TAIL-AWARE TOP-k ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as an effective paradigm for transferring knowledge between language models, where a student is trained to align its next-token distribution with the teacher's along its own trajectories. To provide dense supervision at a tractable cost, many works minimize the reverse Kullback–Leibler (KL) divergence between the student's and the teacher's normalized distributions over the teacher's top-k tokens. However, this normalized objective discards the information about tail probability: the total probability outside the teacher's top-k tokens. As a result, the optimization can steadily increase the student's tail probability and entropy, empirically degrading downstream accuracy. To address this issue, we propose Tail-Aware Top-k OPD (**TA-OPD**), a novel distillation method that restores the missing tail probability signal. In particular, TA-OPD minimizes the reverse KL divergence over the top-k tokens plus a tail token that carries the tail probability. In effect, TA-OPD faithfully aligns the student's next-token distribution with the teacher's, preventing the tail probability and entropy increase caused by top-k normalization. Extensive experiments demonstrate the superiority of TA-OPD, improving Avg@8 by up to 8.05 points on common benchmarks.

1 INTRODUCTION

Knowledge distillation (Hinton et al., 2015) is a promising approach for transferring the capabilities of large language models (LLMs) to smaller models. However, standard distillation is performed off-policy: the student is trained on teacher-generated sequences (Kim & Rush, 2016; Gu et al., 2024), yet at inference time it must condition on its own generations. This discrepancy between training and inference yields errors that accumulate quickly along the generated sequence (Bengio et al., 2015; Arora et al., 2022) and hinders effective learning (Xu et al., 2025). This limitation motivates on-policy distillation (OPD) (Agarwal et al., 2024; Lu & Lab, 2025), where the student is trained to align its next-token distribution with the teacher's along its own trajectories.

A popular idea, normalized top-k OPD (Fu et al., 2026; Xing et al., 2026), minimizes the reverse Kullback–Leibler (KL) divergence between the student and the teacher's normalized distributions over the teacher's top-k tokens. However, it discards the information about tail probability: the total probability mass outside the teacher's top-k tokens. Theoretically, the optimization increases the student's tail probability whenever the student's distribution is misaligned with the teacher's, a common occurrence in distillation. The increase in students' tail probabilities empirically leads to tokens being sampled more frequently outside the teacher's top-k tokens (see Figure 1), drifting toward prefixes where the teacher's supervision is unreliable. Consequently, the student fails to imitate the teacher's next-token distribution, resulting in degraded downstream performance.

To address this, we propose Tail-Aware Top-k OPD (**TA-OPD**), a novel distillation method that restores the missing tail probability signal. In particular, TA-OPD minimizes the reverse KL divergence over the top-k tokens plus a tail token that carries the tail probability. In effect, TA-OPD redistributes the student's tail probability to the teacher's top-k tokens, preventing the tail probability and entropy increase that top-k normalization would otherwise cause. Theoretically, we show that the TA-OPD objective is a tight lower bound of the full-vocabulary reverse KL divergence, with a gap that scales with the student's tail probability. To close the gap, when the sampled token's probability is available, we derive a sampled variant of TA-OPD that estimates this gap with the sampled token, yielding an unbiased and low-variance estimate of the full-vocabulary reverse KL divergence.

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.

We extensively validate TA-OPD across multiple mathematical benchmarks (MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AMC (Li et al., 2024) and AIME24/25 (Li et al., 2024)) on three model families (Qwen2.5 (Qwen et al., 2025), Qwen3 (Yang et al., 2025), and Llama-3.1 (Grattafiori et al., 2024)). Notably, when distilling OpenThinker3-7B (Guha et al., 2025) into Qwen2.5-7B-Instruct, normalized top-k OPD's training collapses: its training entropy increases to around 6, and its Avg@8 accuracy on MATH500 drops to 68.78%. In contrast, TA-OPD keeps the training entropy below 1.5 and improves Avg@8 accuracy on MATH500 to 77.88%. Through additional analyses, we show that TA-OPD is particularly effective when the student–teacher capability gap is large, and a small k is sufficient for TA-OPD to achieve excellent performance. Our code is available in this anonymous repository.

We summarize our contributions as follows:

- We show that normalized top-k OPD discards the tail of the probability distribution. As a result, minimizing the normalized objective can steadily increase the student's tail probability and entropy, empirically degrading downstream accuracy.
- We propose TA-OPD, a novel distillation method that restores the tail probability signal. We show that TA-OPD addresses the increase in tail probability and entropy caused by top-k normalization. Given the sampled token's probability, we further derive an unbiased variant for the full-vocabulary reverse KL divergence.
- We empirically show that TA-OPD achieves superior performance compared with normalized top-k OPD, improving Avg@8 accuracy by up to 8.05 points on common benchmarks.

2 PRELIMINARIES

On-policy distillation (OPD) aims to distill knowledge from a teacher LLM π_{te} to a student LLM π_θ on trajectories sampled from the student. Given a prompt $x \sim \mathcal{D}_x$, the student samples a response $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim \pi_\theta(\cdot \mid x)$. At each step t, on the prefix $\hat{y}_{<t} = (\hat{y}_1, \ldots, \hat{y}_{t-1})$, we define the student's and teacher's next-token distributions over a vocabulary $\mathcal{V}$ as $p_t = \pi_\theta(\cdot \mid x, \hat{y}_{<t})$ and $q_t = \pi_{\text{te}}(\cdot \mid x, \hat{y}_{<t})$. OPD minimizes the reverse KL divergence over student-sampled trajectories:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x,\, \hat{y}\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\text{KL}}(p_t \| q_t) \right], \tag{1}$$

where $D_{\text{KL}}(p_t\|q_t) = \sum_{v\in\mathcal{V}} p_t(v) \log \frac{p_t(v)}{q_t(v)}$. Different OPD methods vary in how they approximate $D_{\text{KL}}(p_t\|q_t)$ with their per-token losses ℓ_t. Given a student-sampled token $\hat{y}_t \sim p_t$, sampled-token OPD (Lu & Lab, 2025; Xiao et al., 2026) computes $\ell_t^{\text{sample}} = \log p_t(\hat{y}_t) - \log q_t(\hat{y}_t)$, which is cheap to compute but discards the dense information over the remaining vocabulary. Full-vocabulary OPD (Xu et al., 2026) computes $\ell_t^{\text{full}} = D_{\text{KL}}(p_t\|q_t)$ exactly over $\mathcal{V}$. It provides dense supervision but is prohibitively expensive for LLMs.

Normalized top-k OPD provides an intermediate design between sampled-token and full-vocabulary OPD by restricting the divergence computation to a subset. Prior work adopts two choices for the subset: the student's top-k tokens (Li et al., 2026) or the teacher's top-k tokens (Jin et al., 2026; Fu et al., 2026). In this work, we primarily focus on the teacher top-k variant. Formally, we define the teacher's top-k tokens as $S_t^k = \text{TopK}(q_t, k)$. The normalized distributions on S_t^k are given by:

$$\bar{p}_t^{(S_t^k)}(v) = \frac{p_t(v)\,\mathbf{1}[v \in S_t^k]}{\sum_{u\in S_t^k} p_t(u)}, \qquad \bar{q}_t^{(S_t^k)}(v) = \frac{q_t(v)\,\mathbf{1}[v \in S_t^k]}{\sum_{u\in S_t^k} q_t(u)},$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Distillation is then performed by minimizing the subset reverse KL divergence $\ell_t^{\text{norm}} = D_{\text{KL}}\big(\bar{p}_t^{(S_t^k)} \,\|\, \bar{q}_t^{(S_t^k)}\big)$, yielding the trajectory-level objective:

$$\mathcal{L}_{\text{OPD}}^{\text{norm}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x,\, \hat{y}\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\text{KL}}\big(\bar{p}_t^{(S_t^k)} \,\|\, \bar{q}_t^{(S_t^k)}\big) \right]. \tag{2}$$

Normalized top-k OPD reduces the teacher-query cost while retaining dense, multi-token supervision over S_t^k, making it a practical approximation to full-vocabulary OPD. However, the objective

discards the tail probability, i.e., the total probability mass outside the teacher's top-k tokens:

$$p_t^{\text{tail}} = \sum_{v \notin S_t^k} p_t(v), \qquad q_t^{\text{tail}} = \sum_{v \notin S_t^k} q_t(v). \tag{3}$$

Concerningly, the objective can be minimized even when the student's tail probability is much higher than the teacher's. The student's next-token distribution can therefore diverge substantially from the teacher's. We proceed by analyzing how the method affects the student's tail probability.

3 Motivation

In this section, we investigate how the normalized objective in Eq. (2) changes the student's tail probability. We find that minimizing the objective can steadily increase the student's tail probability.

3.1 Theoretical Analysis of the Tail Probability

Fix a step t and let $z_{t,v}$ denote the student logit for token v under the prefix, so that $p_t(v) = \frac{e^{z_{t,v}}}{\sum_{u \in \mathcal{V}} e^{z_{t,u}}}$. We present the following propositions on the optimization of the normalized objective.

Proposition 1. *For every step t, the gradient of ℓ_t^{norm} with respect to the student logit $z_{t,v}$ is*

$$\frac{\partial \ell_t^{\text{norm}}}{\partial z_{t,v}} = \begin{cases} \bar{p}_t^{(S_t^k)}(v) \left[\log \dfrac{\bar{p}_t^{(S_t^k)}(v)}{\bar{q}_t^{(S_t^k)}(v)} - \ell_t^{\text{norm}} \right], & v \in S_t^k, \\ 0, & v \notin S_t^k. \end{cases}$$

Moreover, the gradients on the logits of the top-k tokens sum to zero: $\sum_{v \in S_t^k} \frac{\partial \ell_t^{\text{norm}}}{\partial z_{t,v}} = 0$.

The proof is in Appendix B.1. The proposition shows that the objective provides no explicit mechanism for decreasing the student's tail probability. Decreasing the tail probability requires raising the top-k logits relative to the tail logits. However, the normalized objective can do neither: the tail logits receive zero gradient, and the gradients on the top-k logits sum to zero. We further explore how the zero-sum gradient within the top-k tokens affects the student's tail probability.

Proposition 2. *Assume that the student π_θ is a tabular softmax policy, where each token v at each step t is associated with an independent logit parameter $z_{t,v} = \theta_{t,v}$. Write $\bar{p} := \bar{p}_t^{(S_t^k)}$ and $\bar{q} := \bar{q}_t^{(S_t^k)}$ for the normalized distributions on S_t^k, and let $\mathrm{Cov}_{v \sim r}(f, g)$ denote the covariance of $f(v)$ and $g(v)$ under $v \sim r$. After one gradient descent step $\theta \leftarrow \theta - \eta \frac{\partial \ell_t^{\text{norm}}}{\partial \theta}$ with learning rate η, the student's tail probability change Δp_t^{tail} is given by:*

$$\Delta p_t^{\text{tail}} = \eta\, p_t^{\text{tail}} \big(1 - p_t^{\text{tail}}\big) \Big[\underbrace{\operatorname*{Cov}_{v \sim \bar{p}} \big(\bar{p}(v), \log \bar{p}(v)\big)}_{\text{student self-covariance}} - \underbrace{\operatorname*{Cov}_{v \sim \bar{p}} \big(\bar{p}(v), \log \bar{q}(v)\big)}_{\text{student–teacher covariance}} \Big] + O(\eta^2).$$

Consequently, the student's tail probability increases to first order if and only if the student self-covariance exceeds the student–teacher covariance.

The proof is provided in Appendix B.2. The proposition[1] shows that minimizing the normalized objective strictly increases the student's tail probability if the student self-covariance exceeds the student–teacher covariance. The student self-covariance is non-negative, since $\log \bar{p}$ is monotonically increasing with $\bar{p}$. In contrast, the student–teacher covariance is positive only when the student concentrates its probability on the teacher's high-probability tokens within S_t^k, and can be negative when the two distributions are poorly aligned.

In practice, when the student–teacher capability gap is large, the student cannot imitate the teacher's next-token distribution well, and its probability mass cannot concentrate on the teacher's high-probability tokens. As a result, the student–teacher covariance tends to be smaller than the student self-covariance, making the normalized objective exhibit a systematic bias toward increasing the student's tail probability. This analysis motivates us to empirically examine whether the tail probability increases and its influence on the training dynamics of OPD.

[1] We use the proposition as a motivation rather than a formal theoretical guarantee. In Appendix F.1, we empirically validate the connection between the covariance metric and the change in tail probability.

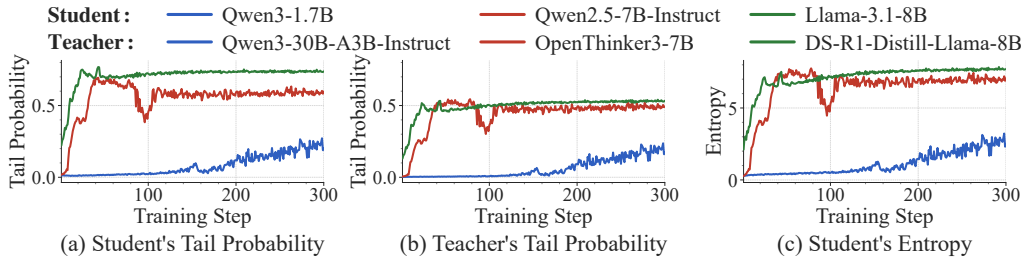


Figure 1: **The student's tail probability, the teacher's tail probability, and the student's token-level entropy across training steps.** Normalized top-k OPD consistently increases the student's tail probability and entropy across all three model pairs.

3.2 Empirical Study of the Tail Probability Increase

Setup. We conduct experiments with three student–teacher model pairs: Qwen3-1.7B (Yang et al., 2025) with Qwen3-30B-A3B-Instruct-2507, Qwen2.5-7B-Instruct with OpenThinker3-7B, and Llama-3.1-8B (Grattafiori et al., 2024) with DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025). All models are trained on DAPO-MATH-17K (Yu et al., 2026) with $k = 16$.

