DISTILLING STRONG TEACHERS INTO SMALL MODELS: TEACHER-PREFERRED DISTRIBUTION ALIGNMENT MATTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as an effective approach for transferring post-trained capabilities from strong teacher models to students because the teacher and student can exhibit substantial distributional mismatch. However, in cross-origin settings, the substantial distributional mismatch between the teacher and student can make sampled-token OPD produce unreliable gradients whose directions fail to correct the student's suboptimal reasoning behavior. Exploiting teacher-preferred token supervision becomes critical. In this work, we systematically compare three teacher-preferred-token supervision strategies within the OPD framework: Online SFT, Teacher Top-K, and Hybrid Teacher Top-K. Across our experiments, Teacher Top-K consistently outperforms the other two strategies. Although Teacher Top-K OPD is conceptually simple, its effectiveness depends critically on several implementation and design choices. In particular, the REINFORCE-style loss must use the renormalized truncated teacher distribution to match the RKL objective, renormalization is essential to avoid the mass-shrink shortcut, and K controls the teacher-support coverage. Our analysis leads to a simple recipe, **Teacher Top-K OPD with renormalization**, which achieves SOTA performance across mathematical reasoning and code generation. Scaling experiments show that increasing training compute improves student performance, while stronger teachers yield a higher performance ceiling. Notably, our Qwen3-14B student outperforms the official Qwen3-14B model by **14.7** points on AIME24/25 and AMC23. These findings show that Teacher Top-K provides a scalable and effective solution for cross-origin distillation from strong teachers to smaller students.

1 INTRODUCTION

Recently, on-policy distillation (OPD) (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025) has emerged as a core component of LLM post-training, enabling efficient knowledge transfer (Team et al., 2026; Xiao et al., 2026; Yang et al., 2025). OPD samples trajectories from the student's current policy and queries the teacher to provide dense, token-level supervision at the prefixes along these trajectories. The resulting dense supervision facilitates the efficient transfer of capabilities acquired by the teacher during post-training. Qwen3 achieves 74.4% on AIME24 with OPD using 1,800 GPU hours, versus 67.6% with RL using 17,920 GPU hours (Yang et al., 2025). Recent model releases, such as Kimi K3 (Team et al., 2026) and MiMo V2 Flash (Xiao et al., 2026), employ multi-teacher OPD to integrate complementary capabilities within a unified model.

While these gains have been demonstrated primarily for *same-origin* teacher–student pairs (Ma et al., 2026; Yang et al., 2026a), OPD can become substantially less effective when the teacher and student originate from different base models (Li et al., 2026b; Oh et al., 2026; Ma et al., 2026). Li et al. (2026a) find that a stronger teacher does not improve distillation performance, which they attribute to the anisotropy of token-level distillation signals from larger teacher models. These observations suggest that stronger teacher capabilities do not automatically translate into larger student gains, motivating us to study when and how such capabilities can be effectively transferred. Motivated by this gap, we investigate how teacher-student distribution mismatch affects token-level supervision and how this supervision can be designed to better transfer the capabilities of stronger teachers. This leads to our central question: ***How can smaller students learn effectively from stronger teachers?***

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.

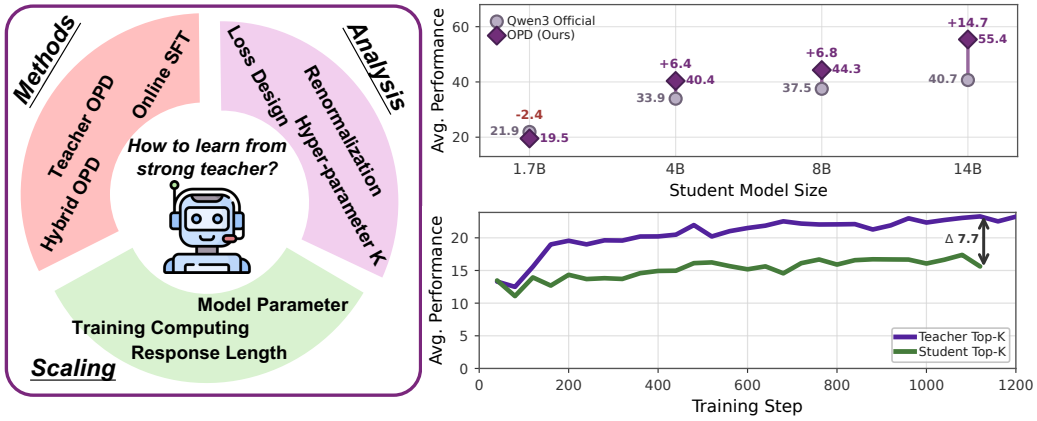


Figure 1: **Left**: Overview of our study. We investigate the distribution mismatch problem in cross-origin OPD, identify the key factors and scaling properties. **Right**: Our recipe achieves SOTA than baselines and distilled Qwen3-14B student outperforms the official Qwen3-14B by **14.7** points. Further, increasing training compute improves performance.

To answer the question, we first analyze the key difference between *same* and *cross-origin* OPD using student-preferred OPD. As shown in Figure 2 (Left), OPD rapidly approaches teacher-level performance in the same-origin setting but yields little improvement in the cross-origin setting. Analysis of the teacher–student token distributions in Figure 2 (Middle) and Figure 2 (Right) shows that same-origin teachers share and sharpen the student's preferred modes, whereas cross-origin teachers favor different modes from student. In cross-origin settings, sampled-tokens receive low probability under the teacher due to different modes. Sampled-token OPD thus suppresses these tokens without directly supervising which teacher-preferred tokens should gain probability mass, limiting knowledge transfer. This motivates retaining student-generated prefixes while shifting token-level supervision toward teacher-preferred tokens.

In principle, replacing student-preferred OPD with the teacher's Top-K distribution should provide an informative signal under cross-origin mismatch. While Fu et al. (2026) report effective Teacher Top-K supervision, Ma et al. (2026) observe instability under cross-origin mismatch, raising the question of what governs its effectiveness. Furthermore, can we design a more effective teacher-preferred distillation method? Inspired by Dagger (Ross et al., 2011), we propose Online SFT, Hybrid OPD and compare with Teacher Top-K OPD. *Teacher Top-K OPD achieves the best performance, while Hybrid OPD consistently degrades as the contribution of the student distribution increases.* We further identify three factors that critically affect Teacher Top-K OPD: (i) REINFORCE-style loss must use the renormalized truncated teacher distribution to match the RKL objective, (ii) renormalization is essential to avoid the mass-shrink shortcut, and (iii) K controls the teacher-support coverage. These analyses lead to a simple yet effective recipe: **Teacher Top-K OPD with renormalization**. Using Qwen3-8B-Base as student model, Teacher Top-K OPD consistently outperforms the corresponding baselines by up to **5.9** points. And Stronger teachers and stronger students tend to yield larger distillation gains. Notably, our Qwen3-14B student model outperforms the official Qwen3-14B by **14.7** points under the non-thinking setting. We summarize our main contributions as follows:

Contribution

- We characterize teacher–student distribution mismatch in cross-origin OPD and identify unstable gradient signals as a key limitation of student-preferred token supervision.
- We propose two teacher-preferred distillation methods and compare with Teacher Top-K.
- We identify implementation and design choices that substantially affect Teacher Top-K OPD, offering a possible explanation for differing observations in prior work.
- We demonstrate consistent gains across baselines, model scales, and different benchmarks. We also show that the recipe benefits from increased training compute and larger teacher models.

