance, in which the student imitates teacher-generated prefixes and then continues generation on-policy. As shown in Figure 1, TrOPD improves over OPD by $+3.34$, $+4.00$, $+5.11$, and $+6.18$ points on math, code, instruction following, and STEM benchmarks, respectively.

Our contributions are summarized as follows:

- We establish a unified benchmark for reasoning-oriented OPD and identify supervision reliability as a key bottleneck of existing methods.
- We propose TrOPD, which combines trust-region on-policy learning, outlier estimation, and off-policy guidance for stable and effective reasoning distillation.
- We train small reasoning models with TrOPD, which consistently outperform SoTA OPD baselines across diverse reasoning domains.

2 Related Works

Reasoning Language Models. Reasoning ability (Zhang et al., 2025; Chen et al., 2024) has become a major driver of performance improvements in large language models. It was initially elicited through reasoning prompts (Yao et al., 2022), and is now acquired through supervised finetuning (Team, 2025), reinforcement learning (Zhang et al., 2025), and on-policy distillation (Yang et al., 2025). Reasoning has since become a foundational approach for improving frontier models, such as in agentic tasks (Singh et al., 2025; Xu et al., 2026) and multimodal understanding (Comanici et al., 2025). However, how to equip SRMs (Allal et al., 2025; Zhao et al., 2025) with strong reasoning capabilities remains underexplored.

Knowledge Distillation. Knowledge distillation was originally introduced by Hinton et al. (2015) to efficiently train compact models. For generative language models, distillation has primarily relied on sequence-level knowledge distillation (Kim & Rush, 2016), an off-policy approach that performs supervised finetuning on teacher-generated responses. To mitigate the resulting exposure bias in sequence generation (Agarwal et al., 2024), full-vocabulary on-policy distillation methods (Ko et al., 2024; 2025; Xu et al., 2025) have been developed, such as MiniLLM (Gu et al., 2024) and GKD (Agarwal et al., 2024). In the post-training of frontier models, multi-teacher on-policy distillation (MOPD) (Xiao et al., 2026; Zeng et al., 2026) is widely adopted to merge the capabilities of multiple domain-specific teachers. For OPD training, KL objectives based on the K_1 estimator (Lu et al., 2025; Yang et al., 2026; Yuan et al., 2026; Fu et al., 2026) have become a standard choice for memory-efficient training. However, their supervision becomes unreliable under large teacher–student distribution mismatch, which is the focus of this work.

3 Problem Formulation

3.1 OPD for Language Models

On-policy distillation (OPD) trains the student on its own generations rather than on teacher-generated responses, thereby reducing the train–inference mismatch and mitigating exposure bias. This on-policy setting naturally aligns with the reverse KL (RKL) divergence, $D_{\mathrm{KL}}(\pi_S \| \pi_T) = \mathbb{E}_{x\sim\pi_S}\left[\log \frac{\pi_S(x)}{\pi_T(x)}\right]$, which is taken under the student distribution and can therefore be directly estimated from student-generated trajectories. At each generation step, RKL provides a token-level reward that keeps the student close to the teacher at the states it actually visits. Accordingly, OPD methods use the token-level RKL as a reward and optimize the student with policy gradient. However, conventional OPD computes RKL or Jensen–Shannon divergence (JSD) over the full vocabulary, which typically includes more than 100k tokens, incurring substantial memory overhead on long generations.

3.2 OPD for Reasoning Models

Reasoning models scale performance by extending the reasoning length, which makes this memory overhead a major bottleneck for distillation. Specifically, full-vocabulary OPD optimizes

$$\mathcal{J}^{\mathrm{OPD}} = -\mathrm{KL}(\pi_S \parallel \pi_T) = -\sum_{x\in\mathcal{V}} \pi_S \log \frac{\pi_S}{\pi_T}, \tag{1}$$

which requires $\mathcal{O}(n \cdot k)$ memory, where n is the sequence length and k is the vocabulary size. For reasoning training with an 8K sequence length (Section 5.1), computing the KL tensor over the full vocabulary requires **2.4 TB GPU memory**, which is prohibitive in most practical settings.

Simply shortening the reasoning length is not a viable remedy, as insufficient reasoning length can directly cause reasoning-model training to fail. To preserve long reasoning sequences, Thinking Machines Lab (Lu et al., 2025) adopts the K_1 estimator, an unbiased single-sample KL estimate that yields the objective:

$$\mathcal{J}^{\mathrm{OPD}} = -\mathrm{KL}(\pi_S \parallel \pi_T) = -\mathbb{E}_{x\sim\pi_S}\left[\log \frac{\pi_S}{\pi_T}\right]. \tag{2}$$

Since only the log-probabilities of the sampled tokens are required, the memory cost reduces to $\mathcal{O}(n)$, *i.e.*, only **16.8 MB** under the same setting, making long reasoning OPD practical.