Normalized top-k OPD steadily increases the student's tail probability. Figure 1 visualizes the students' and teachers' tail probabilities on student-generated prefixes, together with the students' entropy across training steps. The results show that across all three model pairs, the student's tail probability and entropy steadily increase during on-policy training, validating our theoretical analysis. Furthermore, as the student's tail probability increases, it more frequently samples tokens outside the teacher's top-k tokens. This drives the student toward prefixes where the teacher is uncertain, as reflected by the increase in the teacher's tail probability. Prior work shows that the teacher's supervision is unreliable for such uncertain prefixes (Fu et al., 2026; Xie et al., 2026), as the teacher itself exhibits significantly reduced accuracy on these prefixes. Consequently, normalized top-k OPD progressively shifts probability mass away from the teacher's top-k tokens, making the student fail to imitate the teacher's next-token distribution.

Tail probability increase hurts downstream performance. Figure 2 plots validation accuracy against the student's tail probability. Each point is a checkpoint from a normalized top-k OPD run with Qwen2.5-7B-Instruct as the student and OpenThinker3-7B as the teacher. The validation set is MATH500, and the validation sampling temperature is 1. The validation accuracy degrades significantly from above 0.7 to below 0.2 as the tail probability grows from near 0 to around 0.7, with a Pearson correlation coefficient of -0.91.

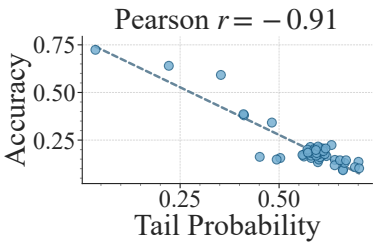


Figure 2: **Validation accuracy versus student's tail probability.**

In Appendix C, we present a detailed analysis of the tail probability increase phenomenon. We find that it occurs when there is a large capability gap between the student and teacher and when k is not sufficiently large (e.g., $k \leq 64$). Moreover, the phenomenon intensifies as the maximum training response length increases.

4 Method

In our previous analysis, we showed that the normalized objective discards information about the tail probability, leading to the tail probability increase issue (Figure 3 (b)). To address this problem, our key idea is to restore the missing tail probability signal.

Tail-Aware Top-k OPD. We propose Tail-Aware Top-k OPD (TA-OPD), which explicitly aligns the student's tail probability with that of the teacher (Figure 3 (c)). In particular, we introduce a tail token v_{tail} that carries the tail probability. We then minimize the reverse KL divergence over the

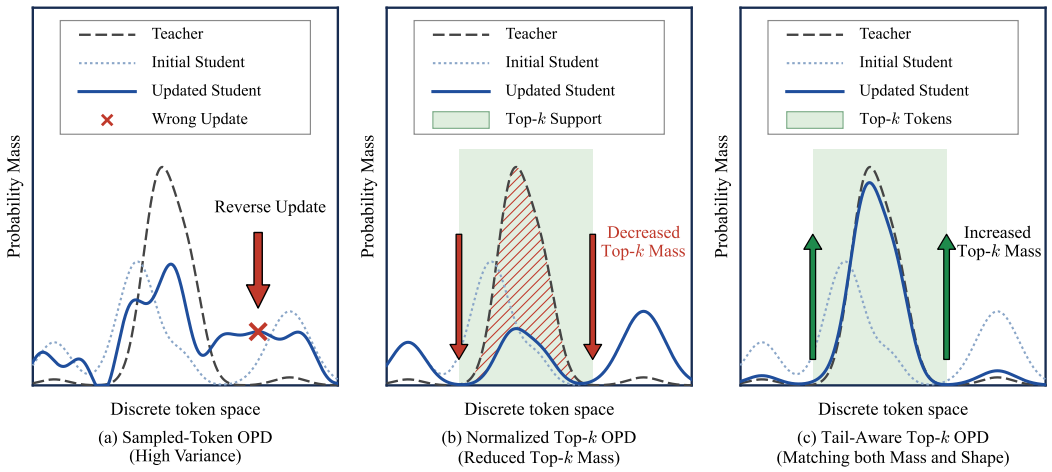


Figure 3: **Geometric Interpretation of OPD Objectives. (a) Sparse Update (Sampled-Token OPD):** The supervision is derived from a single sampled token, making the update direction stochastic and prone to updating some token logits in erroneous directions. **(b) Increased Tail Probability (Normalized Top-k OPD):** The normalized objective discards the tail probability, leading to increased tail probability. **(c) Decreased Tail Probability (Tail-Aware Top-k OPD, Ours):** TA-OPD restores the missing tail probability signal, matching the student's tail probability to the teacher's.

top-k tokens plus the tail token. Formally, we define the augmented token set $S_t^+ = S_t^k \cup \{v_{\text{tail}}\}$. On S_t^+, each top-k token retains its original probability, $p_t(v) = \pi_\theta(v \mid x, \hat{y}_{<t})$, while the tail token carries the tail probability, $p_t(v_{\text{tail}}) = p_t^{\text{tail}}$. The per-token loss function of TA-OPD is given by:

$$\ell_t^{\text{TA}} = \sum_{v \in S_t^+} p_t(v) \log \frac{p_t(v)}{q_t(v)} = \sum_{v \in S_t^k} p_t(v) \log \frac{p_t(v)}{q_t(v)} + p_t^{\text{tail}} \log \frac{p_t^{\text{tail}}}{q_t^{\text{tail}}}. \tag{4}$$

By introducing the tail token v_{tail}, the term $p_t^{\text{tail}} \log \frac{p_t^{\text{tail}}}{q_t^{\text{tail}}}$ explicitly compares the student's and teacher's tail probabilities, thereby driving the student's tail probability toward the teacher's. We next characterize the gradient signal of ℓ_t^{TA}.

Proposition 3. *For every step t, the gradient of ℓ_t^{TA} with respect to the student logit $z_{t,v}$ is*

$$\frac{\partial \ell_t^{\text{TA}}}{\partial z_{t,v}} = \begin{cases} p_t(v)\Big(\log \dfrac{p_t(v)}{q_t(v)} - \ell_t^{\text{TA}}\Big), & v \in S_t^k, \\ p_t(v)\Big(\log \dfrac{p_t^{\text{tail}}}{q_t^{\text{tail}}} - \ell_t^{\text{TA}}\Big), & v \notin S_t^k. \end{cases} \tag{5}$$

The proposition reveals two key properties of TA-OPD. First, while the normalized objective gives no gradient to logits outside the top-k tokens, TA-OPD updates them through the tail log-ratio $\log \frac{p_t^{\text{tail}}}{q_t^{\text{tail}}}$, which compares the student's tail probability with the teacher's. Second, the gradients on the top-k logits no longer sum to zero. The top-k gradient sum $\sum_{v \in S_t^k} \frac{\partial \ell_t^{\text{TA}}}{\partial z_{t,v}} = p_t^{\text{tail}} \Big(\ell_t^{\text{TA}} - \log \frac{p_t^{\text{tail}}}{q_t^{\text{tail}}}\Big)$ is negative whenever the tail log-ratio is larger than the average log-ratio over S_t^k. It can therefore explicitly raise the top-k logits relative to the tail whenever the student overweights the tail relative to the teacher. In the following, we show that ℓ_t^{TA} is not merely a heuristic modification but a principled approximation of the full-vocabulary objective.

Proposition 4 (Lower bound of the full-vocabulary reverse KL)**.** *Let $\tilde{p}_t(v) = p_t(v)/p_t^{\text{tail}}$ and $\tilde{q}_t(v) = q_t(v)/q_t^{\text{tail}}$ for $v \notin S_t^k$ be the normalized tail distributions. Then*

$$\ell_t^{\text{TA}} = \ell_t^{\text{full}} - p_t^{\text{tail}} \, D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t) \le \ell_t^{\text{full}}. \tag{6}$$

The proofs of the above propositions are presented in Appendix B.3 and B.4. This proposition shows that the loss functions of TA-OPD and full-vocabulary OPD differ only by a non-negative residual

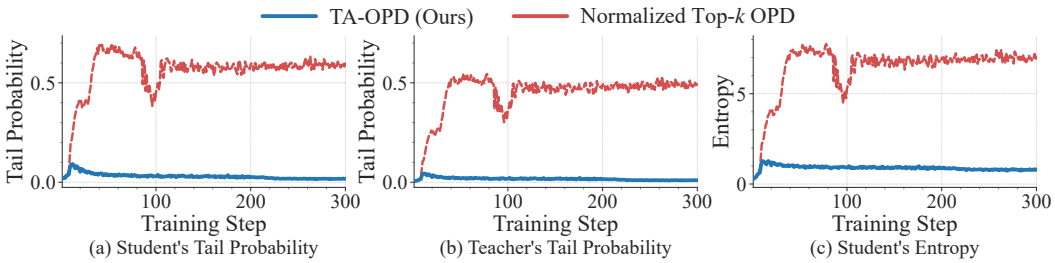


Figure 4: **Tail probabilities and token-level entropy of TA-OPD and normalized top-k OPD over training.** The student and teacher models are Qwen2.5-7B-Instruct and OpenThinker3-7B.

term $p_t^{\text{tail}}\, D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t)$. In practice, TA-OPD drives both the student's and teacher's tail probabilities close to zero (see Figure 4 (a) and (b)). As a result, the residual term becomes small, making ℓ_t^{TA} a tight lower bound of the ideal full-vocabulary reverse KL objective. We empirically validate the tightness in Appendix F.2.

5 Experiments

5.1 Experimental Setup

Models and Training Dataset. For the main experiments, we use the three student–teacher model pairs as in Section 3.2: Qwen3-1.7B with Qwen3-30B-A3B-Instruct-2507, Qwen2.5-7B-Instruct with OpenThinker3-7B, and Llama-3.1-8B with DeepSeek-R1-Distill-Llama-8B. We disable thinking modes for Qwen3 models. The training set is DAPO-MATH-17K.

Evaluation. We evaluate on six math reasoning benchmarks—MATH500, Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AMC (Li et al., 2024) and AIME24/25 (Li et al., 2024)—and two out-of-distribution benchmarks, ARC-c (Clark et al., 2018) and MMLU-Pro (Wang et al., 2024). We use a rollout temperature of 0.7, top-p sampling with $p = 0.95$, and a maximum response length of 8192 tokens. For MMLU-Pro, we report Pass@1. For other datasets, we sample 8 responses per question and report the average accuracy (Avg@8).

Compared Methods. We compare our method with sampled-token OPD, normalized top-k OPD, and MOPD top-k distillation (Ma et al., 2026). In Appendix F.3, we conduct an additional comparison with unnormalized top-k OPD. We provide an introduction to these methods in Appendix D.

Implementation. All OPD experiments use the same training setup: 300 training steps with a learning rate of 1×10^{-6}. Unless otherwise specified, we set $k = 16$ for top-k OPD. We use a prompt batch size of 72 and sample 4 rollouts per prompt, with a maximum generation length of 7168 tokens. More details of implementation are provided in Appendix E.

5.2 Results

TA-OPD achieves the best performance across student–teacher model pairs. Table 1 compares TA-OPD with sampled-token OPD, normalized top-k OPD, and MOPD top-k distillation. On mathematical benchmarks, TA-OPD attains the best average accuracy on every student–teacher pair, outperforming normalized top-k OPD by **+5.26** points on Qwen2.5-7B-Instruct and **+8.05** points on Llama-3.1-8B. Notably, TA-OPD demonstrates a significantly greater advantage on MATH500 with Llama-3.1-8B over normalized top-k OPD, improving Avg@8 from 31.30 to 50.83 (**+19.53** points). Regarding out-of-distribution performance, TA-OPD improves ARC-c accuracy over normalized top-k OPD from 23.07 to 76.30 with Qwen2.5-7B-Instruct and from 10.01 to 65.70 with Llama-3.1-8B, indicating that the tail probability increase harms not only mathematical reasoning but also the student's general capabilities. Overall, these results demonstrate that TA-OPD consistently achieves the best in-distribution and out-of-distribution performance across different model pairs.

Table 1: **Results on math reasoning and out-of-distribution (OOD) benchmarks.** We compare TA-OPD with sampled-token OPD (Sampled-token), normalized top-k OPD (Norm. top-k), and MOPD top-k distillation. Best results are shown in **bold**.

Methods	In-Distribution Performance							OOD Performance		
	MATH500	Minerva	Olympiad	AMC	AIME 24	AIME 25	Avg.	ARC-c	MMLU-Pro	Avg.
Student: ***Qwen2.5-7B-Instruct*** Teacher: ***OpenThinker3-7B***										
Sampled-token	74.60	31.25	41.22	**45.48**	14.58	16.25	37.23	74.99	47.66	61.33
Norm. top-k	68.78	25.55	36.02	41.42	13.75	15.00	33.42	23.07	37.38	30.23
MOPD top-k	76.63	32.49	41.07	43.18	15.42	15.42	37.37	57.91	**52.63**	55.27
TA-OPD (Ours)	**77.88**	**32.58**	**42.41**	45.03	**16.67**	**17.50**	**38.68**	**76.30**	50.35	**63.33**
Student: ***Llama-3.1-8B*** Teacher: ***DeepSeek-R1-Distill-Llama-8B***										
Sampled-token	43.45	12.78	16.56	17.62	2.08	1.25	15.62	41.88	28.99	35.44
Norm. top-k	31.30	9.24	9.96	11.45	1.25	0.42	10.60	10.01	18.33	14.17
MOPD top-k	24.23	8.23	8.22	9.04	0.83	0.83	8.56	21.42	19.18	20.30
TA-OPD (Ours)	**50.83**	**14.94**	**20.65**	**20.48**	**2.50**	**2.50**	**18.65**	**65.70**	**34.57**	**50.14**
Student: ***Qwen3-1.7B*** Teacher: ***Qwen3-30B-A3B-Instruct-2507***										
Sampled-token	82.98	49.49	**34.93**	52.41	**26.25**	19.58	44.27	**83.91**	50.08	67.00
Norm. top-k	81.80	47.20	33.28	50.60	23.33	17.08	42.22	82.48	49.66	66.07
MOPD top-k	81.25	47.66	34.46	51.51	24.58	**22.50**	43.66	83.61	50.37	66.99
TA-OPD (Ours)	**83.48**	**49.82**	34.50	**52.86**	**26.25**	20.83	**44.62**	83.59	**50.42**	**67.01**

TA-OPD prevents the increase in tail probability and entropy. In Figure 4, we compare the training dynamics of TA-OPD and normalized top-k OPD on the Qwen2.5-7B-Instruct and OpenThinker3-7B model pair. With normalized top-k OPD, the student's tail probability, the teacher's tail probability, and the student's entropy increase to approximately 0.6, 0.5, and 6 over the course of training. In contrast, TA-OPD keeps the two tail probabilities close to 0 and the student's entropy below 1.5. The same trends are consistently observed for three additional model pairs in Figure 19. These results show that TA-OPD addresses the tail-probability and entropy increases caused by top-k normalization.