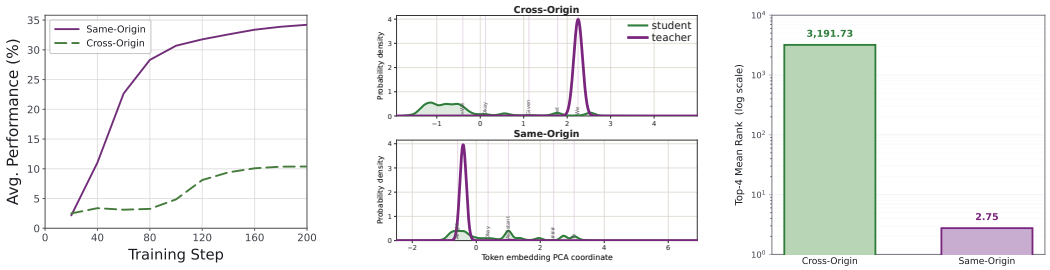


Figure 2: **Limitations of student-preferred OPD under cross-origin mismatch. Left:** Student-preferred OPD improves validation accuracy in the same-origin setting but yields little improvement across origins. **Middle:** Same-origin teachers sharpen an existing student preference mode, whereas cross-origin teachers favor a shifted mode. **Right:** In cross-origin pairs, the student's Top-K tokens rank far lower under the teacher, revealing a substantial gap in token preferences.

2 Preliminaries

On-policy distillation (OPD) trains the student on its own generated trajectories, inheriting the on-policy nature of policy training while providing dense token-level credit assignment, which makes it an efficient post-training paradigm (Lu & Thinking Machines Lab, 2025; Xiao et al., 2026). Given an input prompt $x \sim \mathcal{D}$, the student samples a response $y \sim \pi_\theta(\cdot \mid x)$, and OPD minimizes the reverse KL divergence from the student policy π_θ to the teacher policy π_T:

$$\mathcal{J}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[D_{\mathrm{KL}} \left(\pi_\theta(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x) \right) \right].$$

This on-policy formulation allows the teacher to provide supervision on the states the student actually visits during training. In practice, current OPD methods often use sampled-token OPD as the approximation:

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

Under this approximation, the negative log-ratio can be viewed as token-level advantage. Many existing methods optimize the resulting objective through a REINFORCE-style loss, using this advantage to weight the log probability of each student-sampled token, because this formulation integrates naturally with existing RL training frameworks.

3 Failure Modes of Student-Guided OPD

Takeaway

- Same-origin teachers largely share the student's preference distribution and sharpen its existing modes, whereas cross-origin teachers induce a shifted preference mode.

We use Student Top-K OPD to examine how knowledge transfer differs between *same-origin* and *cross-origin* teacher–student pairs. Specifically, we apply OPD to distill a Qwen3-4B-GRPO (Li et al., 2026a) teacher into a Qwen3-4B-Base student, and a Qwen3-4B (Yang et al., 2025) teacher into a Qwen3-1.7B-Base student. The same-origin setting trains stably and successfully transfers the teacher's capability into student model (Figure 2 (Left)), whereas the cross-origin setting fails: generated responses become progressively longer throughout training (Figure 7) and eventually collapse into repetitive outputs (see Appendix G for examples).

To understand the failure reasoning of student-preferred token optimization, we analyze the teacher–student token-level mismatch on student-generated trajectories. First, we visualize the token distributions for a specific example in Figure 2 (Middle). For same-origin teacher–student pairs, the teacher distribution is aligned with the student distribution and can be viewed as a sharpened version

of it. In the cross-origin setting, however, the teacher assigns its probability mass to a different set of high-preference tokens, resulting in a shifted preference mode. We further quantify average teacher rank of student Top-4 tokens across the entire response with 128 responses. Their average rank increases from 2.75 in the same-origin setting to 3,191.73 in the cross-origin setting, which also indicates that teacher-preferred tokens consistently fall outside the student's high-probability region. Under such distribution mismatch, student top-K tokens merely reduce the probabilities of the student's preferred tokens without providing a direction toward the teacher distribution. This motivates constructing the learning signal from teacher-preferred tokens, which explicitly indicate which tokens should receive higher probability.

4 Teacher-Preferred Distribution Alignment Methods

Takeaway

- Teacher Top-K OPD consistently outperforms Online SFT, suggesting that preserving the teacher's soft preference structure is beneficial.
- Hybrid Teacher Top-K improves performance as the weight assigned to the teacher distribution increases, with the pure Teacher Top-K target achieving the best performance.

In this section, we investigate the behavior of different distribution alignment methods. We propose Online SFT, Hybrid OPD and compare with Teacher Top-K OPD.

Online SFT A straightforward approach is supervised fine-tuning (SFT) on teacher-generated responses, but offline SFT suffers from compounding error (Xu et al., 2020) because the student is trained only on teacher trajectories. Inspired by DAgger, we consider online SFT. Online SFT queries the teacher at states drawn from this distribution and trains the student to predict the teacher's highest-probability next token. Formally, given a student-generated trajectory $y \sim \pi_\theta(\cdot \mid x)$, the teacher utilizes the highest-probability token as target token

$$y_t^T = \arg\max_v \pi_T(v \mid x, y_{<t}),$$

and Online SFT uses the teacher's target token and minimizes the following loss:

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

Teacher Top-K OPD Online SFT aligns the student's preferred action with the teacher throughout the student-visited region, but one-hot label cannot align their complete next-token distributions. A natural approach to distribution alignment is to supervise the student with the teacher's soft probabilities over a candidate set. We consider Teacher Top-K OPD, which constructs the supervision signal directly from the teacher's K highest-probability tokens. Specifically, letting $h_t = (x, y_{<t})$, we define the teacher Top-K support and the corresponding renormalized distributions as

$$\mathcal{S}_K(h_t) = \text{TopK}_v\, \pi_T(v \mid h_t), \qquad \hat{\pi}_i(v \mid h_t) = \frac{\pi_i(v \mid h_t)}{\sum_{u\in\mathcal{S}_K(h_t)} \pi_i(u \mid h_t)}, \quad i \in \{\theta, T\},$$

We then minimize a truncated reverse KL over this support:

$$\mathcal{L}_{\text{TopK}} = \mathbb{E}_{x,y\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{v\in\mathcal{S}_K(x,y_{<t})} \hat{\pi}_\theta(v \mid x, y_{<t}) \log \frac{\hat{\pi}_\theta(v \mid x, y_{<t})}{\hat{\pi}_T(v \mid x, y_{<t})}\right].$$

Hybrid Top-K OPD Directly matching the teacher distribution can be difficult when the teacher–student gap is large. To reduce this learning difficulty, we further consider Hybrid Top-K OPD, which constructs an intermediate target by mixing the teacher distribution with the student's current distribution. This target remains closer to the student while retaining an optimization direction toward the teacher. At each student-generated prefix, we construct the student-selected support $\mathcal{S}_K^{\pi_\theta}(x, y_{<t})$ and renormalize both distributions on this support. We then define the detached mixture target

$$\tilde{\pi}_\alpha(v \mid x, y_{<t}) = \alpha\, \hat{\pi}_T^{\pi_\theta}(v \mid x, y_{<t}) + (1-\alpha)\, \text{sg}[\hat{\pi}_\theta^{\pi_\theta}(v \mid x, y_{<t})], \quad v \in \mathcal{S}_K^{\pi_\theta}(x, y_{<t}),$$

where $\alpha \in [0, 1]$ controls the strength of teacher supervision and $\mathrm{sg}[\cdot]$ denotes stop-gradient. The target interpolates between pure teacher matching ($\alpha = 1$) and the student's detached current prediction ($\alpha = 0$). It therefore acts as a proximal soft target that moderates overly aggressive updates when the teacher–student gap is large. The corresponding objective is

$$\mathcal{L}_{\text{HybridTopK}} = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{v\in\mathcal{S}_K^{\pi_\theta}(x,y_{<t})}\hat{\pi}_\theta^{\pi_\theta}(v \mid x, y_{<t})\log\frac{\hat{\pi}_\theta^{\pi_\theta}(v \mid x, y_{<t})}{\tilde{\pi}_\alpha(v \mid x, y_{<t})}\right].$$