Method	FKL Objective	RKL Objective	AIME 24	AIME 25	AMC 23	Avg.
DeepSeek-Qwen2.5-1.5B	–	–	28.64	24.16	71.01	41.27
OPD (RKL)	–	$\log \pi_T/\pi_S$	35.83	29.16	75.39	46.79
Distribution Mixing Strategies						
FKL	$\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	–	0.00	0.00	4.21	1.40
JSD	$\beta \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{M,v}/\pi_{T,v})$	$(1-\beta)\log(\pi_M/\pi_S)$	37.91	30.72	75.07	47.90
Entropy-Aware Token Selection Strategies						
Entropy OPD 20%	–	$\mathbb{I}[\mathrm{H} > \tau] \log \pi_T/\pi_S$	35.52	29.06	73.82	46.13
Outlier-Aware Token Selection Strategies						
Clip Outlier	–	$\max(\log \pi_T/\pi_S, \tau)$	36.97	30.83	75.78	47.86
Mask Outlier	–	$\mathbb{I}[\mathrm{R} > \tau] \log \pi_T/\pi_S$	37.08	30.62	75.46	47.72
FKL Outlier	$\overline{\mathbb{M}} \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	$\mathbb{M} \log \pi_T/\pi_S$	39.16	29.89	77.96	49.00
TrOPD	$\overline{\mathbb{M}} \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	$\mathbb{M} \log \pi_T/\pi_S$	38.54	32.50	78.51	**49.85**

Table 1: Comparison of OPD methods on math-domain reasoning benchmarks. All OPD methods are trained with Skywork-OR1-Math-7B as the teacher model.

Despite its memory efficiency, K_1-based OPD suffers from two key optimization bottlenecks:

Significant policy-gradient outliers. The K_1 estimator uses the token-level log-ratio $\log \frac{\pi_T(x)}{\pi_S(x)}$ as the reward, which is unbounded from below. When the teacher and student distributions diverge substantially, the teacher may assign extremely low probabilities to tokens sampled by the student, *i.e.*, $\pi_T(x) \approx 0$ for $x \sim \pi_S$. For such tokens, the K_1-based policy-gradient signal becomes extremely negative, *i.e.*, $\nabla\mathcal{J} = \frac{1}{\pi_S(x)} \log \frac{\pi_T(x)}{\pi_S(x)} \to -\infty$. These outliers arise precisely where the teacher cannot provide reliable supervision, yet they dominate the gradients of the remaining tokens. As a result, they destabilize OPD optimization and ultimately limit its attainable performance.

Low-quality student generations. Beyond individual tokens, trajectory quality also limits OPD. Since OPD learns only from student-sampled trajectories, the student may generate low-quality responses on challenging problems that deviate substantially from the teacher distribution. The teacher then assigns low likelihood to most tokens in these trajectories, leaving little informative supervision for the student to discover correct reasoning paths.

4 Trust Region Distillation

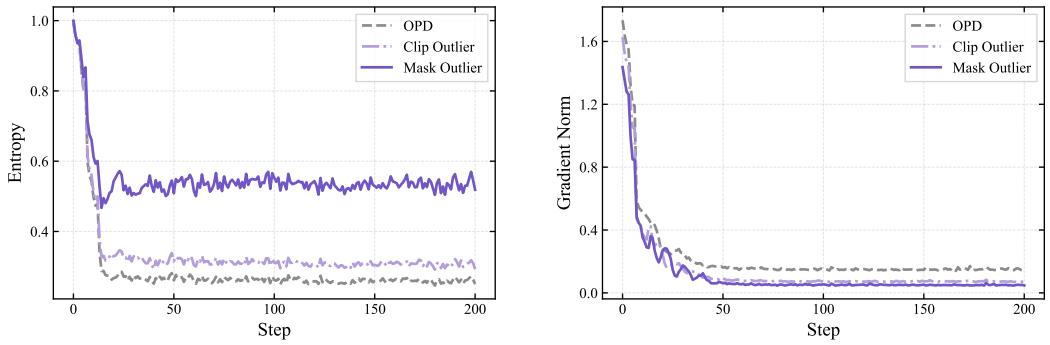
4.1 Benchmarking OPD Baselines

Compared with conventional full-vocabulary OPD, OPD for long reasoning models remains an emerging research direction. Existing studies adopt different experimental configurations, making recent SoTA methods difficult to compare directly. We therefore benchmark representative OPD methods under a unified setting (Table 1), focusing on two fundamental questions: (1) Do divergence objectives developed for full-vocabulary OPD remain effective for token-level OPD based on the K_1 estimator? (2) Can existing strategies for suppressing erroneous policy gradients effectively improve K_1-based OPD?

Divergence Evaluation. To answer question (1), we evaluate the *Distribution Mixing Strategies* in Table 1. Due to the asymmetry of KL divergence, full-vocabulary distillation methods typically adopt forward KL (FKL) for mode covering and reverse KL (RKL) for mode seeking, while GKD (Agarwal et al., 2024) employs JSD to balance the two. Under the memory constraints of long-thinking distillation, we implement FKL over the top-k tokens of the teacher distribution, *i.e.*, $\mathcal{J}_{\mathrm{FKL}}^{\text{top-}k} = -\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{S,v})$, and implement RKL with the token-level K_1 estimator. Combining the two yields the generalized JSD objective with a balancing coefficient β:

$$\mathcal{J}_{\mathrm{JSD}}^{\beta} = \beta \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{M,v}}{\pi_{T,v}} + (1-\beta) \log \frac{\pi_{M,x}}{\pi_{S,x}}, \tag{3}$$

where $x \sim \pi_S$, $\pi_{M,v} = \beta\pi_{T,v} + (1-\beta)\pi_{S,v}$, and $\beta = 0.5$ by default.



(a) Entropy comparison. (b) Gradient norm comparison.