A small k is sufficient for TA-OPD. In Figure 5, we ablate how the number of top-k tokens affects the performance of TA-OPD. In particular, we fix Qwen3-1.7B-Base as the student and Qwen3-8B as the teacher, and vary k over $\{1, 2, 4, 8, 16, 32\}$. For each k, we report the macro-averaged accuracy across the six math reasoning benchmarks. The results show that the accuracy is insensitive to k once $k \geq 2$: $k = 2$ attains an average accuracy of 30.43, only 0.43 points below the best result of 30.86 at $k = 32$. Overall, TA-OPD can be applied with a small k.

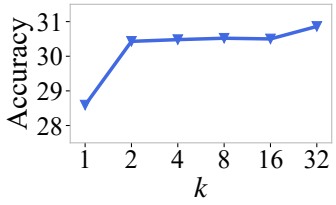


Figure 5: **Ablation study on k.**

Additional results. Due to space constraints, we defer additional analyses to the appendix. Appendix F.4 conducts a synthetic experiment to provide an intuitive understanding of TA-OPD and the optimization of normalized top-k OPD. Appendix F.5 shows that TA-OPD consistently achieves higher student–teacher top-k overlap ratios compared to normalized top-k OPD. Appendix F.6 shows that TA-OPD incurs negligible computational overhead over normalized top-k OPD.

6 Discussion

Sample-corrected TA-OPD. While TA-OPD's loss function is a tight lower bound of the full-vocabulary reverse KL divergence, it remains a biased estimate of the latter. By Proposition 4, the bias is $p_t^{\text{tail}} D_{\text{KL}}(\tilde{p}_t \,\|\, \tilde{q}_t)$. When the sampled token's probability is additionally available, we

Table 2: **Comparison between TA-OPD and sample-corrected TA-OPD on math reasoning and out-of-distribution (OOD) benchmarks.** Best results are shown in **bold**.

Methods	In-Distribution Performance							OOD Performance		
	MATH500	Minerva	Olympiad	AMC	AIME 24	AIME 25	Avg.	ARC-c	MMLU-Pro	Avg.
Student: ***DeepSeek-R1-Distill-Qwen-1.5B*** Teacher: ***JustRL-DeepSeek-1.5B***										
TA-OPD	86.13	30.97	57.04	76.20	41.67	29.73	53.62	35.40	**25.98**	**30.69**
SC-TA-OPD	**87.00**	**32.08**	**57.65**	**76.51**	**42.92**	**30.17**	**54.39**	**35.55**	25.76	30.66
Student: ***Qwen3-1.7B*** Teacher: ***Qwen3-30B-A3B-Instruct-2507***										
TA-OPD	83.48	**49.82**	34.50	52.86	26.25	**20.83**	44.62	83.59	**50.42**	67.01
SC-TA-OPD	**84.00**	49.54	**34.93**	**53.16**	**26.67**	19.58	**44.65**	**83.91**	50.28	**67.10**

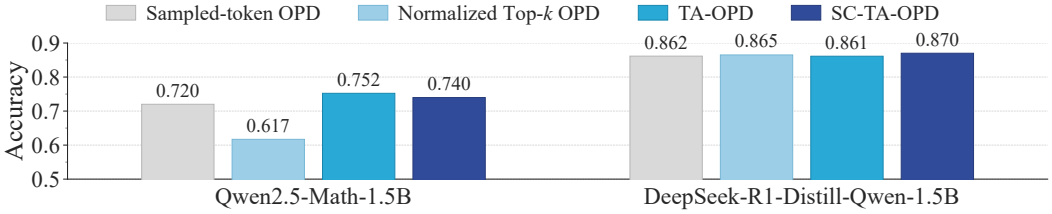


Figure 6: **Ablation study on student–teacher capability gaps.** With JustRL-DeepSeek-1.5B as the teacher, we compare four OPD objectives on two students: Qwen2.5-Math-1.5B (large gap) and DeepSeek-R1-Distill-Qwen-1.5B (small gap). We report Avg@8 accuracy on MATH500.

can estimate the bias with the sampled token $\hat{y}_t$, and add this estimate back to ℓ_t^{TA} to yield an unbiased estimate of the full-vocabulary reverse KL divergence. Formally, the loss function of sample-corrected TA-OPD (SC-TA-OPD) is given by:

$$\ell_t^{\text{SC-TA}} = \sum_{v \in S_t^+} p_t(v)\,\mathrm{sg}\left(\log \frac{p_t(v)}{q_t(v)}\right) + \mathbf{1}[\hat{y}_t \notin S_t^k]\,\frac{p_t(\hat{y}_t)}{\mathrm{sg}(p_t(\hat{y}_t))}\,\mathrm{sg}\left(\log \frac{p_t(\hat{y}_t)/p_t^{\text{tail}}}{q_t(\hat{y}_t)/q_t^{\text{tail}}}\right), \tag{7}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. We next characterize its theoretical properties.

Proposition 5 (Unbiasedness)**.** *For every step t, taking the randomness over $\hat{y}_t \sim p_t$, we have*

$$\mathbb{E}_{\hat{y}_t \sim p_t}\left[\ell_t^{\text{SC-TA}}\right] = D_{\mathrm{KL}}(p_t \,\|\, q_t), \quad \mathbb{E}_{\hat{y}_t \sim p_t}\left[\nabla_\theta\, \ell_t^{\text{SC-TA}}\right] = \nabla_\theta\, D_{\mathrm{KL}}(p_t \,\|\, q_t).$$

The proof is provided in Appendix B.5. Proposition 5 shows that $\ell_t^{\text{SC-TA}}$ is an unbiased estimate of the full-vocabulary reverse KL in both value and gradient, thereby combining the advantages of sampled-token OPD's unbiasedness and top-k OPD's dense supervision.

Table 2 compares TA-OPD with its sample-corrected variant on two student–teacher pairs. Correcting the bias term yields only marginal gains: the macro-average on mathematical reasoning improves from 53.62 to 54.39 for DeepSeek-R1-Distill-Qwen-1.5B and from 44.62 to 44.65 for Qwen3-1.7B, while the OOD averages remain essentially unchanged in both settings (30.69 vs. 30.66 and 67.01 vs. 67.10). We attribute this to the tightness of TA-OPD: its bias relative to full-vocabulary OPD is negligible in practice. Overall, SC-TA-OPD provides rigorous theoretical guarantees of unbiasedness, while TA-OPD achieves performance comparable to SC-TA-OPD.

TA-OPD is particularly effective when the student–teacher capability gap is large. We fix JustRL-DeepSeek-1.5B (He et al., 2025) as the teacher and compare two students, Qwen2.5-Math-1.5B and DeepSeek-R1-Distill-Qwen-1.5B. DeepSeek-R1-Distill-Qwen-1.5B is obtained by applying supervised fine-tuning to Qwen2.5-Math-1.5B, and the teacher is obtained by applying RLVR to DeepSeek-R1-Distill-Qwen-1.5B. The capability gap between Qwen2.5-Math-1.5B and the teacher is therefore larger than that between DeepSeek-R1-Distill-Qwen-1.5B and the teacher.

Figure 6 presents the Avg@8 accuracy of four OPD objectives on MATH500 for the two students. On Qwen2.5-Math-1.5B, TA-OPD attains an Avg@8 of 75.20 on MATH500, outperforming normalized top-k OPD by 13.50 points. On DeepSeek-R1-Distill-Qwen-1.5B, the four objectives achieve similar performance, with sample-corrected TA-OPD attaining the best accuracy of 87.00. The larger improvement on Qwen2.5-Math-1.5B stems from a more pronounced tail probability increase under the larger capability gap, as shown in Figure 20. In summary, TA-OPD brings substantial improvements over baselines when the student–teacher capability gap is large.

7 Related Work

On-policy Distillation. On-policy distillation (OPD) is an effective post-training paradigm that has attracted a surge of interest (Song & Zheng, 2026). To provide dense supervision at tractable cost, top-k OPD has become a popular research direction (Zhu et al., 2026; Xing et al., 2026). Normalized top-k OPD (Li et al., 2026; Fu et al., 2026) directly uses normalized top-k reverse KL divergence as the loss function. Entropy-aware OPD (Jin et al., 2026) improves sampled-token OPD by applying normalized top-k forward KL at high-entropy positions. vOPD (Oh et al., 2026) uses normalized top-k reverse KL as a reward baseline to reduce the variance of sampled-token OPD. While these methods all build on the normalized top-k KL, none of them identifies that directly optimizing it steadily increases the student's tail probability. We note that Dasgupta et al. (2026) studies a related tail probability issue, but it considers off-policy full-vocabulary distillation and is not applicable to top-k OPD. Moreover, the failure mode it studies differs from ours: it studies how full-vocabulary forward KL in the off-policy setting makes the student's tail probability too small, while we study how normalized top-k objectives in the on-policy setting steadily increase it.

KL computation for LLM post-training. KL divergence is a crucial component in post-training of LLMs (Vieillard et al., 2020; Ouyang et al., 2022). In OPD, the student is optimized by minimizing the reverse KL divergence to the teacher (Lu & Lab, 2025), whereas in reinforcement learning (RL), a KL constraint towards the reference model is commonly imposed to prevent the policy from drifting away (Ziegler et al., 2019; Shah et al., 2025). For efficiency, most existing methods estimate the divergence using only the sampled tokens (Schulman et al., 2015; Schulman, 2020). These estimators suffer from high variance and ignore the dense information in the logit space. To provide dense information at tractable cost, top-k approximations of the KL divergence have become an appealing alternative (Zhang & Ba, 2026). Existing top-k OPD methods predominantly adopt a normalized top-k formulation (Li et al., 2026; Fu et al., 2026), which, as we reveal in this work, leads to a steady increase in the student's tail probability. In the context of RL, several top-k KL approximations have also been explored, yet they are devised for purposes different from ours. For instance, DPPO (Qi et al., 2026) derives a KL approximation similar to TA-OPD's loss objective, but uses it as a quantity to decide whether to clip the policy update on a token, thereby substituting heuristic PPO clipping (Schulman et al., 2017) with a principled constraint. In contrast, we directly use the top-k KL approximation as the optimization objective and establish its theoretical properties. Besides, EMA-PG (Zhang & Ba, 2026) proposes a KL approximation similar to our sample-corrected TA-OPD. We provide a discussion of sample-corrected TA-OPD and EMA-PG in Appendix F.7.

8 Conclusion

In this paper, we introduce Tail-Aware Top-k On-Policy Distillation (TA-OPD), a novel distillation method that restores the missing tail probability signal. In particular, TA-OPD minimizes the reverse KL divergence over the teacher's top-k tokens plus a tail token that carries the tail probability. In effect, TA-OPD explicitly aligns the student's tail probability with the teacher's, thereby addressing the tail-probability and entropy increases caused by top-k normalization. We further derive a sampled variant that yields an unbiased estimate of the full-vocabulary reverse KL when the sampled token's probability is available. Extensive experiments show that TA-OPD consistently improves accuracy across benchmarks. Our method can be easily adopted in practice: it is straightforward to implement with existing OPD frameworks and requires no additional teacher queries beyond the top-k probabilities. We hope our insights inspire future research to further explore loss-function designs for OPD.

AI USE STATEMENT

We did not use generative AI tools for any of the tasks listed under required disclosure in the ICLR AI Policy for Authors. In particular, the conceptual framework, mathematical claims, proofs, hypotheses, experimental design, and method implementation were developed and written by the authors; the synthetic data in Appendix F.4 were generated by author-written code; and all results were interpreted by the authors. The remaining required-disclosure tasks—translation, dataset cleaning and reformatting, and qualitative or thematic data analysis—are not applicable to this work. Additionally, we used generative AI tools for two tasks with recommended disclosure: editing the paper to improve readability, and creating or editing auxiliary software code, such as the plotting scripts used to produce our figures. We have reviewed all AI-assisted work: all AI-edited text was checked and revised by the authors to ensure it aligned with the intended meaning, and all AI-assisted code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

In this work, we study how to restore the missing tail probability signal in top-k OPD. As our contribution focuses on designing a better loss function for OPD, we do not see any specific ethical concerns arising from this study.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. Code for reproducing our main experiments is available in this anonymous repository. The experimental setup, including datasets, models, and hyperparameter settings, is described in detail in Appendix E. We have also provided a full description of the prompts, software details, and hardware details in Appendix E to assist others in reproducing our experiments. Additionally, all datasets used, including MATH500 and AIME 2024, are publicly available, ensuring consistent and reproducible evaluation results. We believe these measures will enable other researchers to reproduce our work and further advance the field.

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*, volume 2024, pp. 21246–21263, 2024.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

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

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. Don't ignore the tail: Decoupling top-k probabilities for efficient language model distillation. *arXiv preprint arXiv:2602.20816*, 2026.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

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

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Advances in Neural Information Processing Systems*, 2021.

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

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

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

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

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

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. 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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, volume 2025, pp. 70523–70563, 2025.

John Schulman. Approximating kl divergence. *John Schulman's Homepage*, 5, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Vedant Shah, Johan Obando-Ceron, Vineet Jain, Brian Bartoldson, Bhavya Kailkhura, Sarthak Mittal, Glen Berseth, Pablo Samuel Castro, Yoshua Bengio, Nikolay Malkin, et al. A comedy of estimators: On kl regularization in rl training of llms. *arXiv preprint arXiv:2512.21852*, 2025.