4.1 Comparison of Distribution Alignment Strategies

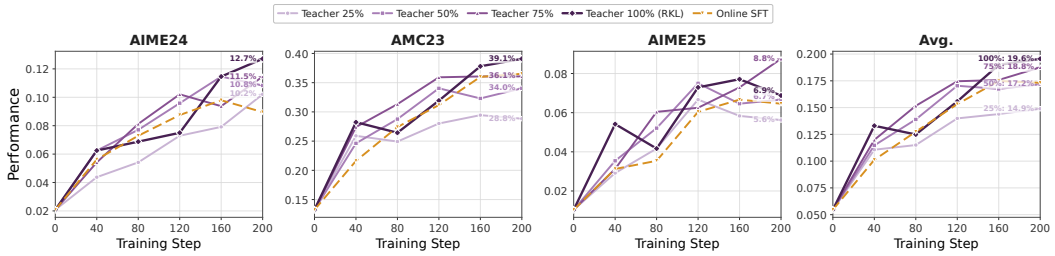


Figure 3: Validation performance of different teacher-preferred alignment strategies. We compare Online SFT with Hybrid Teacher Top-K OPD using teacher mixture ratios of 25%, 50%, 75%, and 100%. Overall, validation performance consistently improves as the teacher mixture ratio increases.

Figure 3 compares Online SFT with Hybrid Top-32 OPD under different teacher mixture ratios. We vary the proportion of the teacher distribution in the hybrid target over 25%, 50%, 75%, and 100%. The 100% setting, which corresponds to Teacher Top-32 OPD without mixing in the student distribution, achieves the best result. We report the $\mathrm{Avg@16}$ performance across AIME24, AIME25, and AMC23. Performance improves consistently as the teacher ratio increases, rising from 14.9% at a 25% teacher ratio to 19.6% with the full teacher distribution. We also find that Online SFT performs slightly worse than Teacher Top-K OPD. We attribute this gap to the one-hot supervision used by Online SFT, which discards useful relative probabilities among plausible teacher tokens. Overall, increasing reliance on the teacher distribution produces stronger alignment, and directly matching the Teacher Top-K distribution is more effective than Hybrid Top-K OPD and Online SFT.

5 Analysis of Key Factors on Teacher Top-K

In this section, we investigate what matters for effective Teacher Top-K OPD by studying three key factors: the loss formulation, renormalization of the Top-K distribution, and the choice of K.

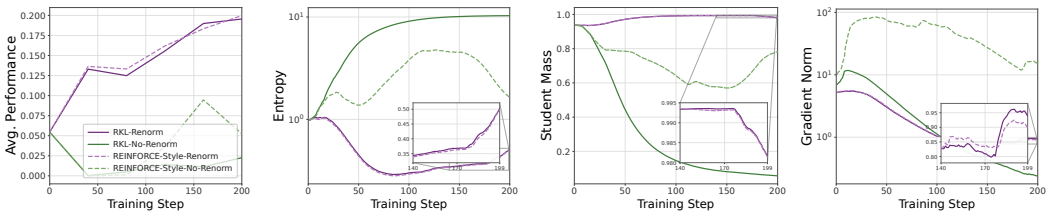
5.1 REINFORCE-Style vs. Direct Reverse KL Loss

Takeaway

- REINFORCE-style loss is equivalent to direct reverse KL *only with* renormalization.

In practice, Top-K OPD is typically implemented in two ways: direct reverse-KL (RKL) loss and REINFORCE-style surrogate. RKL can be implemented as a REINFORCE-style surrogate (Gu et al., 2024; Li et al., 2026a). Following Li et al. (2026a), we implement Teacher Top-K OPD and find that its performance is sensitive to the normalization convention used in the surrogate. Our derivation shows that the REINFORCE-style formulation is equivalent to direct RKL only when the distributions are renormalized. We provide a detailed gradient analysis in Appendix C.1. This motivates a comparison of three implementations of Teacher Top-K: direct RKL, the REINFORCE-style surrogate with renormalization, and its unnormalized variant[1]. As shown in Figure 4, when

[1]Our unnormalized REINFORCE-style implementation follows the open-source implementation of Li et al. (2026a)



(a) Average performance. (b) Policy entropy. (c) Mass on Top-K. (d) Gradient norm.

Figure 4: Effect of renormalization under Teacher Top-K for both REINFORCE-style and direct reverse KL. With renormalization, the REINFORCE-style objective closely matches direct reverse KL and remains stable. Without renormalization, direct reverse KL admits a mass-shrinking solution, where the student can reduce the loss by decreasing the total probability mass assigned to the teacher's Top-K support rather than improving distributional alignment.

renormalization is applied, REINFORCE-style and direct RKL exhibit nearly identical performance and training dynamics. For the REINFORCE-style objective, removing renormalization reduces performance from 20.22% to 5.75%. The gradient norm grows to about 100 times that of the renormalized variant and fluctuates during training.

5.2 Renormalization in RKL Loss

Takeaway

- In RKL Loss, Renormalization is necessary to avoid shrinking probability issue.

Prior studies report differing outcomes for Teacher Top-K supervision (Fu et al., 2026; Ma et al., 2026). Motivated by these observations, we examine renormalization as one factor that can affect its behavior. In our controlled experiments, we find the key difference lies in whether the RKL loss is renormalized. Our analysis shows that, without renormalization, the model can exploit a mass-shrinking shortcut instead of properly matching the teacher distribution.

Mass-Shrinking without Renormalization

Proposition 1. *Let S be the teacher's Top-K support. Define the student and teacher masses on S as $Z^\theta = \sum_{i\in S} p_i$ and $Z^T = \sum_{i\in S} q_i$, and define the corresponding conditioned distributions as $\hat{\pi}_\theta(i) = p_i/Z^\theta$ and $\hat{\pi}_T(i) = q_i/Z^T$. The non-renormalized objective decomposes as*

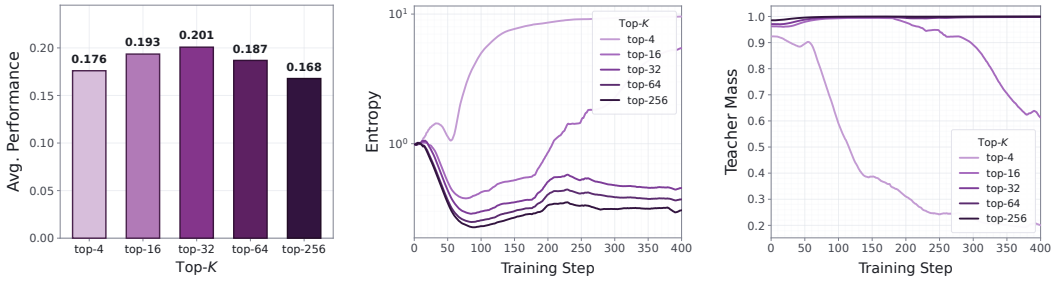
$$L_{\text{non-renorm}} = Z^\theta \left[D_{\text{KL}}(\hat{\pi}_\theta \| \hat{\pi}_T) + \log \frac{Z^\theta}{Z^T} \right].$$

Proposition 1 shows that the non-renormalized objective contains an intrinsic incentive to decrease the student mass on the teacher's support. The model can reduce the loss without improving its alignment with the teacher. Renormalization removes this dependence on Z^θ, forcing optimization to act on the relative distribution within the teacher's Top-K support. Appendix D provides the full derivation. We ablate the effect of renormalization in the RKL loss. The empirical results in Figure 4 confirm this prediction. Removing renormalization reduces performance from 19.55% to 2.26%. The student probability mass assigned to the teacher's Top-K support also collapses from 98.79% with renormalization to only 4.04% without renormalization. Accordingly, we use the renormalized direct reverse-KL loss throughout this paper unless otherwise specified.