Figure 3: Comparison of OPD methods in terms of entropy and gradient norm.

As shown in Table 1, stand-alone top-k FKL fails to train effectively, reaching an average score of only 1.40. This is mainly because top-k FKL is a biased approximation of the full-vocabulary FKL objective, and applying it to all sampled tokens introduces increasingly distorted policy gradients. In contrast, JSD, which combines top-k FKL with K_1-based RKL, improves over OPD from 46.79 to 47.90. Therefore, top-k FKL is unsuitable as a standalone objective, but has great potential to enhance RKL-based OPD objectives.

Token Filtering and Reward Clipping. For question (2), existing methods mainly suppress erroneous policy gradients through two strategies, corresponding to *Entropy OPD 20%* and *Clip Outlier* in Table 1. For entropy-based token filtering, Wang et al. (2026b) show that training only on the top 20% high-entropy tokens suppresses the interference of less informative tokens and accelerates RL convergence. For reward clipping, REOPOLD (Ko et al., 2026) clips token-level rewards at a predefined threshold τ, *i.e.*, $\max(\log \pi_T/\pi_S, \tau)$, to reduce the influence of erroneous supervision.

As shown in Tables 1 and 4, entropy-based token filtering does not consistently benefit OPD and even reduces the average score from 46.79 to 46.13 in Table 1. This suggests that, unlike in RL, the teacher also provides informative supervision on low-entropy tokens, which should not be discarded. Reward clipping improves OPD from 46.79 to 47.86 in Table 1, but its gains become marginal under other settings in Table 4. Moreover, the clipping threshold introduces an additional hyperparameter that may substantially affect learning dynamics and convergence speed.

4.2 Trust-Region On-Policy Learning.

Based on the benchmark results, directly applying conventional FKL alone fails to achieve effective training, while FKL should be properly combined with RKL to enhance the effectiveness of RKL under large distributional mismatch. Meanwhile, simple reward clipping and entropy-based token selection provide only limited correction to the policy gradients. We therefore focus on enabling more effective learning while explicitly suppressing unreliable policy-gradient.

Inspired by trust-region policy optimization (TRPO) in RL, we propose Trust Region On-Policy Distillation (TrOPD) that only optimizes where the policy-gradient is reliable. Given the trust-region $\mathbb{M}_x$ and the outlier $\overline{\mathbb{M}}_x$, the token-level objective in x can be governed by:

$$\begin{aligned} \mathcal{J}_x^{On} &= -\mathbb{M}_x \mathrm{KL}(\pi_S \parallel \pi_T) - \overline{\mathbb{M}}_x \mathrm{KL}(\pi_T \parallel \pi_S) \\ &= -\mathbb{M}_x \log \frac{\pi_S}{\pi_T} - \overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}}. \end{aligned} \tag{4}$$

Since $x \sim \pi_S$, we estimate RKL within the trust region using the K_1 estimator, while approximating the FKL term for outliers using a Top-k estimator: $\sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}}$. In the following, we reason the TrOPD objective step by step.

Outlier Masking. We focus on the first term, $-\mathbb{M}_x \log \frac{\pi_S}{\pi_T}$, and investigate whether effective training can still be achieved after removing outliers. For a fair comparison, we temporarily adopt a static threshold as *Clip Outlier* in REOPOLD. Different from *Clip Outlier*, which clips rewards

Region	Policy	Objective	Estimator	Memory
On-Policy Trust Region	$x \sim \pi_S$	$-\mathrm{KL}(\pi_S \| \pi_T)$	$\log \frac{\pi_T(x)}{\pi_S(x)}$	$\mathcal{O}(n)$
On-Policy Outlier	$x \sim \pi_S$	$-\mathrm{KL}(\pi_T \| \pi_S)$	$\sum_{v \in \mathcal{V}_T^{(k)}} \pi_T(v) \log \frac{\pi_S(v)}{\pi_T(v)}$	$\mathcal{O}(nk)$
Off-Policy Guidance	$x \sim \pi_T$	$-\beta \mathrm{KL}(\pi_T \| \pi_S)$	$\beta \log \frac{\pi_S(x)}{\pi_T(x)}$	$\mathcal{O}(n)$

Table 2: Region-specific learning objectives and estimators in TrOPD. Here, $\mathcal{V}_T^{(k)}$ denotes the top-k vocabulary under the teacher distribution π_T.

below the threshold τ, $\mathrm{R}(x) = \max(\log \frac{\pi_S}{\pi_T}, \tau)$, *Mask Outlier* directly masks the token-level advantage once its reward magnitude exceeds the threshold, $\mathrm{R}(x) = \mathbb{I}[\mathrm{R} > \tau] \log \frac{\pi_S}{\pi_T}$.

As shown in Table 1, both *Mask Outlier* and *Clip Outlier* outperform the vanilla OPD baseline, demonstrating that suppressing outlier policy gradients improves optimization stability and downstream performance. As shown in Figures 3a and 3b, *Mask Outlier* directly eliminates the influence of unreliable gradients. Compared with OPD and *Clip Outlier*, it maintains higher policy entropy, thereby better preserving the exploration capability during OPD training. Moreover, by removing the gradients induced by outlier tokens during backpropagation, TrOPD achieves a lower gradient norm than OPD and *Clip Outlier*, leading to more stable optimization.

