Trust Region On-Policy Distillation

Anonymous ACL submission

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, as teacher supervision on student-generated tokens may yield unreliable policy gradients and even cause optimization failure. This work addresses reliable on-policy token-level supervision through credit assignment strategies, and proposes Trust Region On-Policy Distillation, TrOPD. It features the following characteristics: **1) Trust-Region On-Policy Learning:** TrOPD performs OPD only in regions where the teacher provides reliable supervision, mitigating the optimization difficulty of the K1 reverse-KL estimator under distribution mismatch. **2) Outlier Estimation:** For outlier regions, we explore gradient clipping, masking, and forward-KL estimation to reduce the adverse effects of unreliable supervision. **3) Off-Policy Guidance:** The student continues generation from teacher prefixes and uses forward KL to imitate off-policy guidance, encouraging on-policy exploration toward reliable regions. Experiments show that TrOPD consistently outperforms SoTA OPD baselines, including OPD, EOPD, and REOPOLD, across mathematical reasoning, code generation, and general-domain benchmarks.

1 Introduction

Recent Large Reasoning Models (LRMs) (Zhang et al., 2025; Chen et al., 2024) improve performance by scaling test-time reasoning and have achieved 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. Conventional off-policy distillation (Kim and Rush, 2016) trains students to imitate outputs generated by strong teacher models 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 efficient approach for SRMs.

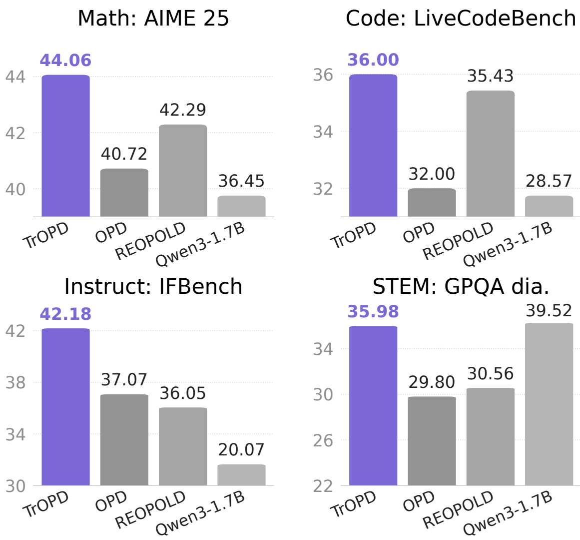

Figure 1: Performance of TrOPD and baselines. All OPD methods are trained from Qwen3-SFT-1.7B.

Despite their potential efficiency advantages, existing OPD methods often suffer from training instability due to unreliable supervision. When the teacher and student distributions diverge substantially, student-generated trajectories may fall outside the teacher's reliable supervision region, yielding erroneous policy gradients and potentially causing training collapse. Moreover, reasoning-oriented OPD cannot afford full-vocabulary supervision due to its prohibitive memory overhead (Agarwal et al., 2024). It therefore typically relies on KL divergence estimators (Lu et al., 2025),