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

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

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

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

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.

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

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

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Haoran Zhang, Yafu Li, Zhi Wang, Zhilin Wang, Shunkai Zhang, Xiaoye Qu, and Yu Cheng. Characterizing, evaluating, and optimizing complex reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IMFgiWw4jd.

Lunjun Zhang and Jimmy Ba. Ema policy gradient: Taming reinforcement learning for llms with ema anchor and top-k kl. *arXiv preprint arXiv:2602.04417*, 2026.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A LIMITATIONS

The performance gain of TA-OPD over normalized top-k OPD diminishes as k increases or the capability gap between student and teacher decreases. Additionally, due to limited computational resources, our experiments are restricted to models with up to 8B parameters. While these sizes are standard for research-stage OPD studies, we do not presume automatic transfer to 30B+ models.

B PROOFS

B.1 PROOF OF PROPOSITION 1

Proof. Fix a step t and a prefix $\hat{y}_{<t}$, and omit the subscript t throughout: write $S = S_t^k = \mathrm{TopK}(q_t, k)$ for the teacher's top-k tokens, $z_v = z_{t,v}$ for the student logit of token v, and $\bar{p} = \bar{p}_t^{(S_t^k)}$, $\bar{q} = \bar{q}_t^{(S_t^k)}$ for the normalized distributions on S, so that

$$\ell^{\mathrm{norm}} = \sum_{u \in S} \bar{p}(u) \log \frac{\bar{p}(u)}{\bar{q}(u)}.$$

Since the token set S is selected from the teacher distribution, both S and $\bar{q}$ are constants with respect to the student logits. We proceed in three steps: we first show that $\bar{p}$ is a softmax over the logits restricted to top-k tokens S, then differentiate the loss through it, and finally sum the resulting gradients over S.

Step 1: $\bar{p}$ can be expressed as a softmax restricted to top-k tokens. Writing the student distribution as $p(v) = e^{z_v} / \sum_{u \in \mathcal{V}} e^{z_u}$, the vocabulary-level normalizer appears in both the numerator and the denominator of $\bar{p}$ and therefore cancels,

$$\bar{p}(v) = \frac{p(v)}{\sum_{w \in S} p(w)} = \frac{e^{z_v}}{\sum_{w \in S} e^{z_w}}, \qquad v \in S. \tag{8}$$

Two consequences of Eq. (8) drive the whole proof: $\bar{p}$ is a softmax over the sub-vector $(z_w)_{w \in S}$, and it does not depend on any logit outside S. Its Jacobian is therefore the standard softmax Jacobian on S and vanishes elsewhere,

$$\frac{\partial \bar{p}(u)}{\partial z_v} = \begin{cases} \bar{p}(u)\big(\mathbf{1}[u = v] - \bar{p}(v)\big), & u, v \in S, \\ 0, & v \notin S. \end{cases} \tag{9}$$

Step 2: the gradient with respect to the logits. Because $\bar{q}$ is fixed with respect to the student logits, the loss depends on the student only through $\{\bar{p}(u)\}_{u \in S}$, with

$$\frac{\partial \ell^{\mathrm{norm}}}{\partial \bar{p}(u)} = \log \frac{\bar{p}(u)}{\bar{q}(u)} + 1, \qquad u \in S.$$

The chain rule over the vocabulary thus gives

$$\frac{\partial \ell^{\mathrm{norm}}}{\partial z_v} = \sum_{u \in S} \frac{\partial \ell^{\mathrm{norm}}}{\partial \bar{p}(u)} \times \frac{\partial \bar{p}(u)}{\partial z_v} = \sum_{u \in S} \left(\log \frac{\bar{p}(u)}{\bar{q}(u)} + 1 \right) \frac{\partial \bar{p}(u)}{\partial z_v}, \tag{10}$$

where the terms with $u \notin S$ are already dropped because ℓ^{norm} does not involve them. For $v \notin S$, every Jacobian entry in Eq. (9) is zero, hence $\partial \ell^{\mathrm{norm}} / \partial z_v = 0$, which is the second case of the claim: the loss is blind to the logits outside the teacher's top-k tokens. For $v \in S$, substituting Eq. (9) into Eq. (10) and separating the diagonal term $u = v$ gives

$$\frac{\partial \ell^{\mathrm{norm}}}{\partial z_v} = \bar{p}(v) \left(\log \frac{\bar{p}(v)}{\bar{q}(v)} + 1 \right) - \bar{p}(v) \sum_{u \in S} \bar{p}(u) \left(\log \frac{\bar{p}(u)}{\bar{q}(u)} + 1 \right). \tag{11}$$

Note that

$$\sum_{u \in S} \bar{p}(u) \left(\log \frac{\bar{p}(u)}{\bar{q}(u)} + 1 \right) = \sum_{u \in S} \bar{p}(u) \log \frac{\bar{p}(u)}{\bar{q}(u)} + \sum_{u \in S} \bar{p}(u) = \ell^{\mathrm{norm}} + 1.$$

By substituting it into Eq. (11), we obtain

$$\frac{\partial \ell^{\mathrm{norm}}}{\partial z_v} = \bar{p}(v)\left[\log\frac{\bar{p}(v)}{\bar{q}(v)} - \ell^{\mathrm{norm}}\right], \qquad v \in S, \tag{12}$$

which is the first case of the claim. Since $\ell^{\mathrm{norm}} = \mathbb{E}_{v\sim\bar{p}}\big[\log(\bar{p}(v)/\bar{q}(v))\big]$ by definition, Eq. (12) says that the gradient on a top-k logit is its log-ratio centered at the $\bar{p}$-weighted mean log-ratio.

Step 3: the top-k gradients sum to zero. Summing Eq. (12) over $v \in S$ and using $\sum_{v\in S}\bar{p}(v)\log\big(\bar{p}(v)/\bar{q}(v)\big) = \ell^{\mathrm{norm}}$ together with $\sum_{v\in S}\bar{p}(v) = 1$ yields

$$\sum_{v\in S}\frac{\partial \ell^{\mathrm{norm}}}{\partial z_v} = \ell^{\mathrm{norm}} - \ell^{\mathrm{norm}} = 0.$$

This is precisely the mean-centering exhibited in Eq. (12): the update redistributes logit mass among the top-k tokens without changing their total, while leaving the tail logits untouched by Step 2. The normalized objective therefore acts only on the shape of the student distribution within S, and never on how much probability the student assigns to S as a whole. □

B.2 Proof of Proposition 2

Proof. Fix a step t and a prefix $\hat{y}_{<t}$, and omit the subscript t throughout, following the convention of Appendix B.1: write $S = S_t^k$, $z_v = z_{t,v}$, $\bar{p} = \bar{p}_t^{(S_t^k)}$, $\bar{q} = \bar{q}_t^{(S_t^k)}$, and $p^{\mathrm{tail}} = p_t^{\mathrm{tail}}$. Under the tabular assumption, each logit z_v is an independent parameter, so the gradient step acts on the logits directly; this assumption is commonly adopted by prior work for analysis (Cui et al., 2025; Zhang et al., 2026; Ren & Sutherland, 2025). We write $p^{\mathrm{tail}}(\eta)$ for the tail probability after one gradient step with learning rate η, and $\Delta p^{\mathrm{tail}} = p^{\mathrm{tail}}(\eta) - p^{\mathrm{tail}}(0)$ for its change. We proceed in three steps: we first reduce $p^{\mathrm{tail}}(\eta)$ to a one-dimensional function of the step size, then identify the derivative of that function as a covariance, and finally expand it to first order in η.

Step 1: only the top-k logits change. Split the softmax normalizer into the contributions of the top-k tokens and of the tail tokens,

$$A := \sum_{v\in S} e^{z_v}, \qquad B := \sum_{v\notin S} e^{z_v}, \qquad Z := A + B,$$

so that $p^{\mathrm{tail}} = B/Z$. By Proposition 1, the gradient of ℓ^{norm} is

$$g_v = \bar{p}(v)\left(\log\frac{\bar{p}(v)}{\bar{q}(v)} - \ell^{\mathrm{norm}}\right) \text{ for } v \in S, \qquad g_v = 0 \text{ for } v \notin S.$$

The update $z_v \leftarrow z_v - \eta g_v$ therefore leaves the tail logits unchanged, and hence leaves B unchanged. The tail probability change is thus determined by A alone:

$$p^{\mathrm{tail}}(\eta) = \frac{B}{A(\eta) + B}, \qquad A(\eta) = \sum_{v\in S} e^{z_v - \eta g_v}. \tag{13}$$

Step 2: the derivative of A is a covariance. Differentiating $A(\eta)$ in Eq. (13) with respect to η and evaluating at $\eta = 0$ gives

$$A'(0) = -\sum_{v\in S} e^{z_v} g_v.$$

We now rewrite this sum in terms of the normalized distribution $\bar{p}$. Using $e^{z_v} = Z\,p(v)$ and $p(v) = (1 - p^{\mathrm{tail}})\,\bar{p}(v)$ for $v \in S$, and then substituting the expression for g_v from Step 1,

$$A'(0) = -Z\big(1 - p^{\mathrm{tail}}\big)\sum_{v\in S}\bar{p}(v)\,g_v = -Z\big(1 - p^{\mathrm{tail}}\big)\left[\sum_{v\in S}\bar{p}(v)^2\log\frac{\bar{p}(v)}{\bar{q}(v)} - \ell^{\mathrm{norm}}\sum_{v\in S}\bar{p}(v)^2\right]. \tag{14}$$

Each of the two sums in Eq. (14) carries one factor $\bar{p}(v)$ that plays the role of a sampling weight, so both are expectations under $v \sim \bar{p}$:

$$\sum_{v \in S} \bar{p}(v)^2 \log \frac{\bar{p}(v)}{\bar{q}(v)} = \mathbb{E}_{v \sim \bar{p}}\left[\bar{p}(v) \log \frac{\bar{p}(v)}{\bar{q}(v)}\right], \qquad \sum_{v \in S} \bar{p}(v)^2 = \mathbb{E}_{v \sim \bar{p}}\big[\bar{p}(v)\big].$$

Moreover $\ell^{\mathrm{norm}} = \mathbb{E}_{v \sim \bar{p}}\big[\log(\bar{p}(v)/\bar{q}(v))\big]$ by definition, so the bracket in Eq. (14) is the difference between the expectation of a product and the product of the expectations, which is exactly a covariance:

$$A'(0) = -Z\big(1 - p^{\mathrm{tail}}\big) \operatorname*{Cov}_{v \sim \bar{p}}\left(\bar{p}(v),\, \log \frac{\bar{p}(v)}{\bar{q}(v)}\right). \tag{15}$$

The covariance appears because, by Proposition 1, the gradient on a top-k logit is its log-ratio centered at the mean log-ratio, while its contribution to A is weighted by its own probability $\bar{p}(v)$.

Step 3: first-order Taylor expansion. Differentiating $p^{\mathrm{tail}}(\eta)$ in Eq. (13) at $\eta = 0$ and substituting Eq. (15) together with $B/Z = p^{\mathrm{tail}}$ yields

$$\left.\frac{\mathrm{d}p^{\mathrm{tail}}}{\mathrm{d}\eta}\right|_{\eta=0} = -\frac{B}{(A+B)^2}\, A'(0) = p^{\mathrm{tail}}\big(1 - p^{\mathrm{tail}}\big) \operatorname*{Cov}_{v \sim \bar{p}}\left(\bar{p}(v),\, \log \frac{\bar{p}(v)}{\bar{q}(v)}\right).$$

A first-order Taylor expansion in η, followed by splitting the log-ratio as $\log \bar{p}(v) - \log \bar{q}(v)$ and using the linearity of the covariance in its second argument, gives the stated identity:

$$\Delta p^{\mathrm{tail}} = \eta\, p^{\mathrm{tail}}\big(1 - p^{\mathrm{tail}}\big)\Big[\operatorname*{Cov}_{v \sim \bar{p}}\big(\bar{p}(v), \log \bar{p}(v)\big) - \operatorname*{Cov}_{v \sim \bar{p}}\big(\bar{p}(v), \log \bar{q}(v)\big)\Big] + O(\eta^2).$$

Since $p^{\mathrm{tail}} \in (0,1)$, the prefactor $p^{\mathrm{tail}}(1-p^{\mathrm{tail}})$ is strictly positive, so for sufficiently small $\eta > 0$ the sign of Δp^{tail} is determined by the sign of the covariance difference. The tail probability, therefore, increases if and only if the student self-covariance exceeds the student–teacher covariance. □

B.3 Proof of Proposition 3

Proof. Fix a step t and a prefix $\hat{y}_{<t}$, and omit the subscript t throughout, following the convention of Appendix B.1: write $S = S_t^k$, $z_v = z_{t,v}$, $p(v) = p_t(v)$, $q(v) = q_t(v)$, and $p^{\mathrm{tail}} = p_t^{\mathrm{tail}}$, $q^{\mathrm{tail}} = q_t^{\mathrm{tail}}$, so that

$$\ell^{\mathrm{TA}} = \sum_{u \in S} p(u) \log \frac{p(u)}{q(u)} + p^{\mathrm{tail}} \log \frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}}, \qquad p^{\mathrm{tail}} = \sum_{u \notin S} p(u).$$

Since S is selected from the teacher distribution, S, q, and q^{tail} are constants with respect to the student logits. Unlike the normalized loss of Proposition 1, ℓ^{TA} depends on the student through the *unnormalized* probabilities, and the tail tokens enter it through their aggregate p^{tail}. We proceed in three steps: we first differentiate ℓ^{TA} with respect to the probabilities of all tokens, then push the derivative through the full softmax Jacobian, and finally read off the two cases of the claim.