5.3 Effect of the Support Size K

Takeaway

- A moderate K (e.g., K=32 in our setting) works best: both too-small and too-large supports degrade distillation quality.



(a) Average performance. (b) Policy entropy. (c) Mass.

Figure 5: Effect of scaling the teacher Top-K. (a) Top-$K = 32$ achieves the best average performance across AMC23, AIME24, and AIME25, while both smaller and larger values lead to lower performance. (b) Smaller teacher Top-K values, $K = 4$ and $K = 16$, result in substantially higher policy entropy, whereas the entropy remains low and stable for $K \geq 32$. (c) Teacher probability mass covered by the Top-K support increases with K.

We then ablate the teacher support size over $K \in \{4, 16, 32, 64, 256\}$. As shown in Figure 5, $K = 32$ achieves the best average performance across AMC23, AIME24, and AIME25, whereas both smaller and larger values degrade results. An overly small K is particularly harmful: under renormalization, the student mass on the teacher support keeps decreasing and policy entropy rises rapidly (Figure 5b). We attribute this failure to the fact that a too-small K covers too little probability mass, so the renormalized objective can be hacked by adjusting only the conditional shape on S rather than learning the teacher's distribution; more discussion is provided in Huang (2026). Conversely, when K is too large, performance also drops. We conjecture that this is due to learning difficulty: a large support admits many low-probability teacher tokens into the objective, diluting the supervision signal with uninformative tail tokens. These results suggest that Teacher Top-K works best with a moderately sized high-probability support that balances coverage and sharpness.

6 Practical Recipe

Based on the analyses above, we arrive at a simple practical recipe for cross-origin distillation: **Teacher Top-K OPD with a renormalized RKL loss**. Unless otherwise specified, we set $K = 32$ and refer to this method simply as **Teacher Top-K OPD**. In this section, we evaluate this recipe across multiple teacher–student pairs and model scales to assess its effectiveness and robustness.

We evaluate Teacher Top-K OPD across multiple model scales on mathematical reasoning and code generation. Specifically, we distill Qwen3-1.7B-Base from Qwen3-4B-Non-Thinking in the non-thinking setting and from a separately trained Qwen3-4B-Coding model in the coding setting (Yang et al., 2025; Hou et al., 2026). We further consider Qwen3-1.7B-Base and Qwen3-8B-Base as students distilled from Qwen3-30B-A3B-Instruct-2507. We use DAPO-Math-17K (Yu et al., 2025) for mathematical reasoning and Eurus-RL-Code (Cui et al., 2025) for code generation.

All methods are implemented in G-OPD (Yang et al., 2026a). We evaluate mathematical reasoning on AIME24, AIME25, and AMC23 using avg@32, and code generation on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench+ (Jain et al., 2025) using avg@4. We compare against Sample OPD (Gu et al., 2024; Xiao et al., 2026; Lu & Thinking Machines Lab, 2025) and Student Top-K OPD (Li et al., 2026a). Full training and evaluation details are provided in Appendix A.

As shown in Table 1, Teacher Top-K OPD consistently achieves the best average performance across all teacher–student pairs and both task domains. It substantially outperforms both Sample OPD and

Student Top-K OPD, with gains over Sample OPD reaching up to 14.08 points on mathematical reasoning and 10.55 points on code generation.

Table 1: Performance on math reasoning and code generation benchmarks. Bold indicates the best result for each benchmark. Human+ denotes HumanEval+. LCB denotes LiveCodeBench.

Method	Math Reasoning				Code Generation			
	AIME24	AMC23	AIME25	Avg.	Human+	MBPP+	LCB	Avg.
Student: Qwen3-1.7B-Base **Teachers:** Qwen3-4B (Math), Qwen3-4B-Coder (Code)								
Teacher	22.50	61.00	18.30	33.90	85.37	71.76	26.00	61.04
Student	2.08	13.25	1.04	5.46	7.16	4.10	1.00	4.09
Sample OPD	10.00	33.28	4.58	15.96	57.60	55.09	4.71	39.13
Student Top-K OPD	8.96	35.02	5.21	16.39	58.08	56.22	3.43	39.24
Online SFT	8.13	31.10	5.21	14.81	48.78	47.88	**13.71**	36.79
Hybrid OPD	8.54	31.85	5.83	15.41	54.12	52.23	12.14	39.50
Teacher Top-K OPD	**10.63**	**35.17**	**8.13**	**17.97**	**61.28**	**57.14**	13.14	**43.85**
Student: Qwen3-1.7B-Base **Teacher:** Qwen3-30B-A3B-Instruct-2507								
Teacher	73.96	91.27	62.29	75.84	83.38	75.58	33.86	64.27
Student	2.08	13.25	1.04	5.46	7.16	4.10	1.00	4.09
Sample OPD	8.75	26.88	3.96	13.20	43.60	49.67	1.86	31.71
Student Top-K OPD	7.29	28.38	6.04	13.90	51.83	53.51	2.71	36.02
Online SFT	8.96	36.52	6.46	17.31	42.38	43.92	6.71	31.00
Hybrid OPD	10.83	34.04	6.67	17.18	42.84	44.58	8.86	32.09
Teacher Top-K OPD	**12.71**	**39.08**	**6.88**	**19.55**	**57.62**	**55.29**	**13.86**	**42.26**
Student: Qwen3-8B-Base **Teacher:** Qwen3-30B-A3B-Instruct-2507								
Teacher	73.96	91.27	62.29	75.84	83.38	75.58	33.86	64.27
Student	11.88	38.18	9.79	19.95	36.30	36.80	6.00	26.37
Sample OPD	22.08	52.11	18.96	31.05	74.39	**73.02**	5.60	51.00
Student Top-K OPD	30.21	62.73	25.00	39.31	75.15	71.69	6.14	50.99
Online SFT	35.42	**70.63**	28.96	45.00	74.70	69.25	18.43	54.13
Hybrid OPD	**36.04**	70.03	27.08	44.39	76.52	70.90	**22.57**	56.66
Teacher Top-K OPD	35.42	70.18	**29.79**	**45.13**	**77.29**	71.43	22.00	**56.91**

7 Scaling Properties of Teacher Top-K OPD

7.1 Scaling Parameter Size

Takeaway

- OPD depends on the joint scaling of teacher capability and student capacity: larger students benefit more from stronger teachers, while small students can be capacity-limited in absorbing additional teacher knowledge.
- With a strong teacher, Teacher Top-K OPD enables the student to surpass the corresponding official Qwen3 model under the non-thinking setting, even with only 200 training steps.

Figure 6 studies the scaling behavior of OPD from two complementary axes. First, when comparing different student sizes after 200 steps, we observe that larger students consistently obtain stronger final performance. Moreover, the benefit from stronger teachers becomes more pronounced as the student capacity increases. For example, the 1.7B student only shows a modest gain when scaling the teacher, whereas the 8B student improves substantially when distilled from the strongest Qwen3-30B-A3B-Instruct-2507 teacher. This suggests that small students may be capacity-limited and cannot fully absorb the additional knowledge provided by stronger teachers.