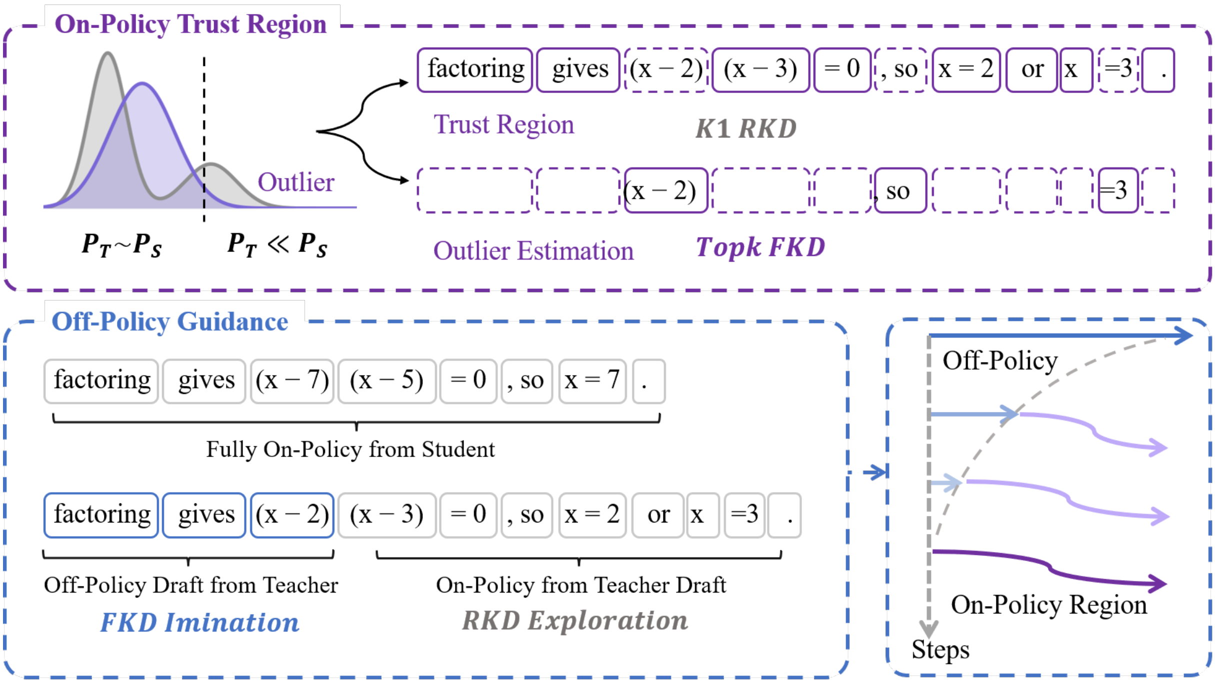

Figure 2: Overview of Trust Region On-Policy Distillation. For the on-policy component, student-generated tokens are divided into the trust region and outliers. The student model is further guided by teacher-generated responses.

which may further reduce the reliability of the supervision signal.

However, reliable OPD for reasoning tasks remains non-trivial. This work establishes a unified benchmark to systematically study this challenge from three perspectives: (1) multi-domain evaluation, covering mathematics, code generation, 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 to enable long-response distillation under practical memory constraints. The resulting evaluation reveals that existing methods fail to effectively suppress erroneous policy gradients. Furthermore, naive reward clipping, as adopted by REOPOLD (Ko et al., 2026), may remove informative supervision together with outlier gradients, resulting in a performance bottleneck.

To improve the reliability of teacher supervision, this work proposes Trust Region On-Policy Distillation (TrOPD), which partitions student-generated tokens according to their supervision reliability. As shown in Figure 2, TrOPD determines whether a token falls into a teacher-verifiable trust region according to the decoding agreement ratio between the teacher and student models. For outliers, we employ a top-k forward-KL estimator to preserve informative reward signals while avoiding unreliable policy gradients. To further encourage the student to generate within teacher-verifiable regions, we introduce off-policy guidance, which performs imitation learning from teacher-generated trajectories. As shown in Figure 1, TrOPD substantially improves 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 general benchmark for reasoning-oriented OPD and identify the supervision reliability issue in OPD.
- We propose Trust Region On-Policy Distillation (TrOPD), achieving high-quality and stable reasoning optimization.
- We train small reasoning models based on TrOPD, further advancing the reasoning capabilities of small language models.

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 (LLMs), initially elicited through reasoning prompts. More recently, reasoning capabilities have been acquired through reinforcement learning (Zhang et al., 2025); supervised finetuning

(Team, 2025a); and on-policy distillation. Reasoning is also increasingly integrated with other core capabilities of LLMs, such as agentic and multi-modal (AI, 2025; Bai et al., 2025) abilities. However, how to acquire strong reasoning capabilities for SRMs remains underexplored.

Knowledge Distillation. Knowledge distillation was originally introduced by Hinton et al. (Hinton et al., 2015) to efficiently train compact models. In recent years, distillation for generative language models has primarily relied on sequence-level knowledge distillation (Kim and Rush, 2016), an off-policy approach that performs supervised fine-tuning on teacher-generated responses. More recently, full-vocabulary on-policy distillation methods (Ko et al., 2024, 2025; Xu et al., 2025), such as MiniLLM (Gu et al., 2024) and GKD (Agarwal et al., 2024), have been developed to mitigate the exposure bias (Agarwal et al., 2024). For reasoning models, KL objectives based on the k_1 estimator (Lu et al., 2025; Yang et al., 2026) have been applied to effectively improve the reasoning performance in post-training stage.