Step 1: the derivative with respect to the probabilities. We treat ℓ^{TA} as a function of the full probability vector $\big(p(u)\big)_{u \in \mathcal{V}}$ rather than eliminating p^{tail}. A top-k token $u \in S$ appears only in its own summand, and a tail token $u \notin S$ appears only inside p^{tail}, on which it acts with $\partial p^{\mathrm{tail}}/\partial p(u) = 1$. Hence

$$\frac{\partial \ell^{\mathrm{TA}}}{\partial p(u)} = \begin{cases} \log \dfrac{p(u)}{q(u)} + 1, & u \in S, \\[2ex] \log \dfrac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}} + 1, & u \notin S. \end{cases} \tag{16}$$

Step 2: pushing through the softmax Jacobian. The student distribution is a softmax over the full vocabulary, $p(u) = e^{z_u} / \sum_{w \in \mathcal{V}} e^{z_w}$, whose Jacobian is $\partial p(u)/\partial z_v = p(u)\big(\mathbf{1}[u = v] - p(v)\big)$ for all $u, v \in \mathcal{V}$. Combining it with Eq. (16) through the chain rule gives

$$\frac{\partial \ell^{\mathrm{TA}}}{\partial z_v} = \sum_{u \in \mathcal{V}} \frac{\partial \ell^{\mathrm{TA}}}{\partial p(u)} \frac{\partial p(u)}{\partial z_v} = p(v) \left.\frac{\partial \ell^{\mathrm{TA}}}{\partial p(u)}\right|_{u=v} - p(v) \sum_{u \in \mathcal{V}} p(u) \frac{\partial \ell^{\mathrm{TA}}}{\partial p(u)}. \tag{17}$$

Note that for the second term, we have

$$\sum_{u\in\mathcal{V}} p(u)\,\frac{\partial \ell^{\mathrm{TA}}}{\partial p(u)} = \sum_{u\in S} p(u)\left(\log\frac{p(u)}{q(u)}+1\right) + \left(\sum_{u\notin S} p(u)\right)\left(\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}}+1\right) = \ell^{\mathrm{TA}}+1,$$

where the second equality uses $\sum_{u\notin S} p(u) = p^{\mathrm{tail}}$ to recover the tail term of ℓ^{TA}, and $\sum_{u\in\mathcal{V}} p(u) = 1$ to collect the constant.

Step 3: the gradient on each logit. Substituting $\ell^{\mathrm{TA}}+1$ for the sum in Eq. (17), we have

$$\frac{\partial \ell^{\mathrm{TA}}}{\partial z_v} = \begin{cases} p(v)\left(\log\dfrac{p(v)}{q(v)} - \ell^{\mathrm{TA}}\right), & v\in S, \\ p(v)\left(\log\dfrac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}} - \ell^{\mathrm{TA}}\right), & v\notin S, \end{cases} \tag{18}$$

which establishes (5). □

B.4 Proof of Proposition 4

Proof. Fix a step t and drop the subscript t as in Appendix B.3. The two loss functions are

$$\ell^{\mathrm{full}} = \sum_{v\in S} p(v)\log\frac{p(v)}{q(v)} + \sum_{v\notin S} p(v)\log\frac{p(v)}{q(v)}, \qquad \ell^{\mathrm{TA}} = \sum_{v\in S} p(v)\log\frac{p(v)}{q(v)} + p^{\mathrm{tail}}\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}},$$

where we have split $\ell^{\mathrm{full}} = D_{\mathrm{KL}}(p\|q)$ into the top-k tokens and the tail tokens so that the two losses can be compared term-by-term. The two objectives share the same top-k term and differ only in how they treat the tail, so taking the difference cancels the top-k term and leaves

$$\ell^{\mathrm{full}} - \ell^{\mathrm{TA}} = \sum_{v\notin S} p(v)\log\frac{p(v)}{q(v)} - p^{\mathrm{tail}}\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}}. \tag{19}$$

It remains to compute the right-hand side of (19). Substituting $p(v) = p^{\mathrm{tail}}\,\tilde{p}(v)$ and $q(v) = q^{\mathrm{tail}}\,\tilde{q}(v)$ for $v\notin S$ into its first term and splitting the logarithm gives

$$\begin{aligned} \sum_{v\notin S} p(v)\log\frac{p(v)}{q(v)} &= \sum_{v\notin S} p^{\mathrm{tail}}\,\tilde{p}(v)\log\frac{p^{\mathrm{tail}}\,\tilde{p}(v)}{q^{\mathrm{tail}}\,\tilde{q}(v)} \\ &= p^{\mathrm{tail}}\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}}\sum_{v\notin S}\tilde{p}(v) + p^{\mathrm{tail}}\sum_{v\notin S}\tilde{p}(v)\log\frac{\tilde{p}(v)}{\tilde{q}(v)} \\ &= p^{\mathrm{tail}}\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}} + p^{\mathrm{tail}}\,D_{\mathrm{KL}}(\tilde{p}\|\tilde{q}), \end{aligned}$$

where the last equality uses that $\tilde{p}$ is a probability distribution over the tail tokens, $\sum_{v\notin S}\tilde{p}(v) = 1$, and recognizes the remaining sum as $D_{\mathrm{KL}}(\tilde{p}\|\tilde{q})$.

Substituting this back into (19), the terms $p^{\mathrm{tail}}\log\frac{p^{\mathrm{tail}}}{q^{\mathrm{tail}}}$ cancel and we obtain

$$\ell^{\mathrm{full}} - \ell^{\mathrm{TA}} = p^{\mathrm{tail}}\,D_{\mathrm{KL}}(\tilde{p}\|\tilde{q}).$$

□

B.5 Proof of Proposition 5

Proof. Fix a step t and drop the subscript t as in Appendix B.3: write $S = S_t^k$, $S^+ = S_t^+$, $p(v) = p_t(v)$, $q(v) = q_t(v)$, $\hat{y} = \hat{y}_t$, $p^{\mathrm{tail}} = p_t^{\mathrm{tail}}$, and $q^{\mathrm{tail}} = q_t^{\mathrm{tail}}$. Recall that the stop-gradient operator $\mathrm{sg}(\cdot)$ acts as the identity in the forward pass and has zero derivative, i.e., $\mathrm{sg}(x) = x$ in value and $\nabla_\theta\,\mathrm{sg}(x) = 0$. Throughout, the expectation is taken with respect to $\hat{y}\sim p$, where the sampling distribution is fixed at the current parameters. We will repeatedly use the identity

$$\mathbb{E}_{\hat{y}\sim p}\big[\mathbf{1}[\hat{y}\notin S]\big] = \sum_{v\notin S} p(v) = p^{\mathrm{tail}}. \tag{20}$$

Unbiasedness in value. Without considering the gradient, $\frac{p(\hat{y})}{\mathrm{sg}(p(\hat{y}))} = 1$, so

$$\begin{aligned}\ell^{\text{SC-TA}} &= \sum_{v \in S^+} p(v)\,\mathrm{sg}\left(\log \frac{p(v)}{q(v)}\right) + \mathbf{1}[\hat{y} \notin S]\,\frac{p(\hat{y})}{\mathrm{sg}(p(\hat{y}))}\,\mathrm{sg}\left(\log \frac{p(\hat{y})/p^{\text{tail}}}{q(\hat{y})/q^{\text{tail}}}\right) \\ &= \sum_{v \in S} p(v) \log \frac{p(v)}{q(v)} + p^{\text{tail}} \log \frac{p^{\text{tail}}}{q^{\text{tail}}} + \mathbf{1}[\hat{y} \notin S]\left(\log \frac{p(\hat{y})}{q(\hat{y})} - \log \frac{p^{\text{tail}}}{q^{\text{tail}}}\right), \end{aligned} \tag{21}$$

where the second equality expands $S^+ = S \cup \{v_{\text{tail}}\}$ using $p(v_{\text{tail}}) = p^{\text{tail}}$ and $q(v_{\text{tail}}) = q^{\text{tail}}$. The first two terms are deterministic given the prefix. For the third term, taking the expectation over $\hat{y} \sim p$ and using Eq. (20) gives

$$\mathbb{E}_{\hat{y}\sim p}\left[\mathbf{1}[\hat{y} \notin S]\left(\log \frac{p(\hat{y})}{q(\hat{y})} - \log \frac{p^{\text{tail}}}{q^{\text{tail}}}\right)\right] = \sum_{v \notin S} p(v) \log \frac{p(v)}{q(v)} - p^{\text{tail}} \log \frac{p^{\text{tail}}}{q^{\text{tail}}}.$$

Summing the three terms, the two occurrences of $p^{\text{tail}} \log \frac{p^{\text{tail}}}{q^{\text{tail}}}$ cancel and we obtain

$$\mathbb{E}_{\hat{y}\sim p}\big[\ell^{\text{SC-TA}}\big] = \sum_{v \in S} p(v) \log \frac{p(v)}{q(v)} + \sum_{v \notin S} p(v) \log \frac{p(v)}{q(v)} = D_{\mathrm{KL}}(p \,\|\, q),$$

which establishes the unbiasedness in value.

Unbiasedness in gradient. We first compute the gradient of the exact full-vocabulary reverse KL. Since q is fixed,

$$\nabla_\theta\, D_{\mathrm{KL}}(p \,\|\, q) = \sum_{v \in \mathcal{V}} \nabla_\theta p(v)\left(\log \frac{p(v)}{q(v)} + 1\right) = \sum_{v \in \mathcal{V}} \nabla_\theta p(v)\ \log \frac{p(v)}{q(v)},$$

where the last equality uses $\sum_{v\in\mathcal{V}} \nabla_\theta p(v) = \nabla_\theta \sum_{v\in\mathcal{V}} p(v) = \nabla_\theta 1 = 0$.

We now differentiate $\ell^{\text{SC-TA}}$. Since the log-ratios are wrapped in stop-gradients, only the leading probabilities carry gradients in the first term. Note that $p^{\text{tail}} = \sum_{v\notin S} p(v)$ depends on θ, so $\nabla_\theta p^{\text{tail}} = \sum_{v\notin S} \nabla_\theta p(v)$, and therefore

$$\nabla_\theta \sum_{v \in S^+} p(v)\,\mathrm{sg}\left(\log \frac{p(v)}{q(v)}\right) = \sum_{v \in S} \nabla_\theta p(v)\ \log \frac{p(v)}{q(v)} + \Big(\sum_{v \notin S} \nabla_\theta p(v)\Big) \log \frac{p^{\text{tail}}}{q^{\text{tail}}}.$$

For the second term, only the numerator $p(\hat{y})$ in $\frac{p(\hat{y})}{\mathrm{sg}(p(\hat{y}))}$ carries gradients, so

$$\nabla_\theta\left[\mathbf{1}[\hat{y} \notin S]\,\frac{p(\hat{y})}{\mathrm{sg}(p(\hat{y}))}\,\mathrm{sg}\left(\log \frac{p(\hat{y})/p^{\text{tail}}}{q(\hat{y})/q^{\text{tail}}}\right)\right] = \mathbf{1}[\hat{y} \notin S]\,\frac{\nabla_\theta p(\hat{y})}{p(\hat{y})}\left(\log \frac{p(\hat{y})}{q(\hat{y})} - \log \frac{p^{\text{tail}}}{q^{\text{tail}}}\right).$$

Taking the expectation over $\hat{y} \sim p$ of the second term gives

$$\begin{aligned}&\mathbb{E}_{\hat{y}\sim p}\left[\mathbf{1}[\hat{y} \notin S]\,\frac{\nabla_\theta p(\hat{y})}{p(\hat{y})}\left(\log \frac{p(\hat{y})}{q(\hat{y})} - \log \frac{p^{\text{tail}}}{q^{\text{tail}}}\right)\right] \\ &\quad= \sum_{v \notin S} p(v) \cdot \frac{\nabla_\theta p(v)}{p(v)}\left(\log \frac{p(v)}{q(v)} - \log \frac{p^{\text{tail}}}{q^{\text{tail}}}\right) = \sum_{v \notin S} \nabla_\theta p(v)\ \log \frac{p(v)}{q(v)} - \Big(\sum_{v \notin S} \nabla_\theta p(v)\Big) \log \frac{p^{\text{tail}}}{q^{\text{tail}}}.\end{aligned}$$

Combining the two terms, the contributions involving $\log \frac{p^{\text{tail}}}{q^{\text{tail}}}$ cancel exactly and we obtain

$$\mathbb{E}_{\hat{y}\sim p}\big[\nabla_\theta\, \ell^{\text{SC-TA}}\big] = \sum_{v \in S} \nabla_\theta p(v)\ \log \frac{p(v)}{q(v)} + \sum_{v \notin S} \nabla_\theta p(v)\ \log \frac{p(v)}{q(v)} = \nabla_\theta\, D_{\mathrm{KL}}(p \,\|\, q),$$

which establishes the unbiasedness in gradient. This completes the proof. □

C Detailed Analysis of the Tail Probability Increase

In this section, we conduct several controlled experiments to identify when the tail probability increase occurs. We find that it arises under two conditions: a large capability gap between the student and teacher, and an insufficiently large k.

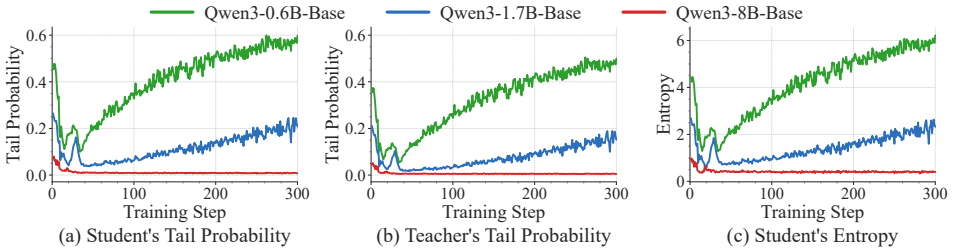


Figure 7: **The student's tail probability, the teacher's tail probability, and the student's entropy across training steps under different student–teacher capability gaps.** The teacher is Qwen3-8B, and the students are Qwen3-0.6B-Base, Qwen3-1.7B-Base, and Qwen3-8B-Base. The tail probability increase is most pronounced for Qwen3-0.6B-Base, weaker for Qwen3-1.7B-Base, and absent for Qwen3-8B-Base.