Adaptive Trust Region. Different from previous predefined threshold τ, we define the trust region according to student policy $\pi_S(x)$ and teacher check $\pi_T(x)$. For each token sampled from the student generation, $x \sim \pi_S$, the probability of being classified into the trust region, $\mathbb{M}_x \sim \mathrm{Bernoulli}(P_{\mathrm{trust}}(x))$, is defined as:

$$P_{\mathrm{trust}}(x) = \min\left(\frac{\pi_T(x)}{\pi_S(x)}, 1\right). \tag{5}$$

This design is motivated by speculative decoding, where the probability that the teacher model agrees with a token decoded by the student model satisfies $P_{\mathrm{accept}}(x) \propto \min\left(\frac{\pi_T(x)}{\pi_S(x)}, 1\right)$. By selecting only the decoding regions accepted by the teacher, it can be ensured that the student model remains within effectively supervised regions under the K_1 estimator.

Outlier Estimation. Outlier regions exhibit substantial distributional mismatch between the teacher and student, but may still contain informative supervisory signals. Simply masking these regions may therefore discard useful knowledge. To partially recover such supervision, we introduce an auxiliary forward KL (FKL) objective in outlier regions. Specifically, because reverse KL estimated from student-sampled tokens may fail to provide reliable supervision under severe distributional mismatch, we instead compute the distillation signal from the teacher perspective, $\mathcal{J}_x^{FKL} = -\overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{S,v})$, where $\mathcal{V}_T^k = \mathrm{TopK}(\pi_T)$ denotes the teacher's top-k vocabulary. When $\exists\, v \in \mathcal{V}_T^k \quad \text{s.t.} \quad \pi_S(v) > 0$, the FKL objective enables imitation learning from informative teacher supported tokens in the outlier region. In contrast, when $\sum_{v \in \mathcal{V}_T^k} \pi_T(v) \to 0$, we have $\mathrm{KL}(\pi_T \parallel \pi_S) \to 0$, such that the auxiliary outlier objective is suppressed and does not interfere with gradient in the trust region. Therefore, this design alleviates the potential loss of supervisory information caused by region masking while preserving stable trust-region optimization.

4.3 Off-Policy Trust-Region Guidance

To guide the student model to follow the teacher's trajectory, we propose an off-policy trust region to provide offline constraints, as illustrated in Fig. 2. The distillation trajectory consists of two parts: an off-policy prefix $x[:l]$ generated by the teacher, followed by an on-policy continuation $x[l:]$ generated by the student. For complex reasoning tasks, this design avoids low-quality outputs caused by the limited capability of the student model. We apply forward KL, $\mathrm{KL}_{x[:l] \sim \pi_T}(\pi_T \parallel \pi_S)$,

for imitation learning from the off-policy guidance:

$$\begin{aligned}\mathcal{J}_x &= -\beta \mathrm{KL}_{x[:l]\sim\pi_T}(\pi_T \parallel \pi_S) + \mathcal{J}^{\mathrm{On}}_{x[l:]} \\ &= -\beta \mathbb{I}[x \sim \pi_T] \log \frac{\pi_T}{\pi_S} + \mathbb{I}[x \sim \pi_S] \mathcal{J}^{\mathrm{On}}_{x[l:]}\end{aligned} \tag{6}$$

For the off-policy region, since samples are generated from the teacher, $x \sim \pi_T$, we adopt the K_1 estimator for the forward KL, $\mathrm{KL}(\pi_T \parallel \pi_S) = \log \frac{\pi_S}{\pi_T}$, which achieves $\mathcal{O}(n)$ memory complexity.

Unified Optimization. We summarize the overall objective of TrOPD as:

$$\begin{aligned}\mathcal{J}_x^{\mathrm{TrOPD}} = &-\mathbb{I}[x \sim \pi_S] \overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}} \\ &- \mathbb{I}[x \sim \pi_S] \mathbb{M}_x \log \frac{\pi_S}{\pi_T} - \beta\, \mathbb{I}[x \sim \pi_T] \log \frac{\pi_T}{\pi_S}.\end{aligned} \tag{7}$$

where the details of each component are in Table 2. Initially, the maximum off-policy trajectory length is set to the maximum training sequence length. During training, it is gradually annealed to zero using a cosine schedule, such that generation becomes fully on-policy by the end of training.

5 Experimental Results

5.1 Implementation Details

Without loss of generality, we benchmark OPD for reasoning models in two settings, i.e., single-domain and multi-domain reasoning distillation.

Model. (1) For single-task distillation with DeepSeek-Distilled-Qwen-1.5B (Zhang et al., 2025), we use the representative mathematical reasoning task, where Skywork-OR1-Math-7B (He et al., 2025b) serves as the teacher model and DeepSeek-Distilled-Qwen-1.5B serves as the student model. Since the student has already undergone extensive off-policy distillation, we directly conduct OPD without additional SFT. (2) For multi-task distillation with DeepSeek-Distilled-Qwen-1.5B, we use Skywork-OR1-7B (He et al., 2025b) as the teacher model and DeepSeek-Distilled-Qwen-1.5B as the student model to evaluate OPD in multi-domain reasoning tasks. (3) For multi-task distillation with Qwen3-SFT-1.7B (Wang et al., 2026a), the teacher model, Qwen3-Nemotron-4B (Wang et al., 2026a), is trained from Qwen3-4B-Base (Yang et al., 2025) using Nemotron SFT & RL data and recipe (Appendix A for details). The student model is trained from Qwen3-1.7B-Base using the same SFT recipe.