3 Problem Formulation

3.1 Distillation for Language Models

Different from off-policy distillation trained from the teacher of generations (ToGs), On-policy distillation instead trains on student of generations (SoGs) to mitigate exposure bias. In reverse KL divergence (RKL)-based OPD, the sequence-level objective can be written as $D_{\mathrm{KL}}(\pi_S \| \pi_T) = \mathbb{E}_{x\sim\pi_S}\left[\log \frac{\pi_S(x)}{\pi_T(x)}\right]$, whose gradient naturally takes a policy-gradient form (Gu et al., 2024): the student samples trajectories from its own policy and is rewarded for generating sequences assigned high probability by the teacher. Since the expectation is taken over the student distribution, RKL strongly penalizes student outputs that fall into low-probability regions of the teacher, while imposing little direct penalty on teacher modes not explored by the student, thereby exhibiting mode-seeking behavior. However, conventional OPD computes RKL or Jensen–Shannon divergence (JSD) over the full vocabulary, making memory overhead a significant bottleneck for long generation tasks.

3.2 OPD for Reasoning Models

Recent on-policy distillation (OPD) methods employ token-level reverse KL as the reward and optimize it using policy gradient. However, computing reverse KL over the full vocabulary incurs $\mathcal{O}(n \cdot k)$ memory overhead, where n is the sequence length and k is the vocabulary size. Traditional instruction LLM distillation methods, such as GKD and speculative KD, compute KL divergence over the full vocabulary, achieving stable optimization:

$$\mathcal{J}^{\mathrm{KD}} = -\mathrm{RKL}(\pi_S \parallel \pi_T) = -\sum_{x\in\mathcal{V}} \pi_S \log \frac{\pi_S}{\pi_T}. \tag{1}$$

In contrast, recent reasoning-oriented models scale performance by extending the reasoning length, which significantly increases output sequence length and makes memory consumption a major bottleneck for model distillation. To address this issue, Thinking Machine Lab proposes using the K1 estimator to obtain an unbiased estimate of the KL divergence, leading to the following optimization objective:

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

However, the K1 estimator suffers from two key optimization bottlenecks:

Significant policy-gradient outliers. When the discrepancy between the teacher and student distributions is large, the teacher may assign extremely low probabilities to trajectories sampled from the student policy, i.e., $\mathbb{E}_{x\sim P_S}[\pi_T(x)] \approx 0$. In such low-confidence regions, the K1-based policy-gradient signal can become extremely negative, i.e., $\nabla\mathcal{J} = \frac{1}{\pi_S(x)} \log \frac{\pi_T(x)}{\pi_S(x)} \to -\infty$. Therefore, student-generated trajectories that receive extremely low confidence from the teacher induce significant policy-gradient outliers, which destabilize OPD optimization and limit its potential final performance.

Low-quality student of generation (SoG). Since OPD is optimized exclusively on trajectories sampled from the student policy, the student may struggle to generate high-quality responses for challenging problems. As a result, low-quality SoG trajectories restrict the effective optimization space and prevent the student from receiving informative supervision from higher-quality responses.

4 Trust Region Distillation

4.1 Benchmarking OPD Baselines

Compared with conventional OPD based on full-vocabulary distributions, OPD for long-thinking

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	$\sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{S,v})$	–	0.00	0.00	4.21	1.40
JSD	$\beta \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{M,v})$	$(1-\beta)\log(\pi_S/\pi_M)$	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	$\overline{\mathbb{M}} \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{S,v})$	$\mathbb{M} \log \pi_T/\pi_S$	39.16	29.89	77.96	49.00
TrOPD	$\overline{\mathbb{M}} \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log(\pi_{T,v}/\pi_{S,v})$	$\mathbb{M} \log \pi_T/\pi_S$	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.

reasoning models remains an emerging research direction. Existing studies are often conducted under different experimental configurations, making it difficult to directly compare recent SoTA methods. Therefore, there is an urgent need to benchmark recent OPD methods under a unified setting. In this section, we first evaluate representative OPD methods by focusing on two fundamental questions: (1) Do divergence objectives developed for full-vocabulary OPD remain effective for recent token-level OPD based on the K_1 estimator? (2) Under a unified experimental setting, how do existing advanced methods compare fairly in terms of performance and generalization?