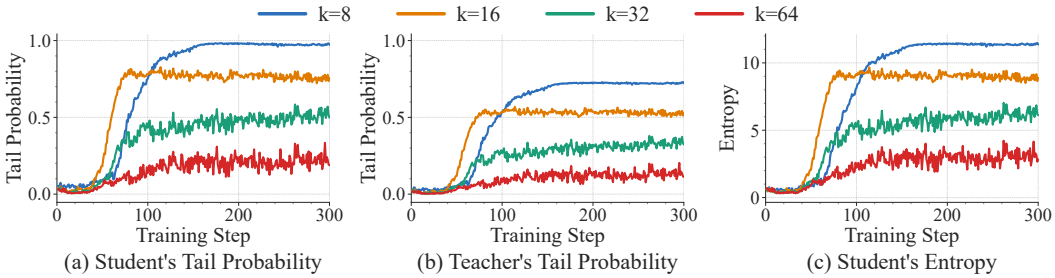


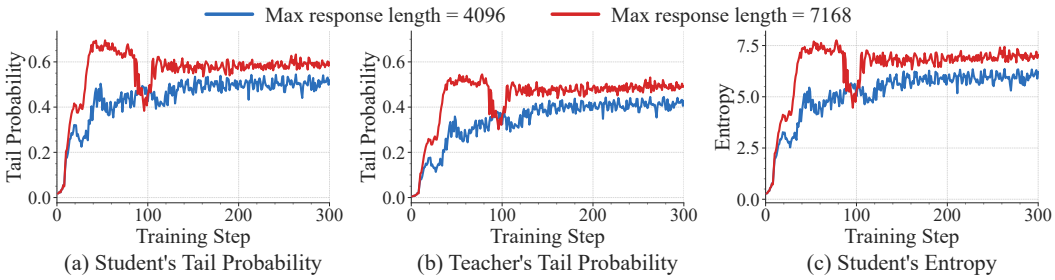
Figure 8: **The student's tail probability, the teacher's tail probability, and the student's entropy across training steps.** The student and teacher models are Qwen2.5-Math-1.5B and DeepSeek-R1-Distill-Qwen-7B, respectively. The tail probability increase occurs when $k \leq 64$.

The tail probability increase occurs when the student–teacher capability gap is large. We conduct an ablation study to examine how the student–teacher capability gap affects the tail probability increase. Specifically, we fix Qwen3-8B as the teacher and compare three students, Qwen3-0.6B-Base, Qwen3-1.7B-Base, and Qwen3-8B-Base, whose capability gaps to the teacher decrease in that order. For this ablation study, we set the maximum response length to 4096.

In Figure 7, we visualize the student's and teacher's tail probabilities on student-generated prefixes, together with the student's entropy. The tail probability increase becomes more pronounced as the capability gap grows. For Qwen3-0.6B-Base, the student's tail probability and entropy rise most sharply over training steps; for Qwen3-1.7B-Base, they also increase but to a lesser extent; for Qwen3-8B-Base, both remain low and stable. Overall, the tail probability increase arises when the student–teacher capability gap is large and is mitigated as the gap narrows.

The tail probability increase occurs when k is not sufficiently large. We conduct an ablation study on how k affects the tail probability increase by fixing the student and teacher, and varying only k. Specifically, we use Qwen2.5-Math-1.5B as the student and DeepSeek-R1-Distill-Qwen-7B as the teacher, and sweep $k \in \{8, 16, 32, 64\}$. For this ablation study, we set the training max response length to 4096.

In Figure 8, we visualize the student's and teacher's tail probabilities on student-generated prefixes, together with the student's entropy, across different values of k. The tail probability increase emerges when k is not sufficiently large ($k \leq 64$): over training steps, the student's tail probability steadily increases together with its entropy. The effect becomes stronger as k decreases, with smaller k yielding higher entropy and a larger increase in tail probability. In principle, this issue vanishes as k approaches the vocabulary size, since the top-k objective then reduces to full-vocabulary OPD, under which the tail probability is explicitly matched to the teacher's. This behavior is expected from Proposition 2: the magnitude of the single-step increase is proportional to $p_t^{\text{tail}}(1 - p_t^{\text{tail}})$, which shrinks as $p_t^{\text{tail}} \to 0$. Since a larger k yields a smaller initial tail probability, each update



(a) Student's Tail Probability (b) Teacher's Tail Probability (c) Student's Entropy

Figure 9: **The student's tail probability, the teacher's tail probability, and the student's entropy across training steps under different maximum response lengths.** The student and teacher models are Qwen2.5-7B-Instruct and OpenThinker3-7B, respectively. The tail probability increase becomes more pronounced as the maximum response length grows.

moves p_t^{tail} less, mitigating the compounding increase over training. Overall, the tail probability increase arises when k is not sufficiently large and is mitigated as k increases.

The tail probability increase becomes more pronounced with longer responses. We study how the response length affects the tail probability increase by fixing the student and teacher, and varying only the maximum response length. Specifically, we use Qwen2.5-7B-Instruct as the student and OpenThinker3-7B as the teacher, and compare maximum response lengths of 4096 and 7168 tokens.

In Figure 9, we visualize the student's and teacher's tail probabilities on student-generated prefixes, together with the student's entropy, under the two length settings. The results show that the tail probability increase is amplified under the longer response length: the student's tail probability and entropy grow faster and reach higher values throughout training. We attribute this to error compounding along longer trajectories: as the student's tail probability rises, sampling more tokens per response increases the chance of drifting outside the teacher's top-k tokens, driving the student toward uncertain prefixes where the teacher's supervision is unreliable and further inflating the tail probability. Overall, the tail probability increase is exacerbated by longer response lengths, suggesting that the issue is particularly concerning for long-horizon tasks.

The tail probability increase does not occur with student top-k OPD. Prior work also adopts the student's top-k tokens as the support (Li et al., 2026). We examine this choice with Qwen3-1.7B-Base as the student and Qwen3-8B as the teacher, varying only the support between $\mathrm{TopK}(q_t, k)$ and $\mathrm{TopK}(p_t, k)$. In each case, the tail probabilities of both models are computed with respect to the support in use, i.e., the teacher's top-k tokens for the teacher top-k variant and the student's top-k tokens for the student top-k variant.

As shown in Figure 10, the tail probability increase is specific to the teacher top-k support. Two mechanisms explain this. First, among all token sets of size k, $\mathrm{TopK}(p_t, k)$ is the one that leaves the least probability mass outside it, so the student's tail probability starts near zero. By Proposition 2, the per-step change is proportional to $p_t^{\text{tail}}(1 - p_t^{\text{tail}})$, and each update therefore moves p_t^{tail} only slightly. Second, the student top-k support is changing given a prefix: it is recomputed from the student at every step, so a token whose probability is pushed down simply drops out of the support and is replaced by another token that the student now ranks in its top k. The support therefore always consists of the student's k most probable tokens, and the probability mass it covers cannot leak away. The teacher top-k support, in contrast, is fixed by the teacher and does not follow the student, so the mass pushed outside it stays outside and accumulates across steps.

Teacher top-k OPD yields better downstream performance than student top-k OPD. Although the student top-k support avoids the tail probability increase, it underperforms the teacher top-k support on downstream benchmarks (Table 3). We attribute this to two factors. First, as shown in Figure 11, the average rollout length under the student top-k support grows monotonically and saturates at the maximum response length of 7168 tokens within the first 50 steps, while the teacher top-k variant maintains a healthy response length. Second, the teacher top-k support directly covers the teacher's high-probability tokens, whereas the student's top-k tokens may carry little teacher

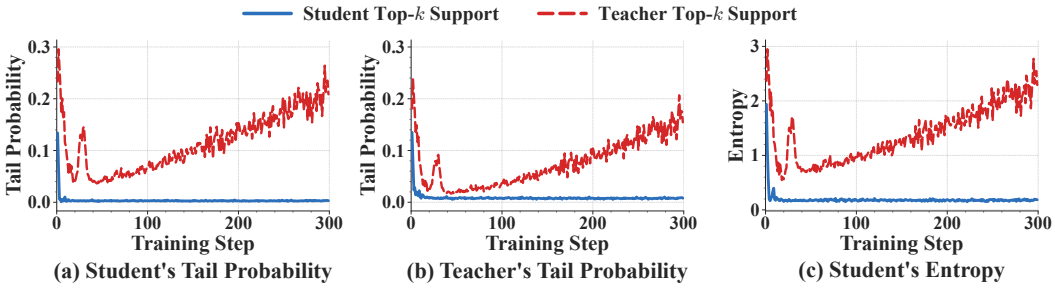


Figure 10: **The student's tail probability, the teacher's tail probability, and the student's entropy across training steps under the student and teacher top-k supports.** The student and teacher models are Qwen3-1.7B-Base and Qwen3-8B. The tail probabilities are computed with respect to the support in use. The tail probability increase occurs only under the teacher top-k support.

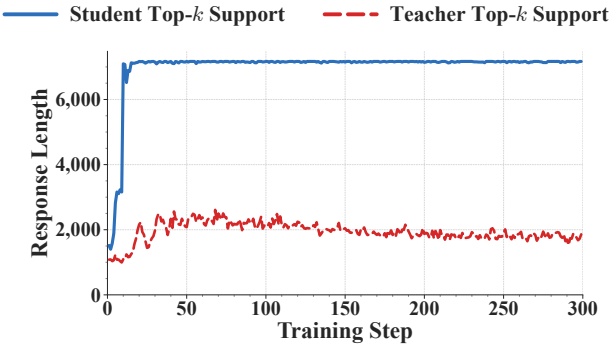


Figure 11: **Average response length across training steps under the student and teacher top-k supports.** The student and teacher models are Qwen3-1.7B-Base and Qwen3-8B, respectively. Under the student top-k support, the response length saturates at the maximum response length of 7168 tokens, while the teacher top-k variant maintains a healthy response length.

probability, allowing the objective to be reduced without moving the student toward the teacher's high-probability modes. Given the superior downstream performance, we mainly study the teacher top-k OPD in this work.

D Compared Methods

Sampled-token OPD. Sampled-token OPD (Lu & Lab, 2025; Xiao et al., 2026) supervises only the token $\hat{y}_t \sim p_t$ sampled by the student:

$$\ell_t^{\text{sample}} = \log p_t(\hat{y}_t) - \log q_t(\hat{y}_t).$$

It is an unbiased estimate of the full-vocabulary reverse KL divergence and requires only the sampled token's log-probability from the teacher, but it discards the dense information over the remaining vocabulary and suffers from high variance.

Unnormalized top-k OPD. Unnormalized top-k OPD restricts the divergence computation to the teacher's top-k tokens $S_t^k = \text{TopK}(q_t, k)$ while keeping the original probabilities:

$$\ell_t^{\text{unnorm}} = \sum_{v \in S_t^k} p_t(v) \log \frac{p_t(v)}{q_t(v)}.$$

It provides dense supervision over S_t^k, but is not a well-defined divergence.

Table 3: **Performance of normalized top-k OPD under the student and teacher top-k supports.** The **Avg.** column represents the macro-average across all six math reasoning benchmarks. Best results are shown in **bold**; ties are bolded for both supports.

Method	MATH500	Minerva	Olympiad	AMC	AIME 24	AIME 25	Avg.
Student: ***Qwen3-1.7B-Base*** Teacher: ***Qwen3-8B***							
Student top-k	68.58	25.32	33.35	**36.00**	**9.17**	6.25	29.78
Teacher top-k	**69.93**	**25.46**	**33.41**	**36.00**	**9.17**	**7.08**	**30.18**

Normalized top-k OPD. Normalized top-k OPD (Li et al., 2026; Fu et al., 2026) instead renormalizes both distributions on S_t^k and minimizes the resulting subset reverse KL divergence:

$$\ell_t^{\text{norm}} = D_{\text{KL}}\big(\bar{p}_t^{(S_t^k)} \,\|\, \bar{q}_t^{(S_t^k)}\big) = \sum_{v \in S_t^k} \bar{p}_t^{(S_t^k)}(v) \log \frac{\bar{p}_t^{(S_t^k)}(v)}{\bar{q}_t^{(S_t^k)}(v)}.$$

It aligns the student's relative shape with the teacher's over S_t^k, but the normalization discards the tail probability, which, as we show in Section 3.1, steadily increases the student's tail probability and entropy.

MOPD top-k distillation. Multi-teacher on-policy distillation (MOPD) top-k distillation (Ma et al., 2026) uses the following loss function:

$$\ell_t^{\text{MOPD}} = \sum_{v \in S_t^k} \left[p_t(v) \log \frac{p_t(v)}{q_t(v)} - p_t(v) + q_t(v) \right].$$

Unlike ℓ_t^{unnorm}, whose minimizer deflates every top-k probability by a factor of e and collapses training (Appendix F.3), ℓ_t^{MOPD} is minimized exactly at $p_t(v) = q_t(v)$ for every $v \in S_t^k$, so its oracle solution does not diverge.

E Implementation Details

E.1 Experimental Details

We implement all methods using the VERL framework (Sheng et al., 2025) and conduct experiments with 6 NVIDIA RTX Pro 6000 GPUs, 8 NVIDIA A100 80 GB GPUs, or 32 NVIDIA A100 40 GB GPUs. Unless otherwise specified, all experiments use the default settings and hyperparameters listed in Table 4. For evaluation, we adopt the same prompt as Yan et al. (2026), which is shown in Prompt E.1. For experiments using Llama-3.1-8B as the student, we adopt the tokenizer and chat template of DeepSeek-R1-Distill-Llama-8B and synchronize the corresponding vocabulary and special-token configurations.

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

Table 4: **Default training and evaluation settings.**

Category	Item	Value
Training	Training temperature	1.0
	Global batch size	72
	Mini batch size	36
	Rollout number	4
	k (Number of Top-k Tokens)	16
	Top-p	1.0
	Max prompt length	1024
	Max response length	7168
	Learning rate	1e-6
	Training steps	300
	Loss aggregation	token-mean
	Optimizer	AdamW
Evaluation	Temperature	0.7
	Top-p	0.95
	Max new tokens	8192

E.2 Numerically Stable Computation of TA-OPD

The per-token TA-OPD loss in Eq. (4) depends on the tail log-probabilities $\log p_t^{\text{tail}}$ and $\log q_t^{\text{tail}}$. Below, we describe a failure case of a naive implementation and how we implement the loss.