Second, we study how student model scale affects the benefit of distillation. Using Qwen3-30B-A3B-Instruct-2507 as the teacher, we distill Qwen3 base models of different sizes and compare the resulting students with their corresponding official Qwen3 models. Under the non-thinking setting,

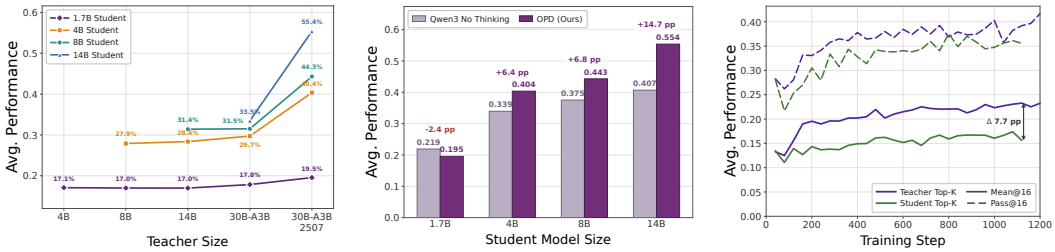


Figure 6: **Left**: Scaling effects of teacher capability and student capacity in on-policy distillation. Larger students achieve better OPD performance and benefit more from stronger teachers. **Middle**: With a strong teacher, Teacher Top-K OPD enables the student to surpass official Qwen3 model under non-thinking setting, even with only 200 steps. **Right**: Scaling training for the 1.7B student.

once the student reaches a sufficient scale (above 1.7B), OPD consistently yields gains of more than 6 percentage points over the corresponding official model. Notably, starting from Qwen3-14B-Base, OPD produces a 14B student that substantially outperforms the official Qwen3-14B model[2]. Moreover, the gain increases steadily with student size, suggesting that larger students have greater capacity to absorb the knowledge encoded in the teacher.

7.2 Scaling with Training Compute

Takeaway

- Teacher Top-K OPD continues to improve with more training compute and consistently outperforms Student Top-K OPD, indicating that teacher-preferred supervision can effectively utilize a larger computation budget.

We investigate whether the benefit of teacher-preferred supervision persists as the training budget increases. Specifically, we extend training to 1,200 training steps for the 1.7B student models and report Avg@16 and Best@16 throughout training. We use Student Top-K OPD which is SOTA among baselines and compare it with Teacher Top-K OPD under the same training setup. As shown in Figure 6 (Right), Teacher Top-K consistently outperforms Student Top-K as scaling with training compute. Moreover, its performance continues to improve at later training stages rather than saturating quickly, indicating that teacher-preferred supervision can effectively use a larger computation budget.

8 Conclusion

In this work, we systematically investigate the limitations and design choices of OPD in cross-origin settings. We show that the distribution gap between a strong teacher and a smaller student can make standard OPD supervision unreliable: student-preferred tokens may lie in low-probability regions of the teacher distribution, resulting in uninformative optimization signals. Motivated by this observation, we investigate supervision based on teacher-preferred tokens and compare Online SFT, Teacher Top-K OPD, and Hybrid Teacher Top-K OPD. Our experiments demonstrate that Teacher Top-K OPD provides the most effective and stable supervision. We further analyze the key factors of Teacher Top-K OPD and show that direct RKL and its REINFORCE-style formulation are equivalent only under renormalization. Without renormalization, direct RKL suffers from a mass-shrinking issue that encourages the student to reduce probability mass on the teacher's Top-K support. Finally, our scaling experiments show that additional training compute continues to improve student performance, while stronger teachers provide a higher performance ceiling. Overall, our findings highlight the importance of aligning the supervision distribution with reliable teacher-preferred regions when distilling strong teachers into small, cross-origin student models, and provide practical insights for designing effective OPD algorithms.

[2]We evaluate the official Qwen3-14B ourselves under the same setting: 30.83% on AIME24, 24.58% on AIME25, and 66.64% on AMC23, averaging 40.68%, which is close to the reported performance.

AI USE STATEMENT

Large language model are used to assist with language editing, manuscript organization. The authors reviewed and revised all AI-assisted text for technical accuracy and consistency with the research. The authors take full responsibility for the methodology, analyses, experimental results, and conclusions in the final manuscript.

REPRODUCIBILITY STATEMENT

We provide the code for production and we also report the model, datasets, reward verification and evaluation procedures, rollout budgets, and training hyperparameters used in our experiments.

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 *International Conference on Learning Representations*, pp. 21246–21263, 2024.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

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

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, pp. 32694–32717, 2024.

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

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Huipeng Huang. Top-k reverse KL for distillation: What normalizing throws away. Blog post, 2026. URL https://huipenghuang.github.io/posts/mass-aware-topk-opd/.

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*, pp. 58791–58831, 2025.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a.

Zhaoyi Li, Deyang Kong, Yuan Wei, Evan Yang, Ranran Shen, Mahardika Krisna Ihsani, Ming Yang, Wei Zhang, Chuan Hao, Jian Yang, et al. Every coin has two sides: On the dual nature of generalization in on-policy distillation of large language models. *arXiv preprint arXiv:2608.16647*, 2026b.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, pp. 21558–21572, 2023.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. KL for a KL: On-Policy Distillation with Control Variate Baseline. *arXiv preprint arXiv:2605.07865*, 2026. URL https://arxiv.org/abs/2605.07865.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

THUNLP OPD. Opd. https://github.com/thunlp/OPD/issues/19, 2026.

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.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, pp. 15737–15749, 2020.

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*, 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*, 2026a.

Zhicheng Yang, Zhijiang Guo, Yifan Song, Minrui Xu, Yongxin Wang, Yiwei Wang, Xiaodan Liang, and Jing Tang. Prune-opd: Efficient and reliable on-policy distillation for long-horizon reasoning. *arXiv preprint arXiv:2605.07804*, 2026b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Zhi-Hua Zhou and Yuan Jiang. Nec4. 5: Neural ensemble based c4. 5. *IEEE Transactions on knowledge and data engineering*, pp. 770–773, 2004.

A Experimental Setup

Experiment Setting. We conduct experiments with three teacher–student pairs across different model scales. Specifically, we use Qwen3-1.7B-Base as the student with Qwen3-4B-Non-Thinking (Yang et al., 2025) as the teacher in the non-thinking setting, and a separately trained Qwen3-4B-Coding model as the teacher in the coding setting, following Hou et al. (2026). For the larger-scale setting, we consider Qwen3-1.7B-Base and Qwen3-8B-Base as students separately, both distilled from Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025) as the teacher.

To verify the effectiveness of Simple OPD for distillation between models of different scales, we conduct experiments on both mathematical reasoning and code generation tasks. Following the experimental settings of G-OPD (Yang et al., 2026a) and Li et al. (2026a), we use DAPO-Math-17K (Yu et al., 2025) as the training dataset for mathematical reasoning and Eurus-RL-Code (Cui et al., 2025), which contains 25K programming problems, as the training dataset for code generation.

Training Settings. All methods are implemented within the G-OPD (Yang et al., 2026a), which is developed based on the VeRL (Sheng et al., 2024) framework. During training, we set the sampling temperature to 1.0, top-p to 1.0, and the maximum response length to 15,360 tokens. For each training batch, we sample 64 prompts and generate 4 responses per prompt, resulting in 256 rollout trajectories per batch.

Evaluation. For mathematical reasoning, we evaluate on three competition-level benchmarks: AIME24, AIME25, and AMC23. For code generation, we evaluate on three widely-used code generation benchmarks: HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench+ (Jain et al., 2025). During evaluation, we set the temperature to 0.7, top-p to 1.0, and maximum generation length to 32,768 tokens. On each mathematical reasoning benchmark, we sample 32 solutions per problem and report the average accuracy as avg@32. For each code generation benchmark, we generate four solutions per problem and report the pass rate as avg@4, measured by the corresponding unit tests.