Divergence Evaluation. Due to the asymmetry of KL divergence, previous full-vocabulary distillation methods typically adopt Forward KL (FKL) for mode covering and Reverse KL (RKL) for mode seeking. More generally, GKD (Agarwal et al., 2024) employs JSD to balance FKL and RKL. Under the memory constraints of long-thinking distillation, we implement FKL over the top-k tokens of the teacher distribution and implement RKL using the token-level K1 estimator. Specifically, the top-k FKL objective is defined as:

$$\mathcal{J}_{\mathrm{FKL}}^{\text{top-}k} = \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}}. \tag{3}$$

Given the definitions of top-k FKL and K1-based RKL, the generalized JSD objective with a balancing coefficient β can be written as:

$$\mathcal{J}_{\mathrm{JSD}}^{\beta} = \beta \sum_{v\in\mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{M,v}} + (1-\beta) \log \frac{\pi_{S,x}}{\pi_{M,x}}, \tag{4}$$

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

As shown in Table 1, stand-alone FKL can not achieve effective training when computed over only a small subset of the vocabulary. This is mainly because top-k FKL constitutes a biased approximation of the full-vocabulary FKL objective, and applying this biased divergence to all sampled tokens can introduce increasingly distorted policy gradients. Therefore, FKL is not suitable as the standalone objective for OPD under constrained vocabulary, while there would be great potential to use FKL enhancing OPD objectives like JSD.

Token Filtering and Reward Clipping. To mitigate erroneous policy gradients induced by outlier tokens, existing methods mainly adopt two strategies: entropy-based token filtering and reward clipping. In GRPO (Shao et al., 2024), training only on the top 20% high-entropy tokens is commonly used to suppress the interference of less informative tokens and accelerate RL convergence. REOPOLD (Ko et al., 2026) instead applies reward clipping to reduce the influence of erroneous token-level supervision signals. Specifically, it requires a predefined clipping threshold, and rewards exceeding this threshold are clipped to the corresponding upper bound during training.

As shown in Tables 1 and 4, entropy-aware token selection (Wang et al., 2026b) does not consistently benefit OPD: restricting optimization to high-entropy tokens often degrades performance. This suggests that, in the OPD setting, the teacher can also provide sufficiently informative supervision on ordinary tokens, which should not be discarded during training. Reward clipping improves

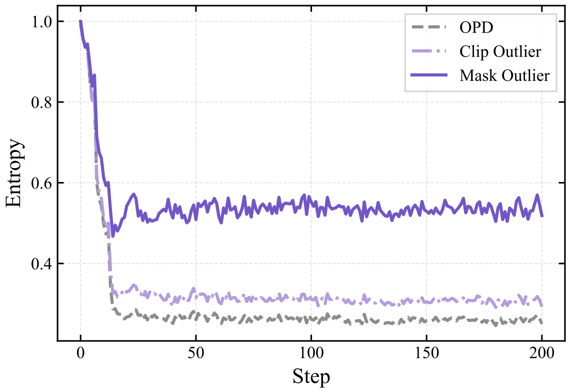

Figure 3: Entropy comparison of OPD methods.

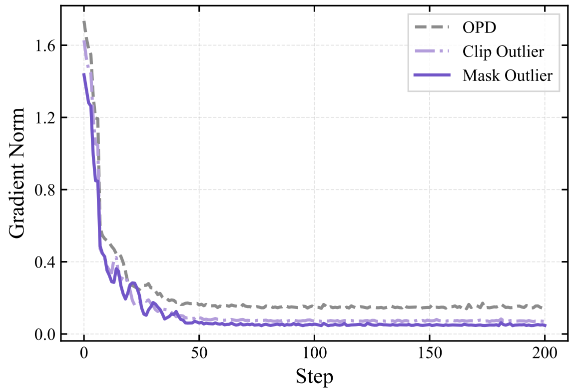

Figure 4: Gradient norm comparison of OPD methods.

model performance in Table 1; however, as shown in Table 4, its gains become marginal under other settings. Moreover, the choice of clipping threshold introduces an additional hyperparameter that may substantially affect the learning dynamics and convergence speed.