Dataset. (1) For single-domain distillation, we use prompts from the OpenThoughts3 dataset (Guha et al., 2025) and retain only examples from the mathematics domain. (2) For multi-domain distillation, we use prompts from OpenThoughts3, covering the mathematics, code, and science domains. Only the prompts are retained for training.

Benchmark Training. To fairly compare existing OPD methods, we train all benchmarks using the same training settings. Specifically, we perform OPD training for 200 steps using a fixed learning rate of 5×10^{-6}. For FKL-based methods implemented with a top-k support set, we uniformly set $k = 64$. For off-policy guidance, we set $\beta = 0.001$ for imitation learning. We use a prompt batch size of 128 with 4 rollouts per prompt and a maximum generation length of 8192 tokens.

Benchmark Evaluation. We evaluate the distilled models across mathematics, STEM, and instruction following domains. For mathematical reasoning, we report results on AIME 2024, AIME 2025 (Shi et al., 2025), and AMC 2023, while each result is the average accuracy of 32 times evaluation. For STEM reasoning and instruction following, we use GPQA Diamond, MMLU-Redux v2 (Gema et al., 2025), and IFBench (Pyatkin et al., 2025). For code generation, we evaluate on LiveCodeBench v6 (Jain et al., 2025).

Method	AIME 24	AIME 25	AMC 23	LiveCodeBench v6	GPQA diamond	Avg.
DeepSeek-Qwen2.5-1.5B	28.64	24.16	71.01	15.43	34.22	34.69
Single-Domain Distillation						
Teacher	66.14	51.87	92.34	34.86	47.22	58.48
OPD	35.83	29.16	75.39	17.14	28.03	37.11
EOPD	36.97	29.79	75.23	15.43	32.58	38.00
Entropy OPD 20%	35.52	29.06	73.82	14.29	31.82	36.90
REOPOLD 2Stage	34.47	29.89	73.35	16.57	30.18	36.89
REOPOLD	36.97	30.83	75.78	18.29	32.07	38.79
TrOPD	**38.54**	**32.50**	**77.03**	**18.86**	**36.24**	**40.63**
Multi-Domain Distillation						
Teacher	65.62	52.81	91.79	36.57	47.22	58.80
OPD	30.10	21.66	61.56	20.57	31.06	32.99
REOPOLD	34.27	25.83	63.90	19.43	**34.47**	35.58
TrOPD	**36.04**	**27.60**	**70.93**	**22.29**	31.19	**37.61**

Table 3: Performance comparison using DeepSeek-R1-Distill-Qwen-1.5B as the student model. Skywork-OR1-Math-7B and Skywork-OR1-7B are teacher models for the single-domain and multi-domain distillation respectively.

Method	**Math**			**STEM**		**Instruct**	**Code**	**Avg.**
	AIME 24	AIME 25	AMC 23	GPQA dia.	MMLU red.	IFBench	LCB.v6	
Qwen3-SFT-1.7B	35.41	26.45	68.90	25.25	66.60	26.19	30.29	39.87
Multi-Domain Distillation								
Teacher	81.66	75.72	98.98	58.86	77.03	62.93	58.86	73.43
OPD	48.02	40.72	81.79	29.80	68.60	37.07	32.00	48.29
EOPD	47.08	40.83	81.32	33.84	68.26	36.39	34.29	48.86
Entropy OPD	43.54	42.70	79.53	29.92	68.51	38.78	33.71	48.10
REOPOLD	45.62	42.29	81.64	30.56	68.30	36.05	35.43	48.56
TrOPD	**52.08**	**44.06**	**83.04**	**35.98**	**68.74**	**42.18**	**36.00**	**51.73**

Table 4: Performance comparison using Qwen3-SFT-1.7B as the student model. Qwen3-Nemotron-4B is the teacher model for the multi-domain distillation.

5.2 Main Results

Single-Domain Distillation. As shown in Table 3, we primarily evaluate mathematical reasoning performance on AIME 2024, AIME 2025, and AMC 2023. We further report out-of-domain (OOD) results to assess robustness to domain shifts. Compared with OPD, TrOPD improves the average performance by +3.06 points on mathematical reasoning tasks and by +3.52 points on average of 5 tasks. REOPOLD corrects unreliable policy gradients by reward clipping; TrOPD still outperforms it by 1.99 and 1.84 points in the mathematical and general domains, showing the necessity of trust-region learning and outlier estimation. Furthermore, compared with entropy-based token selection methods, including EOPD (Jin et al., 2026), Entropy OPD (Ko et al., 2026; Wang et al., 2026b), and REOPOLD 2Stage (Ko et al., 2026), TrOPD achieves improvements of 2.63, 3.73, and 3.74 points, respectively. Outlier-aware token selection is thus a more effective criterion than entropy-based selection.

Multi-Domain Distillation. As shown in Table 3 and Table 4, we evaluate multi-domain distillation performance with Skywork-OR1-Math-7B and Qwen3-Nemotron-4B as the teacher models, respectively. Since Skywork-OR1-Math-7B is primarily trained for math and code, we evaluate its distilled domains on AIME 2024, AIME 2025, AMC 2023, and LiveCodeBench. Compared with OPD, TrOPD consistently improves the performance of both DeepSeek-Qwen2.5-1.5B and Qwen3-SFT-1.7B, achieving substantial average gains of +4.62 and +3.44 points, respectively. These gains hold across different teacher–student configurations and reasoning tasks.