Baseline. We consider two baselines: (1) the sample OPD (Gu et al., 2024; Xiao et al., 2026; Lu & Thinking Machines Lab, 2025), where the student policy is optimized using tokens sampled from the student model itself, and (2) the Student Top-K OPD (Li et al., 2026a), where the loss is computed only over the student's top-K candidate tokens.

B More Analysis among Cross-origin OPD

We further analyze response length and the teacher's <EOS> token rank under same-origin and cross-origin OPD. As shown in Figure 7, cross-origin OPD produces substantially longer responses, while the student's <EOS> token is ranked much lower by the teacher. This suggests that preference misalignment extends to sequence termination, making it harder for the student to stop generation.

C REINFORCE-Style Implementation of Reverse-KL OPD

Reverse-KL OPD is commonly implemented with an advantage-weighted, REINFORCE-style loss (Li et al., 2026a; Yang et al., 2026a). This appendix derives the REINFORCE-style surrogate, specializes it to Top-K under renormalization, and analyzes the discrepancy induced by partial renormalization in open-source implementations.

For a student-generated prefix $(x, y_{<t})$, consider the next-token reverse KL

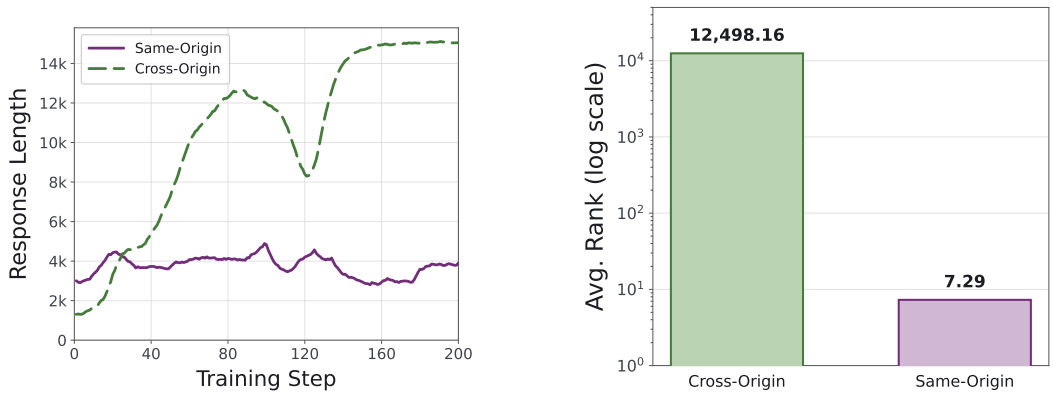
$$D_t^{\mathrm{RKL}} = D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_T(\cdot \mid x, y_{<t}))\,.$$

Let

$$A_t(v) \triangleq -\left(\log \pi_\theta(v \mid x, y_{<t}) - \log \pi_T(v \mid x, y_{<t})\right).$$

Applying the product rule and using $\nabla_\theta \pi_\theta(v) = \pi_\theta(v) \nabla_\theta \log \pi_\theta(v)$, the gradient expands into two terms:

$$\nabla_\theta D_t^{\mathrm{RKL}} = \sum_{v \in \mathcal{V}} \pi_\theta(v) \left(\log \pi_\theta(v) - \log \pi_T(v)\right) \nabla_\theta \log \pi_\theta(v)$$



(a) Response.

(b) <EOS> Rank.

Figure 7: (a) Cross-origin OPD leads to substantially longer responses than same-origin OPD. (b) The student's <EOS> token is ranked 12498.16 by the teacher in the cross-origin setting.

$$+ \sum_{v \in \mathcal{V}} \pi_\theta(v) \nabla_\theta \log \pi_\theta(v).$$

The second term vanishes because π_θ is normalized over the full vocabulary:

$$\begin{aligned} \sum_{v \in \mathcal{V}} \pi_\theta(v) \nabla_\theta \log \pi_\theta(v) &= \sum_{v \in \mathcal{V}} \nabla_\theta \pi_\theta(v) \\ &= \nabla_\theta \sum_{v \in \mathcal{V}} \pi_\theta(v) = \nabla_\theta 1 = 0. \end{aligned}$$

Therefore, using the definition of $A_t(v)$ above, the descent direction is

$$-\nabla_\theta D_t^{\mathrm{RKL}} = \sum_{v \in \mathcal{V}} \pi_\theta(v) A_t(v) \nabla_\theta \log \pi_\theta(v).$$

Accordingly, one may use the detached surrogate

$$\ell_t^{\text{RKL-style}} = - \sum_{v \in \mathcal{V}} \mathrm{sg}[\pi_\theta(v)] \, \mathrm{sg}[A_t(v)] \, \log \pi_\theta(v).$$

At the current policy and without PPO clipping or stale-policy corrections, this surrogate has the same gradient as the reverse-KL objective. We refer to it as a *REINFORCE-style implementation*.

C.1 Top-K as a renormalized special case

Top-K OPD restricts the action space to a teacher-selected candidate set. Let $S_t \triangleq \mathcal{S}_K(x, y_{<t})$ be the teacher Top-K support. Following the standard stop-gradient-through-TopK convention, S_t and the teacher quantities are treated as constants. Because the probabilities restricted to S_t do not sum to one, both policies must first be renormalized on the Top-K support:

$$Z_t^\theta \triangleq \sum_{u \in S_t} \pi_\theta(u \mid x, y_{<t}), \qquad Z_t^T \triangleq \sum_{u \in S_t} \pi_T(u \mid x, y_{<t}),$$

$$\hat{\pi}_\theta(v \mid x, y_{<t}) = \frac{\pi_\theta(v \mid x, y_{<t})}{Z_t^\theta}, \qquad \hat{\pi}_T(v \mid x, y_{<t}) = \frac{\pi_T(v \mid x, y_{<t})}{Z_t^T}, \quad v \in S_t.$$

The Top-K reverse-KL objective is then

$$D_t^{\mathrm{TopK}} = D_{\mathrm{KL}}(\hat{\pi}_\theta(\cdot \mid x, y_{<t}) \,\|\, \hat{\pi}_T(\cdot \mid x, y_{<t})) \,.$$

After renormalization, the same reverse-KL derivation as in the full-vocabulary case applies directly on S_t. Define the renormalized log-ratio signal

$$\hat{R}_t(v) \triangleq \log \hat{\pi}_T(v \mid x, y_{<t}) - \log \hat{\pi}_\theta(v \mid x, y_{<t}).$$

The gradient can be implemented by the detached surrogate

$$\ell_t^{\text{TopK-style}} = -\sum_{v\in S_t} \underbrace{\text{sg}[\hat{\pi}_\theta(v)]}_{\text{candidate weight}} \underbrace{\text{sg}[\hat{R}_t(v)]}_{\text{log-ratio signal}} \underbrace{\log \hat{\pi}_\theta(v)}_{\text{live policy score}}. \tag{1}$$

At the current policy, this surrogate matches the gradient of D_t^{TopK} *only when* the candidate weights, log-ratio signal, and live policy score are all defined consistently on the renormalized distributions. Otherwise, REINFORCE-style and direct reverse KL are not equivalent.