4.2 Trust-Region On-Policy Learning.

Based on the benchmark results, directly applying conventional FKL or KL objectives defined on intermediate mixture distributions fails to effectively address the erroneous policy gradients caused by the 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:

$$\begin{aligned}\mathcal{J}_x^{On} &= -\mathbb{M}_x \mathrm{KL}(\pi_S \parallel \pi_T) - \overline{\mathbb{M}}_x \mathrm{KL}(\pi_T \parallel \pi_S) \\ &= -\mathbb{M}_x \log \frac{\pi_S}{\pi_T} - \overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}}.\end{aligned} \tag{5}$$

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 exceeding 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 3 and 4, *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, $\mathbb{M}_x \sim \mathrm{Bernoulli}(P_{\text{trust}}(x))$, is defined as:

$$P_{\text{trust}}(x) = \min\left(\frac{\pi_T(x)}{\pi_S(x)}, 1\right). \tag{6}$$

This design is motivated by speculative decoding, where the probability that the teacher model agrees with a token decoded by the student model satisfies $P_{\text{accept}}(x) \propto \min\left(\frac{\pi_T(x)}{\pi_S(x)}, 1\right)$. 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.

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

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.

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:

$$\mathcal{J}_x^{FKL} = -\overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T(v)} \log \frac{\pi_{T(v)}}{\pi_{S(v)}}, \tag{7}$$

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 $\sum_{v \in \mathcal{V}_T^k} \pi_S(v) \to 0$, 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:

$$\begin{aligned}\mathcal{J}_x &= -\beta \mathrm{KL}_{x[:l] \sim \pi_T}(\pi_T \parallel \pi_S) + \mathcal{J}_{x[l:]}^{\mathrm{On}} \\ &= -\beta \mathbb{I}[x \sim \pi_T] \log \frac{\pi_T}{\pi_S} + \mathbb{I}[x \sim \pi_S] \mathcal{J}_{x[l:]}^{\mathrm{On}}\end{aligned} \tag{8}$$

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:

$$\begin{aligned}\mathcal{J}_x^{\mathrm{TrOPD}} &= -\mathbb{I}[x \sim \pi_S] \overline{\mathbb{M}}_x \sum_{v \in \mathcal{V}_k^T} \pi_{T,v} \log \frac{\pi_{T,v}}{\pi_{S,v}} \\ &- \mathbb{I}[x \sim \pi_T] \mathbb{M}_x \log \frac{\pi_S}{\pi_T} - \beta\, \mathbb{I}[x \sim \pi_T] \log \frac{\pi_T}{\pi_S},\end{aligned} \tag{9}$$

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)

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.

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 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 and sample 4 rollouts for each prompt, with a maximum generation length of 8096 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).

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 results on out-of-

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

Method	AIME 24	AIME 25	AMC 23	Avg.
Single-Domain Distillation				
DS-1.5B	28.64	24.16	71.01	41.27
OPD	35.83	29.16	75.39	46.79
+ Outlier Estimation				
Mask Outlier	37.08	30.62	75.46	47.72
Clip Outlier	36.97	30.83	75.78	47.86
Full FKL	0.00	0.00	4.21	1.40
FKL Outlier	39.16	29.89	77.96	49.00
+ Off-Policy Guidance				
TrOPD Mask	40.10	30.41	75.85	48.79
TrOPD Clip	37.39	31.77	77.03	48.73
TrOPD FKL	38.54	32.50	78.51	49.85

Table 6: Ablation Studies of TrOPD.

domain (OOD) tasks to assess the continual learning capability of OPD methods and their robustness to domain shifts. Compared with OPD, TrOPD improves the average performance by +3.06 points on mathematical reasoning tasks and by +2.63 points on general-domain tasks. REOPOLD corrects unreliable policy gradients using simple reward clipping; nevertheless, TrOPD outperforms it by 1.99 and 1.84 points in the mathematical and general domains, respectively, demonstrating 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. These results indicate that outlier-aware token selection provides 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 mathematical reasoning and code generation, we mainly 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 results demonstrate that TrOPD can consistently improve distillation performance across different teacher–student configurations and diverse reasoning tasks.

5.3 Ablation Studies and Discussion

As shown in Table 6, 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 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.

We also notice the concurrent work AOPD (Jia et al., 2026). As shown in Table 5, 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 combining complementary OPD optimization strategies is a promising direction for future work.