5.3 Ablation Studies and Discussion

Outlier Estimation and Off-Policy Guidance. As shown in Table 5, applying FKL only to outlier regions achieves better performance than masking or clipping outliers, demonstrating its effectiveness for outlier estimation. Incorporating off-policy guidance further improves the average scores of

Method	Outlier Objective	AIME 24	AIME 25	AMC 23	Avg.
Single-Domain Distillation					
DS-1.5B	–	28.64	24.16	71.01	41.27
OPD	$\log \pi_T/\pi_S$	35.83	29.16	75.39	46.79
+ Outlier Estimation					
Mask Outlier	0	37.08	30.62	75.46	47.72
Clip Outlier	τ	36.97	30.83	75.78	47.86
Full FKL	$\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	0.00	0.00	4.21	1.40
FKL Outlier	$\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	39.16	29.89	77.96	49.00
+ Off-Policy Guidance					
TrOPD Mask	0	40.10	30.41	75.85	48.79
TrOPD Clip	τ	37.39	31.77	77.03	48.73
TrOPD FKL	$\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{S,v}/\pi_{T,v})$	38.54	32.50	78.51	49.85

Table 5: Ablation Studies of TrOPD in math-domain distillation.

Method	Verifier	Gold Label	AIME 24	AIME 25	AMC 23	Avg.
GRPO	✓	✓	33.22	24.79	75.78	44.60
GRPO + OPD	✓	✓	37.08	30.93	74.37	47.46
OPD	✗	✗	37.39	29.79	77.73	48.30
TrOPD	✗	✗	**41.45**	**31.14**	**80.00**	**50.86**

Table 6: Stand-alone OPD versus reinforcement learning, retrained on DAPO-math-17k. *Verifier* and *Gold Label* indicate whether a method requires a domain-specific verifier and ground-truth answers during training.

the Clip, Mask, and FKL Outlier variants, confirming its complementary benefit. Consequently, the three TrOPD variants, i.e., TrOPD Mask, TrOPD Clip, and TrOPD FKL, outperform OPD by 2.00, 1.94, and 3.06 points on average, respectively.

Design Choices and Extensibility. Appendix B further shows that the adaptive acceptance criterion outperforms a fixed threshold on the probability ratio, and that TrOPD is robust to the off-policy weight, the annealing schedule, and the number of training steps. We also note the concurrent work AOPD (Jia et al., 2026): TrOPD outperforms it, while their combination further raises the average score from 40.63 to 41.67 (Appendix C). This suggests that AOPD is orthogonal to TrOPD, and that combining complementary OPD strategies is a promising direction for future work.

Comparison with Reinforcement Learning. OPD is used here as a stand-alone objective rather than an auxiliary RL loss, raising the question of whether an RL term is needed at all. Because GRPO (Shao et al., 2024) requires ground-truth labels and a domain-specific verifier, which TrOPD does not, we retrain all four configurations on DAPO-math-17k (Yu et al., 2025) with a math verifier for this comparison; the training setup is detailed in Appendix D. As shown in Table 6, both stand-alone OPD methods outperform GRPO and the hybrid GRPO + OPD, and the advantage of the hybrid over GRPO comes almost entirely from its OPD term. TrOPD attains the best result, exceeding GRPO by 6.26 points and the hybrid GRPO + OPD by 3.40 points while requiring neither ground-truth labels nor a verifier.

6 Conclusion

This work proposes Trust Region On-Policy Distillation, a reliable and stable framework for reasoning-oriented OPD. By trust region optimization and outlier estimation, TrOPD suppresses unreliable policy gradients while preserving informative supervision, and off-policy guidance further encourages exploration toward teacher-supported trajectories. Extensive multi-domain results highlight the importance of supervision reliability and the potential of trust-region learning for training high-quality small reasoning models.

AI USE STATEMENT

In preparing this paper, we used GPT-5.6 Sol solely for grammar checking and minor language polishing. The LLM was not used for research ideation, method design, mathematical derivation, code implementation, experiments, data analysis, or the interpretation of results. All technical content was produced by the authors, and all suggested edits were reviewed and verified by the authors, who take full responsibility for the content of this paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Anthropic. Claude 3.7 sonnet and claude code, 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*, 2025.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024. URL https://arxiv.org/abs/2412.18925.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Jon M. Laurent, Muhammed T. Razzak, Andrew D. White, Michaela M. Hinks, and Samuel G. Rodriques. Robin: A multi-agent system for automating scientific discovery, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Yang Liu, and Yahui Zhou. Skywork open reasoner series, 2025a. Notion Blog.

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

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

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

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

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

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=lsHZNNoC7r.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rc65N9xIrY.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. *arXiv preprint arXiv:2203.07814*, 2022.

Kevin Lu et al. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation/, 2025. Thinking Machines Blog, accessed on 2025-10-27.

NVIDIA Corporation. Opensciencereasoning-2 dataset. Hugging Face Dataset, 2025. Available at: https://huggingface.co/datasets/nvidia/OpenScienceReasoning-2.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. Codeforces cots. https://huggingface.co/datasets/open-r1/codeforces-cots, 2025.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025.

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025.