Numerical issue of a naive implementation. The inference engine vLLM only provides access to the teacher's top-k log-probabilities $\log q_t(v)$, not the probabilities themselves. A naive implementation of TA-OPD loss is to exponentiate these log-probabilities back to probability space, sum them to obtain the total probability of the top-k tokens $1 - q_t^{\text{tail}} = \sum_{v \in S_t^k} q_t(v)$, and assign the remaining tail probability q_t^{tail} to v_{tail}. However, this is unstable. When the teacher's tail probability is close to zero, numerical error can make the sum $\sum_{v \in S_t^k} q_t(v)$ exceed one. The tail probability q_t^{tail} then becomes negative, rendering $\log q_t^{\text{tail}}$ ill-defined ($-\infty$ or NaN) and producing NaN gradients.

Our log-space implementation. To address the numerical issue, our implementation avoids forming p_t^{tail} and q_t^{tail} in probability space and carries out the whole computation in log space. We provide the pseudocode of our implementation in Listing 1. In particular, we first obtain the log total probability of the top-k tokens directly from the top-k log-probabilities via a log-sum-exp,

$$\log\big(1 - p_t^{\text{tail}}\big) = \text{logsumexp}_{v \in S_t^k} \log p_t(v), \qquad \log\big(1 - q_t^{\text{tail}}\big) = \text{logsumexp}_{v \in S_t^k} \log q_t(v),$$

which is numerically stable and never overflows. To further rule out the boundary case $p_t^{\text{tail}} \to 0$, we clamp the log total probability such that $\log(1 - p_t^{\text{tail}}) \le -\epsilon$, ensuring a strictly positive tail probability and a finite log-domain computation. The tail log-probability term $\log p_t^{\text{tail}}$ is then computed using the `log1mexp` primitive, which directly evaluates $\log(1 - \exp(a))$ in log space:

$$\log p_t^{\text{tail}} = \text{log1mexp}\big(\log(1 - p_t^{\text{tail}})\big).$$

The teacher term $\log q_t^{\text{tail}}$ is computed analogously. In this way, we avoid the numerical issues of the naive implementation.

E.3 Sample-Corrected TA-OPD with Old Log-Probabilities

Recall that the loss function of sample-corrected TA-OPD is given by

$$\ell_t^{\text{SC-TA}} = \sum_{v \in S_t^+} p_t(v)\, \text{sg}\left(\log \frac{p_t(v)}{q_t(v)}\right) + \mathbf{1}[\hat{y}_t \notin S_t^k]\, \frac{p_t(\hat{y}_t)}{\text{sg}(p_t(\hat{y}_t))}\, \text{sg}\left(\log \frac{p_t(\hat{y}_t)/p_t^{\text{tail}}}{q_t(\hat{y}_t)/q_t^{\text{tail}}}\right). \tag{22}$$

In our implementation, the global batch size (72) is larger than the mini-batch size (36). Thus, in each rollout batch, trajectories sampled from $\pi_{\theta_{\text{old}}}$ are reused for two gradient updates as θ evolves,

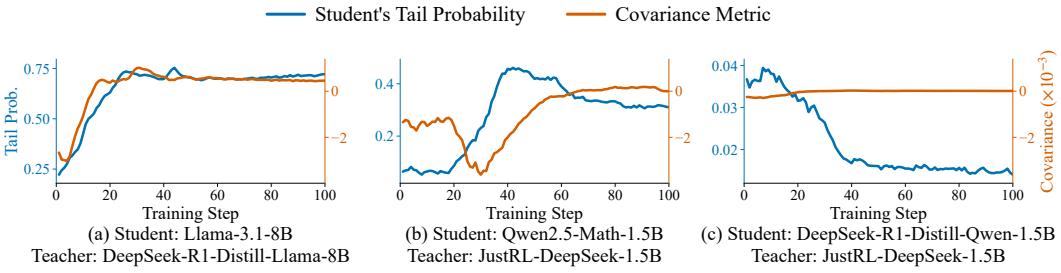
Listing 1: Pseudocode for the TA-OPD loss implementation.

```
def compute_taopd_loss(student_topk_log_probs, teacher_topk_log_probs, eps):
    # Input:
    #   student_topk_log_probs: student log-probs on the teacher's top-k tokens
    #   teacher_topk_log_probs: teacher top-k log-probs, log q(v)
    #   eps:                    small constant to keep the tail prob positive
    # Return:
    #   loss: per-token TA-OPD reverse KL divergence on the augmented token set

    # 1. log total probability of the top-k tokens via logsumexp
    student_log_topk_prob = logsumexp(student_topk_log_probs, dim=-1)   # log(1 - p_tail
    )
    teacher_log_topk_prob = logsumexp(teacher_topk_log_probs, dim=-1)   # log(1 - q_tail
    )

    # 2. tail-token log-prob log(p_tail) via log1mexp
    student_tail_log_prob = log1mexp(clamp_max(student_log_topk_prob, -eps))
    teacher_tail_log_prob = log1mexp(clamp_max(teacher_log_topk_prob, -eps))

    # 3. append the tail token and compute reverse KL on the augmented token set
    student_log_p = cat([student_topk_log_probs, student_tail_log_prob], dim=-1)
    teacher_log_q = cat([teacher_topk_log_probs, teacher_tail_log_prob], dim=-1)
    taopd_loss = sum(exp(student_log_p) * (student_log_p - teacher_log_q), dim=-1)
    return taopd_loss
```



(a) Student: Llama-3.1-8B Teacher: DeepSeek-R1-Distill-Llama-8B
(b) Student: Qwen2.5-Math-1.5B Teacher: JustRL-DeepSeek-1.5B
(c) Student: DeepSeek-R1-Distill-Qwen-1.5B Teacher: JustRL-DeepSeek-1.5B

Figure 12: **The student's tail probability and the covariance metric C_t across training steps.**

so the sampled token $\hat{y}_t$ is drawn from $\pi_{\theta_{\text{old}}}$ rather than from the current student distribution π_θ used to compute the loss. Since $\hat{y}_t$ is sampled from $\pi_{\theta_{\text{old}}}$, using $\text{sg}(p_t(\hat{y}_t))$ as the stop-gradient denominator in the correction term of Eq. (22) is no longer unbiased; instead, unbiasedness requires the denominator to be the old-policy probability $\text{sg}(\pi_{\theta_{\text{old}}}(\hat{y}_t))$:

$$\ell_t^{\text{SC-TA}} = \sum_{v \in S_t^+} p_t(v)\,\text{sg}\left(\log \frac{p_t(v)}{q_t(v)}\right) + \mathbf{1}[\hat{y}_t \notin S_t^k]\,\frac{p_t(\hat{y}_t)}{\text{sg}(\pi_{\theta_{\text{old}}}(\hat{y}_t))}\,\text{sg}\left(\log \frac{p_t(\hat{y}_t)/p_t^{\text{tail}}}{q_t(\hat{y}_t)/q_t^{\text{tail}}}\right). \quad (23)$$

F Additional Experiments

F.1 Empirical Validation of Proposition 2

Proposition 2 states that one gradient step of the normalized objective changes the student's tail probability by $\Delta p_t^{\text{tail}} = -\eta\, C_t + O(\eta^2)$, where

$$C_t := p_t^{\text{tail}}\big(1 - p_t^{\text{tail}}\big)\Big[\operatorname*{Cov}_{v\sim\bar{p}_t}\big(\bar{p}_t(v), \log \bar{q}_t(v)\big) - \operatorname*{Cov}_{v\sim\bar{p}_t}\big(\bar{p}_t(v), \log \bar{p}_t(v)\big)\Big].$$

We plot C_t, averaged over all tokens in a training batch, together with the student's tail probability across training steps. The smaller C_t is, the faster the theory predicts the tail probability to grow.

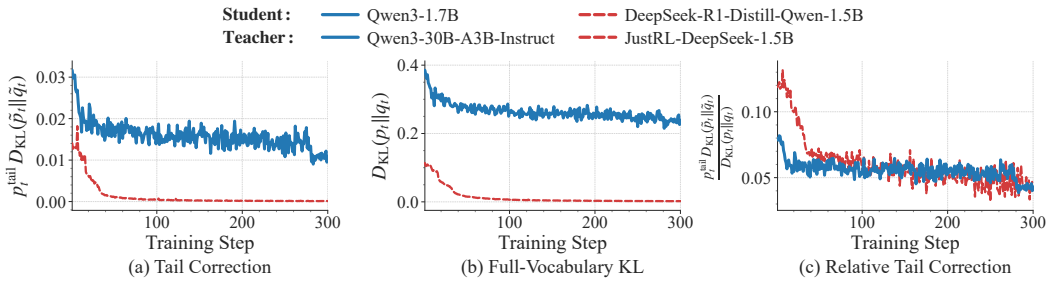


Figure 13: **The tail correction term, the full-vocabulary reverse KL, and their ratio across training steps for two student–teacher model pairs.**

In Figure 12, we plot C_t together with the student's tail probability for three student–teacher pairs. For Llama-3.1-8B (Figure 12a), C_t is most negative at the beginning of training and returns to zero within roughly 20 steps. Over the same window, the tail probability rises most steeply, from about 0.2 to above 0.7, and it stays flat once C_t vanishes. For Qwen2.5-Math-1.5B (Figure 12b), the growth is delayed: the tail probability stays around 0.06 during the first 20 steps, while C_t is only mildly negative. It then rises sharply to about 0.45 between steps 20 and 40, which coincides with C_t reaching its minimum of roughly -3.5×10^{-3} near step 30. Once C_t returns to zero after about step 60, the tail probability stops growing and settles around 0.3. For DeepSeek-R1-Distill-Qwen-1.5B (Figure 12c), C_t stays close to zero throughout training, and the tail probability shows no increase, remaining below 0.04. Across all three pairs, strongly negative values of C_t coincide with rapid growth of the tail probability, and near-zero values coincide with no growth. The covariance metric thus tracks both the direction and the magnitude of the tail probability change, empirically supporting Proposition 2.

F.2 Empirical Validation of the Tightness of TA-OPD

Proposition 4 shows that TA-OPD differs from full-vocabulary OPD only by the tail correction term $p_t^{\text{tail}} D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t)$. Here we measure how large this term is on actual training prefixes.

Setup. We train the student with TA-OPD and track three quantities along training: the tail correction term $p_t^{\text{tail}} D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t)$, the full-vocabulary reverse KL $D_{\text{KL}}(p_t \| q_t)$, and their ratio $p_t^{\text{tail}} D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t) / D_{\text{KL}}(p_t \| q_t)$, i.e., the fraction of the full-vocabulary reverse KL missed by TA-OPD. Computing the first two quantities exactly requires the teacher's log-probabilities over the full vocabulary, which is prohibitively expensive. We therefore report unbiased estimates built from the token $\hat{y}_t \sim p_t$. For the tail correction term, we use

$$\hat{r}_t := \mathbf{1}\left[\hat{y}_t \notin S_t^k\right] \log \frac{p_t(\hat{y}_t)/p_t^{\text{tail}}}{q_t(\hat{y}_t)/q_t^{\text{tail}}},$$

which satisfies $\mathbb{E}_{\hat{y}_t \sim p_t}[\hat{r}_t] = p_t^{\text{tail}} D_{\text{KL}}(\tilde{p}_t \| \tilde{q}_t)$. For the full-vocabulary reverse KL, we use $\ell_t^{\text{TA}} + \hat{r}_t$, where the TA-OPD loss ℓ_t^{TA} is computed exactly on the top-k set and the tail bucket; by Proposition 5, this is an unbiased estimate of $D_{\text{KL}}(p_t \| q_t)$. We consider two student–teacher pairs: Qwen3-1.7B with Qwen3-30B-A3B-Instruct-2507, and DeepSeek-R1-Distill-Qwen-1.5B with JustRL-DeepSeek-1.5B.

The tail correction term is small. As shown in Figure 13, the tail correction term is consistently more than an order of magnitude smaller than the full-vocabulary reverse KL divergence after the first few steps. For Qwen3-1.7B, the tail correction term drops from about 0.03 to 0.015 within the first 20 steps and further decreases to about 0.01 by the end of training, while the full-vocabulary reverse KL decreases slowly from about 0.38 to 0.25. As a result, the relative tail correction drops from about 8% to 6% early on and gradually decreases to about 5% afterwards. For DeepSeek-R1-Distill-Qwen-1.5B, both the tail correction term and the full-vocabulary reverse KL divergence rapidly approach zero within the first 50 steps. The relative tail correction peaks at about 12% during the first few steps, then quickly drops to about 6% and closely tracks that of Qwen3-1.7B for the rest

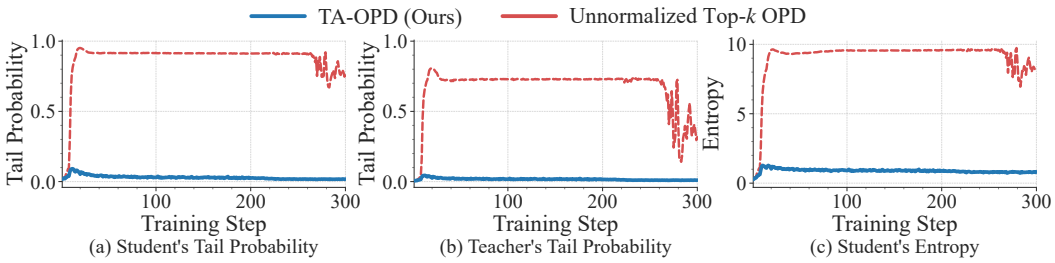


Figure 14: **The student's tail probability, the teacher's tail probability, and the student's token-level entropy across training steps.** Without the tail token, the student's tail probability rises above 0.9 and training collapses.

Table 5: **Avg@8 of unnormalized top-k OPD and TA-OPD.** The **Avg.** column represents the macro-average across all six benchmarks. Best results are shown in **bold**.