Gradient analysis. Fix $(x, y_{<t})$ and omit the position subscript. Let $S \triangleq \mathcal{S}_K(x, y_{<t})$, $s(v) \triangleq \log \pi_\theta(v)$, $\tau(v) \triangleq \log \pi_T(v)$, and write $\bar{s}(v) = \log \hat{\pi}_\theta(v) = s(v) - \log Z^\theta$, $\bar{\tau}(v) = \log \hat{\pi}_T(v) = \tau(v) - \log Z^T$. For $D = D_{\text{KL}}(\hat{\pi}_\theta \| \hat{\pi}_T)$, direct differentiation yields

$$\nabla_\theta D = \sum_{v\in S} \hat{\pi}_\theta(v)\big(\bar{s}(v) - \bar{\tau}(v)\big)\nabla_\theta \bar{s}(v),$$

because $\sum_{v\in S} \hat{\pi}_\theta(v)\nabla_\theta \bar{s}(v) = \nabla_\theta 1 = 0$. This justifies Eq. (1) under full renormalization. Moreover, writing $r(v) = s(v) - \tau(v)$ and $c = \log(Z^\theta/Z^T)$, we have $\bar{s}(v) - \bar{\tau}(v) = r(v) - c$. Replacing the renormalized log-ratio by the raw full-vocabulary counterpart therefore changes only a candidate-independent baseline and does not affect the gradient *when the live score remains renormalized*. The live score must still be renormalized consistently with the candidate weights.

Partial renormalization in open-source implementations. The open-source REINFORCE-style implementation of Li et al. (2026a) applies Top-K renormalization only to the first factor in Eq. (1), i.e., the detached candidate weight, while the log-ratio signal and the live policy score remain defined by the full-vocabulary policy. With detached Top-K weights, its update is

$$g_{\text{impl}} = \sum_{v\in S} \hat{\pi}_\theta(v) r(v) \nabla_\theta s(v).$$

The gradient of the renormalized Top-K reverse-KL objective is instead

$$\begin{aligned} g_{\text{TopK}} &= \sum_{v\in S} \hat{\pi}_\theta(v) r(v) \nabla_\theta \bar{s}(v) \\ &= g_{\text{impl}} - \mathbb{E}_{v\sim\hat{\pi}_\theta}[r(v)]\nabla_\theta \log Z^\theta. \end{aligned}$$

Because $\mathbb{E}_{v\sim\hat{\pi}_\theta}[r(v)] = D + \log(Z^\theta/Z^T)$, the discrepancy is

$$\boxed{g_{\text{TopK}} = g_{\text{impl}} - \left(D + \log\frac{Z^\theta}{Z^T}\right)\nabla_\theta \log Z^\theta.} \tag{2}$$

The extra term in g_{impl} is an update along the sensitivity of the total student mass on S; it is not part of the renormalized Top-K reverse-KL gradient, which depends only on the relative student distribution within S. When $S = \mathcal{V}$, one has $Z^\theta = Z^T = 1$ and $\nabla_\theta \log Z^\theta = 0$, so the gap vanishes and the usual full-vocabulary reverse-KL implementation is recovered. For $S \subsetneq \mathcal{V}$, however, the correction is generally nonzero.

This distinction is especially important for teacher Top-K: the teacher-selected support need not contain most of the student's probability mass, so Z^θ can be substantially below one and can vary during training. The correction in Eq. (2) is therefore much larger than in student Top-K, which selects the student's own highest-probability tokens and typically retains more mass. The student Top-K check reported in (Li et al., 2026a; THUNLP OPD, 2026) found little difference with or without renormalization. Our teacher Top-K experiments, however, show a substantial difference under the evaluated settings (Figure 4). Together, these observations suggest that the practical impact of the normalization convention should be assessed separately for student- and teacher-selected supports.

D Why Does Renormalization Matter for Direct Reverse KL?

We provide a more detailed explanation of why renormalization is important for direct reverse KL over a truncated Top-K support. As above, let S denote the teacher's Top-K support, and let $p_i = \pi_\theta(i)$

and $q_i = \pi_T(i)$ denote the student and teacher probabilities, respectively. The non-renormalized objective is

$$L_{\text{non-renorm}} = \sum_{i \in S} p_i \log \frac{p_i}{q_i}. \tag{3}$$

Importantly, the probabilities restricted to S do not generally sum to one. Consistent with Appendix C.1, define

$$Z^\theta = \sum_{i \in S} p_i, \qquad Z^T = \sum_{i \in S} q_i. \tag{4}$$

Thus, Eq. 3 is not the reverse KL between the student and teacher distributions conditioned on S. Instead, it jointly depends on both the *relative allocation* of probability within S and the *total student mass* Z^θ assigned to S.

To make this distinction explicit, define the renormalized distributions

$$\hat{\pi}_\theta(i) = \frac{p_i}{Z^\theta}, \qquad \hat{\pi}_T(i) = \frac{q_i}{Z^T}, \qquad i \in S, \tag{5}$$

The non-renormalized objective can then be decomposed as

$$\begin{aligned} L_{\text{non-renorm}} &= Z^\theta \sum_{i \in S} \hat{\pi}_\theta(i) \log \frac{Z^\theta \hat{\pi}_\theta(i)}{Z^T \hat{\pi}_T(i)} \\ &= Z^\theta \left[D_{\text{KL}}(\hat{\pi}_\theta \| \hat{\pi}_T) + \log \frac{Z^\theta}{Z^T} \right]. \end{aligned}$$

This decomposition reveals two distinct effects. The first term encourages the student to match the teacher's *relative probability allocation* within the Top-K support. The second term, however, makes the objective explicitly dependent on the total student mass Z^θ assigned to the support. Therefore, the optimization is not solely concerned with distributional alignment within S. Indeed, if the conditional student distribution $\hat{\pi}_\theta$ is held fixed and only Z^θ is optimized, then

$$\frac{\partial L_{\text{non-renorm}}}{\partial Z^\theta} = D_{\text{KL}}(\hat{\pi}_\theta \| \hat{\pi}_T) + \log \frac{Z^\theta}{Z^T} + 1. \tag{6}$$

The stationary point is

$$Z^{\theta *} = Z^T \exp(-1 - D_{\text{KL}}(\hat{\pi}_\theta \| \hat{\pi}_T)), \tag{7}$$

which is strictly smaller than Z^T. Thus, even with a fixed conditional distribution, the non-renormalized objective contains an explicit incentive to reduce the student's mass on S; the incentive becomes stronger when the conditional student–teacher mismatch is large.

In particular, consider uniformly scaling the student probabilities on S by a factor $\alpha \in (0, 1)$, while leaving their relative proportions unchanged:

$$p'_i = \alpha p_i, \qquad i \in S. \tag{8}$$

The corresponding objective becomes

$$L' = \alpha \sum_{i \in S} p_i \left(\log \frac{p_i}{q_i} + \log \alpha \right).$$

Hence, reducing the total student mass on S changes the objective even when the relative distribution $\hat{\pi}_\theta$ within S remains exactly the same. This introduces an undesirable optimization direction: the model can reduce the contribution of the Top-K terms by decreasing the total probability mass assigned to the teacher's support, rather than improving the relative alignment with the teacher.

This effect is particularly problematic when K is large, since the Top-K support then covers a substantial portion of the teacher distribution while the student may still assign considerable probability mass outside S. Without renormalization, the objective does not explicitly require the student to redistribute its probability mass toward the teacher's support.

By contrast, the renormalized objective first conditions both distributions on S:

$$L_{\text{renorm}} = \sum_{i \in S} \hat{\pi}_\theta(i) \log \frac{\hat{\pi}_\theta(i)}{\hat{\pi}_T(i)}.$$

This is a valid KL divergence, so $L_{\text{renorm}} \geq 0$, with equality if and only if $\hat{\pi}_\theta = \hat{\pi}_T$ on S. More importantly, the total mass Z^θ disappears from the objective: replacing p_i by αp_i for every $i \in S$ leaves $\hat{\pi}_\theta$ and L_{renorm} unchanged. The student therefore cannot lower the renormalized loss by shrinking all probabilities on the support. Renormalization removes this competing optimization direction and focuses the learning signal on the relative probability allocation among the teacher's Top-K tokens. It does not, however, explicitly force Z^θ to match Z^T.