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

Yexuan Shi, Mingyu Wang, Yunxiang Cao, Hongjie Lai, Junjian Lan, Xin Han, Yu Wang, Jie Geng, Zhenan Li, Zihao Xia, et al. Aime: Towards fully-autonomous multi-agent framework. *arXiv preprint arXiv:2507.11988*, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Kimi Team. Kimi k2: Open agentic intelligence, 2025. URL `https://arxiv.org/abs/2507.20534`.

Haoqing Wang, Xiang Long, Ziheng Li, Yilong Xu, Tingguang Li, and Yehui Tang. To mix or to merge: Toward multi-domain reinforcement learning for large language models. *arXiv preprint arXiv:2602.12566*, 2026a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026b.

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

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arxiv preprint arXiv:2505.09388*, 2025.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao, Zhanfeng Mo, Qi Zhang, et al. 100 days after deepseek-r1: A survey on replication studies and more directions for reasoning language models. *arXiv preprint arXiv:2505.00551*, 2025.

Changsheng Zhao, Ernie Chang, Zechun Liu, Chia-Jung Chang, Wei Wen, Chen Lai, Sheng Cao, Yuandong Tian, Raghuraman Krishnamoorthi, Yangyang Shi, et al. Mobilellm-r1: Exploring the limits of sub-billion language model reasoners with open training recipes. *arXiv preprint arXiv:2509.24945*, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024.

APPENDIX

A TRAINING DETAILS OF TEACHER MODEL QWEN3-NEMOTRON-4B

The training pipeline is initialized from Qwen3-4B-Base and consists of both supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR). For SFT, publicly available datasets released with Nemotron 3 Nano[1] (Blakeman et al., 2025) are adopted. Entries without the `messages` field are removed, and datasets from multiple domains are combined. To match the domain distribution ratios reported in the corresponding technical report, smaller datasets are up-sampled while larger datasets are randomly downsampled, resulting in approximately 14M training samples.

For RLVR, the publicly available training blend released with Nemotron 3 Nano[2] is adopted. The resulting training mixture covers four domains: (1) *Math*: 22,056 samples from DAPO (Yu et al., 2025) and Skywork (He et al., 2025b;a); (2) *Coding*: 19,169 samples from CodeContests (Li et al., 2022) and Open-R1 (Penedo et al., 2025); (3) *Science*: 19,670 samples from OpenScienceReasoning-2 (NVIDIA Corporation, 2025); and (4) *Instruction Following*: 16,575 samples from WildChat-1M (Zhao et al., 2024), with instructions sourced from Open-Instruct (Lambert et al., 2024).

During SFT, the Adam optimizer (Kingma, 2014) is used with a learning rate of 5×10^{-5} and a weight decay of 0.1. The warmup phase accounts for 10% of the total training steps. The batch size is set to 512, with an average response length of approximately 7K tokens.

During RLVR, GRPO is employed with a group size of 16, together with masked importance sampling to improve consistency between training and inference. The batch size is set to 128, and model parameters are updated every 2048 rollouts. The maximum generation length is capped at 32K tokens, and the sampling temperature is set to 1.0 to encourage exploration. More details on the training recipe of the multi-task reinforcement learning can be found in Wang et al. Wang et al. (2026a).

B ABLATION STUDIES ON TROPD DESIGN CHOICES

This appendix ablates the main design choices of TrOPD: the form of the trust-region criterion (Appendix B.1), the off-policy weight (Appendix B.2), the off-policy annealing schedule (Appendix B.3), and the number of training steps (Appendix B.4). Unless stated otherwise, every run follows the single-domain distillation setting of Section 5.1 with DeepSeek-Qwen2.5-1.5B as the student, and varies one factor at a time while keeping all remaining components of TrOPD unchanged.

B.1 TRUST-REGION CRITERION: ADAPTIVE ACCEPTANCE VERSUS A FIXED THRESHOLD

TrOPD admits a student token into the trust region with probability $P_{\mathrm{trust}}(x) = \min\left(\pi_T(x)/\pi_S(x),\, 1\right)$, a per-token rule that depends on the teacher–student probability ratio rather than on a single global cutoff. A simpler alternative is to declare a token reliable whenever its reward magnitude stays below a fixed threshold. To isolate this choice, we replace P_{trust} with the clipping threshold $\ln(\lambda)/(\lambda - 1)$ at $\lambda = 0.1$ adopted by REOPOLD (Ko et al., 2026), taken as a representative fixed trust-region boundary, and leave every other component of TrOPD untouched.

As shown in Table 7, the adaptive acceptance criterion outperforms the fixed threshold by $+1.46$ and $+1.78$ points on AIME 24 and AIME 25, the two benchmarks that demand the longest chains of thought, and by $+0.49$ points on average. A fixed threshold applies the same cutoff to every position and therefore cannot adapt to token-specific uncertainty, whereas the acceptance rule retains exactly those tokens on which the teacher already agrees with the student, which is the regime in which the K_1 estimator provides reliable supervision.

[1] `https://huggingface.co/collections/nvidia/nemotron-post-training-v3`

[2] `https://huggingface.co/datasets/nvidia/Nemotron-3-Nano-RL-Training-Blend`

Trust-Region Criterion	AIME 24	AIME 25	AMC 23	Avg.
Fixed Threshold $\ln(\lambda)/(\lambda-1)$	37.08	30.72	78.82	48.87
Adaptive Acceptance $\min(\pi_T/\pi_S, 1)$	38.54	32.50	77.03	**49.36**

Table 7: Ablation on the form of the trust-region criterion. The adaptive acceptance probability derived from speculative decoding is compared against a fixed threshold on the probability ratio, with all other components of TrOPD unchanged.