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 effectively suppresses unreliable policy gradients while preserving informative supervision. We further introduce off-policy guidance to encourage exploration toward teacher-supported trajectories. Extensive multi-domain results highlight the importance of supervision reliability in on-policy reasoning distillation and demonstrate the future potential of trust-region learning for training high-quality small reasoning models.

Limitations

The primary limitation of this work is the lack of practical deployment and application studies on small reasoning models. In real-world scenarios, training high-performing small reasoning models often requires incorporating mid-training to further improve their post-training reasoning capabilities. This work focuses primarily on OPD-based post-training using DeepSeek-Qwen2.5-1.5B and Multi-Domain Distillation, which may constrain the upper bound of the resulting reasoning performance. Future work should investigate how additional stages, such as mid-training, can further enhance small reasoning models in practical deployment settings. Nevertheless, this study focuses on fair and controlled comparisons among OPD methods.

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*.

Moonshot AI. 2025. Kimi-researcher: End-to-end rl training for emerging agentic capabilities. https://moonshotai.github.io/Kimi-Researcher/. Accessed: 2025-08-13.

Anthropic. 2025. Claude 3.7 sonnet and claude code.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, and 1 others. 2025. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*.

Bytedance-Seed-Foundation-Code-Team, :, Yao Cheng, Jianfeng Chen, Jie Chen, Li Chen, Liyu Chen, Wentao Chen, Zhengyu Chen, Shijie Geng, Aoyan Li, Bo Li, Bowen Li, Linyi Li, Boyi Liu, Jiaheng Liu, Kaibo Liu, Qi Liu, Shukai Liu, and 37 others. 2025. Fullstack bench: Evaluating llms as full stack coders. *Preprint*, arXiv:2412.00535.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *Preprint*, arXiv:2412.18925.

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, and 1 others. 2025. 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)*, pages 5069–5096.

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. 2025. Robin: A multi-agent system for automating scientific discovery.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

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. 2025a. Skywork open reasoner series. Notion Blog.

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

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

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

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

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

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

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

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

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

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, and 7 others. 2022. Competition-level code generation with alphacode. *arXiv preprint arXiv:2203.07814*.

Kevin Lu and 1 others. 2025. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation/. Thinking Machines Blog, accessed on 2025-10-27.

NVIDIA Corporation. 2025. Opensciencereasoning-2 dataset. Hugging Face Dataset. 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. 2025. Codeforces cots. https://huggingface.co/datasets/open-r1/codeforces-cots.

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

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, and 1 others. 2025. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*.

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*.

Yexuan Shi, Mingyu Wang, Yunxiang Cao, Hongjie Lai, Junjian Lan, Xin Han, Yu Wang, Jie Geng, Zhenan Li, Zihao Xia, and 1 others. 2025. Aime: Towards fully-autonomous multi-agent framework. *arXiv preprint arXiv:2507.11988*.

Kimi Team. 2025a. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Qwen Team. 2025b. Qwq-32b: Embracing the power of reinforcement learning.

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2026b. 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.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arxiv preprint arXiv: 2505.09388*.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao, Zhanfeng Mo, Qi Zhang, and 1 others. 2025. 100 days after deepseek-r1: A survey on replication studies and more directions for reasoning language models. *arXiv preprint arXiv:2505.00551*.

Changsheng Zhao, Ernie Chang, Zechun Liu, Chia-Jung Chang, Wei Wen, Chen Lai, Sheng Cao, Yuandong Tian, Raghuraman Krishnamoorthi, Yangyang Shi, and 1 others. 2025. Mobilellm-r1: Exploring the limits of sub-billion language model reasoners with open training recipes. *arXiv preprint arXiv:2509.24945*.

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

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 upsampled 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. Multi-task RLVR training is conducted for 1000 steps using a constant learning rate of 2×10^{-6}. The Adam optimizer is used with a weight decay of 0.1.

For reward verification, the evaluator released with Qwen QwQ-32B (Team, 2025b) is used for mathematical reasoning tasks. The IFEvalG verifier (Lambert et al., 2024) is used for instruction-following tasks. For coding tasks, Sandbox-Fusion (Bytedance-Seed-Foundation-Code-Team et al., 2025) serves as the execution sandbox for obtaining unit-test outcomes.

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

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