Overall, this analysis explains why the two implementations, despite both being described as direct reverse KL over the teacher's Top-K support, can exhibit substantially different empirical behavior. Renormalization effectively changes the objective from an unnormalized partial divergence to a reverse KL between distributions conditioned on the Top-K support.

E More Results

E.1 Response Length

Takeaway

- A longer training response length works better: increasing the response length from 3K to 15K consistently improves performance across benchmarks.

We further evaluate the impact of the maximum response length used during training. Specifically, we vary the training response length from 3K to 15K, using 3K, 7K, 10K, 13K, and 15K as the maximum response lengths, while keeping the evaluation length fixed at 32K across all benchmarks. As shown in Figure 8, we observe a consistent performance improvement as the training response length increases. On AIME24, AIME25, and AMC23, longer training lengths generally lead to higher rewards, indicating that allowing longer responses during training enables the model to benefit from more extended reasoning trajectories. The improvement is particularly evident when increasing the training length from 3K to 10K, while the gains become smaller at longer lengths. Overall, these results suggest that our method benefits from longer training responses and remains effective across different response-length budgets.

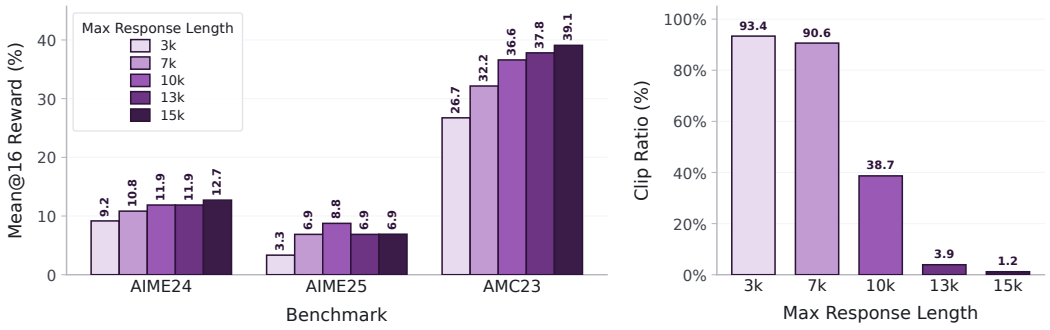


Figure 8: Scaling with response length

F Related Work

F.1 On-Policy Distillation

Knowledge distillation has long been used to transfer the behavior of a large teacher model into a smaller student model (Hinton et al., 2015), while ten years earlier a similar idea had already been published by Zhou & Jiang (2004). In sequence generation, early distillation methods train the student on teacher-generated sequences or token-level teacher distributions, leading to effective compression but still relying on fixed off-policy trajectories (Kim & Rush, 2016). This creates a train–test mismatch similar to exposure bias: during training, the student observes teacher or gold prefixes, whereas during inference it must condition on its own generated tokens (Bengio et al.,

2015). Reasoning distillation further extends this paradigm by using teacher-generated rationales or chain-of-thought traces as additional supervision for smaller models (Yang et al., 2025). However, these methods still mainly supervise the student on teacher-induced trajectories and fail to fully address the compounding errors arising from the distribution mismatch between teacher and student trajectories (Ross et al., 2011). On-policy distillation addresses this limitation by letting the student sample its own rollouts and querying the teacher on these student-generated states. GKD trains autoregressive students from self-generated outputs with teacher feedback (Agarwal et al., 2024), while MiniLLM studies reverse-KL distillation for generative language models (Gu et al., 2024). DistiLLM further improves LLM distillation efficiency with skew KL and adaptive off-policy training (Ko et al., 2024). More recently, OPD has become a practical post-training recipe for reasoning models, as also reflected in large-scale systems and surveys (Yang et al., 2025; Song & Zheng, 2026; Xiao et al., 2026).

F.2 Cross-origin Distillation and Distributional Shift.

Most existing distillation methods implicitly assume that the teacher and student policies are sufficiently aligned, such as sharing similar model families, tokenizers. This assumption becomes fragile in cross-origin distillation, where the student may generate trajectories that deviate from the teacher's preferred reasoning path or fall outside the teacher's high-probability support. Recent studies have increasingly identified such teacher–student distribution mismatch as a key bottleneck of OPD: when student-generated prefixes enter regions where the teacher is unreliable, token-level supervision can become weak, noisy, or even misleading (Li et al., 2026a; Fu et al., 2026; Xie et al., 2026). To mitigate this issue, recent methods have explored different ways to restore reliable supervision, including pruning or truncating drifted rollout segments in long-horizon reasoning (Yang et al., 2026b), using speculative or interleaved sampling to keep training trajectories closer to teacher-compatible regions (Xu et al., 2025), and constraining distillation within trust regions or constructing proximal teachers to explicitly reduce the teacher–student distribution gap (Xing et al., 2026). These works suggest that directly distilling a stronger teacher is insufficient for cross-origin OPD. In our work, we identify obtaining a reliable teacher signal under severe distribution mismatch as a key challenge and systematically analyze different distribution alignment strategies to address it.

G Failure Case of Student Top-K OPD

System

Janet rolls a standard 6-sided die 4 times and keeps a running total of the numbers she rolls. What is the probability that at some point, her running total will equal 3? The final answer can be written in the form $\frac{m}{n}$, where m and n are relatively prime positive integers. What is $m+n$? Please reason step by step, and put your final answer within $\square$.
assistant
<think>
</think>

User

GetSize
We are given a transformation T_k that consists of two steps:
1. **Rotate the plane k degrees counterclockwise around the origin.** 2. **Reflect the plane across the y-axis.**
We are to find the smallest positive integer n such that the sequence of transformations $T_1, T_2, T_3, \ldots, T_n$ maps the point $(1, 0)$ back to itself.
Let's analyze the transformation step by step.
—
...
inter
1. We are given a transformation T_k that first rotates the plane k degrees counterclockwise around the origin and then reflects the plane across the y-axis. 2. We are to find the smallest positive integer n such that the sequence of transformations $T_1, T_2, \ldots, T_n$ maps the point $(1, 0)$ back to itself. 3. We analyze the action of T_k on the point $(1, 0)$: - Rotate by k degrees counterclockwise: $\begin{pmatrix} \cos k & -\sin k \\ \sin k & \cos k \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos k \\ \sin k \end{pmatrix}$ - Reflect across the y-axis: $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos k \\ \sin k \end{pmatrix} = \begin{pmatrix} -\cos k \\ \sin k \end{pmatrix}$ 4. So, $T_k \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} -\cos k \\ \sin k \end{pmatrix}$ 5. We are to find the smallest n such that the composition of n transformations of the form T_k returns $(1, 0)$ to itself. 6. This is equivalent to finding the smallest n such that the composition of n transformations is the identity transformation. 7. The group of transformations T_k is isomorphic to the dihedral group D_2, which has 4 elements (identity, rotation by $180°$, reflection across y-axis, and their composition). 8. Therefore, the smallest such n is $\boxed{4}$.

$$\boxed{4}$$

is the smallest positive integer such that the sequence of transformations $T_1, T_2, \ldots, T_n$ returns the point $(1, 0)$ to itself.[3]
Thus, the final answer is:

$$\boxed{4}$$

$$\boxed{4}$$

$$\boxed{4}$$

$$\boxed{4}$$

$$\boxed{4}$$

$$\boxed{4}$$

(repeats until the maximum generation length)