B.2 Off-Policy Weight

Following the classic PPO practice for KL penalties, the off-policy weight β of the forward-KL guidance term is set to 0.001 as a safe default throughout this work. Table 8 varies it by an order of magnitude. Average performance stays within 0.33 points across the two settings, so the method is not sensitive to this choice; per-benchmark differences trade off against one another rather than favoring a single value. Further tuning may yield better results but is beyond the scope of this paper, and for the sake of algorithmic generality we retain the same KL penalty weight as in classic PPO.

Off-Policy Weight β	AIME 24	AIME 25	AMC 23	Avg.
0.01	39.89	30.20	78.98	**49.69**
0.001	38.54	32.50	77.03	49.36

Table 8: Sensitivity to the off-policy weight β of the forward-KL guidance term. The highlighted row is the default adopted throughout this work.

B.3 Off-Policy Annealing Schedule

The off-policy prefix length l governs the ratio of off-policy to on-policy learning: it is initialized to the maximum training sequence length and annealed to zero over training, so that generation becomes fully on-policy by the end. We ablate how the prefix length is annealed by comparing three schedules at a fixed budget: a Step schedule that shortens the prefix every 40 of the 200 training steps, a Linear schedule, and the Cosine schedule adopted in the main text.

Table 9 shows that the three schedules land within 0.03 points of one another on average, so TrOPD is insensitive to the particular shape of the schedule; per-benchmark differences trade off against one another, and we adopt the Cosine schedule throughout the main text. The choice that matters is therefore annealing itself rather than its shape: holding l fixed throughout training leaves a constant fraction of every trajectory off-policy, so exposure bias is never removed and the objective at convergence is not the on-policy one. This is consistent with recent findings that the beginning of a response is more critical than its end in OPD training, which makes a schedule that front-loads teacher guidance and then withdraws it the natural choice.

Annealing Schedule	AIME 24	AIME 25	AMC 23	Avg.
Step	38.33	31.14	78.51	49.33
Linear	38.15	30.72	79.21	49.36
Cosine	38.54	32.50	77.03	49.36

Table 9: Ablation on the annealing schedule of the off-policy prefix length l. The Step schedule shortens the prefix every 40 of the 200 training steps.

B.4 Training Steps

Modern OPD methods converge within a short number of training steps, and state-of-the-art approaches in this field consistently adopt budgets of 100–200 steps: on-policy distillation (Lu et al., 2025) uses 100 steps, REOPOLD (Ko et al., 2026) uses 40 or 200, and AOPD (Jia et al., 2026) uses 90 or 130. In the comparative experiments of the main text we therefore extend all baselines uniformly to 200 steps, which is at or above the budget each method was originally tuned for.

To confirm that the ranking is not an artifact of this budget, Table 10 extends both OPD and TrOPD to 300 steps. The gap does not close with additional training: TrOPD leads OPD by 2.55 points on average at 300 steps, so the improvement reflects the optimization behavior of the objective rather than a difference in convergence speed.

Method (300 steps)	AIME 24	AIME 25	AMC 23	Avg.
OPD	35.93	30.20	76.40	47.51
TrOPD	40.10	32.81	77.26	**50.06**

Table 10: Comparison under an extended training budget of 300 steps, against the 200 steps used throughout the main experiments.

C Comparison with Concurrent AOPD

We also notice the concurrent work AOPD (Jia et al., 2026), which likewise targets token-level supervision in OPD but balances exploitation and imitation asymmetrically at the token level rather than restricting optimization to a trust region. As shown in Table 11, TrOPD outperforms AOPD, while their combination further improves the average score from 40.63 to 41.67. This suggests that AOPD is orthogonal to TrOPD, and that combining complementary OPD optimization strategies is a promising direction for future work.

Method	AIME 24	AIME 25	AMC 23	LiveCodeBench v6	GPQA diamond	Avg.
DeepSeek-Qwen2.5-1.5B	28.64	24.16	71.01	15.43	34.22	34.69
OPD	35.83	29.16	75.39	17.14	28.03	37.11
AOPD	39.89	30.00	77.18	20.57	31.31	39.79
TrOPD	38.54	32.50	77.03	18.86	36.24	40.63
TrOPD + AOPD	42.08	31.87	78.20	21.71	34.47	**41.67**

Table 11: Performance comparison between TrOPD and concurrent AOPD. TrOPD + AOPD indicates TrOPD adding the AOPD objective for the AOPD positive samples.

D Training Setup for the Comparison with Reinforcement Learning

The comparison in Table 6 requires a setting in which reinforcement learning is applicable, and therefore departs from the OPD-only setup of the main experiments. GRPO and GRPO + OPD both need ground-truth annotated data together with a domain-specific verifier, which are difficult to obtain outside narrow domains; TrOPD requires neither. For this experiment we therefore retrain all four configurations on the DAPO-math-17k mathematical dataset (Yu et al., 2025), using a math verifier to supply the reinforcement-learning reward, with DeepSeek-distilled-Qwen-1.5B as the student and Skywork-OR1-Math-7B as the teacher. All remaining hyperparameters follow the benchmark training settings described in Section 5.1.