Methods	MATH500	Minerva	Olympiad	AMC	AIME 24	AIME 25	Avg.
Student: ***Qwen2.5-7B-Instruct*** Teacher: ***OpenThinker3-7B***							
Unnorm. top-k	3.35	1.70	1.42	3.00	0.42	0.42	1.72
TA-OPD (Ours)	**77.88**	**32.58**	**42.41**	**45.03**	**16.67**	**17.50**	**38.68**

of training, settling around 5%. Overall, TA-OPD recovers about 95% of the full-vocabulary reverse KL divergence after the early stage of training for both student–teacher pairs, confirming that it is a tight lower bound in practice.

F.3 Ablation study on the Tail Token

Our analysis attributes the tail probability increase issue to top-k normalization. A natural question is therefore whether simply removing the normalization is already sufficient, making the tail token in TA-OPD unnecessary. We show that it is not: without the tail token v_{tail}, training will diverge.

Setup. We ablate the tail token by comparing TA-OPD with unnormalized top-k OPD, which removes the tail term in TA-OPD's loss function. Specifically, its loss function is given by:

$$\ell_t^{\text{unnorm}} = \sum_{v \in S_t^k} p_t(v) \log \frac{p_t(v)}{q_t(v)} = \ell_t^{\text{TA}} - p_t^{\text{tail}} \log \frac{p_t^{\text{tail}}}{q_t^{\text{tail}}}.$$

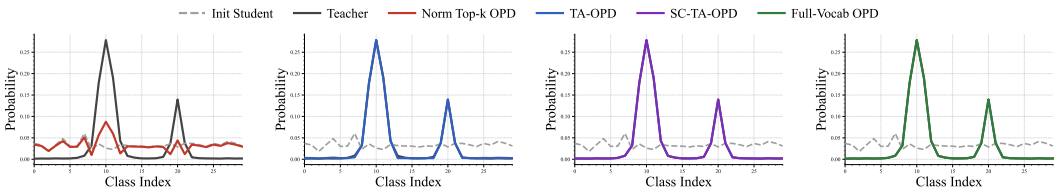
The student and teacher are Qwen2.5-7B-Instruct and OpenThinker3-7B, with all other settings following Table 4.

Removing the tail token collapses training. Figure 14 compares the training dynamics of the two objectives. Under unnormalized top-k OPD, the student's tail probability rises above 0.9 within 50 steps and its entropy grows monotonically, while the teacher's tail probability increases in tandem. The collapse is reflected in downstream accuracy (Table 5): its Avg@8 on MATH500 is only 3.35%, whereas TA-OPD reaches 77.88%.

The unnormalized objective is not a divergence. The failure is not an optimization artifact but a property of the objective itself: the objective is not minimized at $p_t = q_t$. Its minimizer admits the closed form

$$p_t^\star(v) = q_t(v)/e \quad \forall v \in S_t^k, \qquad \text{hence} \qquad p_t^{\text{tail},\star} = 1 - \frac{1 - q_t^{\text{tail}}}{e} \;\geq\; 1 - e^{-1} \approx 0.63.$$

That is, the objective is minimized by deflating every top-k probability by a factor of e and assigning the removed mass to the tail, which drives the loss negative. The optimum therefore assigns at least



(a) Normalized Top-k OPD (b) TA-OPD (c) SC-TA-OPD (d) Full-Vocabulary OPD

Figure 15: **Student policies under different OPD objectives.** Normalized top-k OPD (red) only matches the relative shape of the teacher distribution over the top-k tokens, but fails to align the tail probability. In contrast, TA-OPD (blue) and sample-corrected TA-OPD (purple) closely recover the bimodal teacher distribution.

63% of the probability mass outside the teacher's top-k tokens, regardless of the teacher distribution. Adding the tail token contributes exactly the missing term $p_t^{\text{tail}} \log(p_t^{\text{tail}}/q_t^{\text{tail}})$, which turns the objective into a reverse KL divergence on S_t^+: it is non-negative and uniquely minimized when the student matches the teacher on both the top-k tokens and the tail.

Overall, introducing the tail token is necessary for TA-OPD.

F.4 Synthetic Experiment

To provide an intuitive understanding of different OPD objectives, we construct a synthetic experiment based on a 30-armed bandit. Both the student and teacher policies are parameterized by vectors in $\mathbb{R}^{30}$, and the corresponding probability distributions are obtained by applying the softmax function to these vectors. The teacher policy is kept fixed during training and is defined over the discrete class space $\mathcal{V} = \{1, \ldots, 30\}$ as the following bimodal distribution:

$$q(v) \propto \exp\left(-\frac{(v-10)^2}{2 \cdot 2^2}\right) + 0.88 \cdot \exp\left(-\frac{(v-20)^2}{2 \cdot 2^2}\right), \quad v \in \mathcal{V}.$$

The student policy is parameterized as a categorical distribution over $\mathcal{V}$:

$$p_\theta(v) = \text{softmax}(z_\theta)_v, \quad z_\theta \in \mathbb{R}^{30},$$

where z_θ is initialized from $\mathcal{N}(0, 0.01^2)$. The student is optimized for 20000 steps using AdamW with a learning rate 1×10^{-3}. We compare four objectives: full-vocabulary OPD, normalized top-k OPD, TA-OPD, and sample-corrected TA-OPD. For top-k OPD, k is set to 8.

TA-OPD matches both the shape and tail probability of the teacher policy. Figure 15 compares the final student policies trained with different OPD objectives. TA-OPD closely recovers the bimodal teacher distribution. While normalized top-k OPD matches the relative shape over the teacher's top-k tokens, it assigns a substantially larger tail probability than the teacher. Overall, TA-OPD preserves the missing tail probability information and therefore provides a much closer approximation to the teacher policy than normalized top-k OPD.

TA-OPD better approximates full-vocabulary OPD than normalized top-k OPD. Figure 16 further visualizes the optimization trajectories on the full-vocabulary reverse-KL landscape. In particular, we collect the logit trajectories of all methods together with the teacher logits, center them in logit space, and project them onto the first two principal components. We then evaluate the full-vocabulary reverse KL on this two-dimensional plane and overlay the trajectories of different objectives. The results show that full-vocabulary OPD and TA-OPD move toward the same low-loss region around the teacher policy, while normalized top-k OPD converges to a point that remains far from the teacher under the full-vocabulary KL. Overall, TA-OPD's objective provides a better top-k estimate of the full-vocabulary reverse KL than the normalized objective.

F.5 Top-k Overlap Ratio of Different OPD Methods

Li et al. (2026) find that the top-k overlap ratio between the student and the teacher predicts the success of OPD well. Here we compare TA-OPD and normalized top-k OPD under this metric.

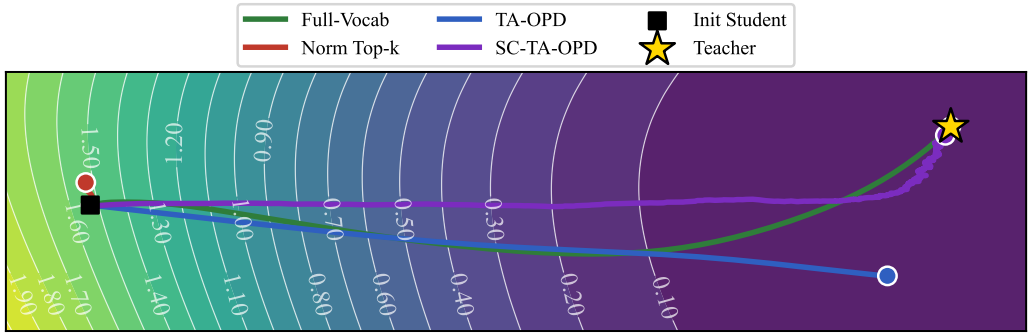
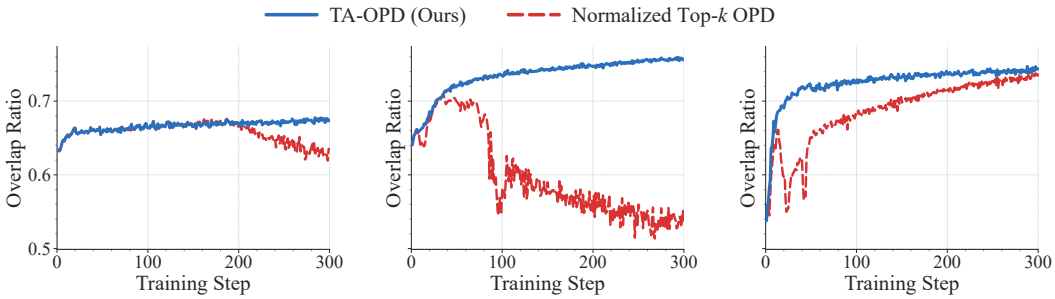


Figure 16: **Learning dynamics of different OPD objectives.** TA-OPD better approximates full-vocabulary OPD.



(a) Qwen3-1.7B from Qwen3-30B-A3B-Instruct

(b) Qwen2.5-7B-Instruct from OpenThinker3-7B

(c) Llama-3.1-8B from DS-R1-Distill-Llama-8B

Figure 17: **Top-k overlap ratio between the student and the teacher across training steps.** TA-OPD achieves higher overlap ratios than normalized top-k OPD across different model pairs.

Setup. On each student-generated prefix, the top-k overlap ratio is the fraction of the teacher's top-k tokens that also fall in the student's top-k tokens,

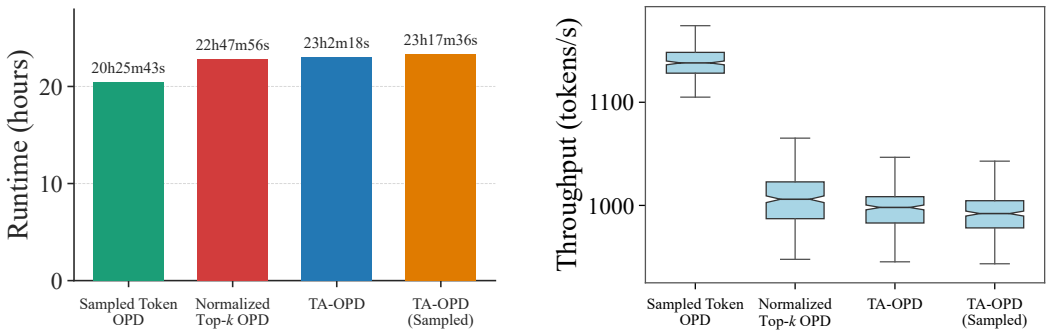
$$\mathrm{Overlap}_t^k = \frac{\left|\mathrm{TopK}(p_t, k) \cap \mathrm{TopK}(q_t, k)\right|}{k},$$

which we average over all tokens in a training batch. A higher ratio means the student and the teacher agree on which tokens are plausible, so the top-k objective supervises a token set that is meaningful to the student. We use the three student–teacher pairs of Section 3.2 and report the ratio across training steps.

TA-OPD attains a higher top-k overlap ratio. Figure 17 compares the overlap ratio of the two objectives. TA-OPD steadily increases the overlap ratio and keeps it stable throughout training. In contrast, normalized top-k OPD is consistently lower: for Qwen2.5-7B-Instruct the ratio peaks early and then collapses, and for Llama-3.1-8B it recovers only slowly and remains below TA-OPD for the entire run. Overall, TA-OPD keeps the student and the teacher aligned on the top-k tokens, which is consistent with its stronger downstream accuracy.

F.6 Computational Overhead of TA-OPD

TA-OPD introduces no additional computational overhead compared with normalized top-k OPD. Both methods require exactly the same teacher-side information: the log-probabilities of the teacher's top-k tokens. Given these quantities, TA-OPD only additionally computes the tail log-probabilities $\log p_t^{\mathrm{tail}}$ and $\log q_t^{\mathrm{tail}}$, which involve a single log-sum-exp over k values per token. This cost is negligible relative to the forward and backward passes of the student model. Sample-corrected TA-OPD additionally requires the teacher's log-probability of the sampled token $\hat{y}_t$. Although this appears to be an extra query, it incurs negligible additional cost in practice: when queried



(a) Total training runtime.

(b) Throughput distribution during training.

Figure 18: Computational overhead comparison between sampled-token OPD, normalized Top-k OPD, TA-OPD, and sample-corrected TA-OPD. Left: total training runtime. Right: throughput distribution across training steps. The student and teacher models are Qwen3-1.7B and Qwen3-30B-A3B-Instruct-2507, respectively.

for the top-k log-probabilities, inference engines such as vLLM (Kwon et al., 2023) always compute and return the sampled token's log-probability as well, even if it falls outside the top-k tokens. Therefore, the teacher-query cost of sample-corrected TA-OPD is identical to that of TA-OPD.

We empirically verify this by measuring the wall-clock cost of sampled-token OPD, normalized top-k OPD, TA-OPD, and sample-corrected TA-OPD under the same setup. We use Qwen3-1.7B as the student and Qwen3-30B-A3B-Instruct-2507 as the teacher, with all other settings following Table 4. As shown in Figure 18 (a), completing the full 300 training steps takes 23h2m18s for TA-OPD and 22h47m56s for normalized top-k OPD, i.e., a difference of 14m22s, or 1.05% of the total runtime. This 14-minute gap might stem from the slightly longer average response length under TA-OPD. Figure 18 (b) further compares the per-step throughput distributions, whose medians are around 1000 and 1005 tokens/s for TA-OPD and normalized top-k OPD, respectively. Overall, TA-OPD incurs negligible computational cost compared to normalized top-k OPD.

F.7 Discussion of EMA-PG

Exponential Moving Average-Policy Gradient (EMA-PG) (Zhang & Ba, 2026) proposes a top-k KL estimator that is related to but distinct from our sample-corrected TA-OPD. Its per-token loss is:

$$\ell_t^{\mathrm{EMA}} = \underbrace{\sum_{v\in S_t^k} p_t(v)\,\mathrm{sg}\left(\log \frac{p_t(v)}{q_t(v)}\right)}_{=\ \ell_t^{\mathrm{unnorm}} \text{ in value}} + \mathbf{1}[\hat{y}_t \notin S_t^k]\,\frac{p_t(\hat{y}_t)}{\mathrm{sg}(p_t(\hat{y}_t))}\,\mathrm{sg}\left(\log \frac{p_t(\hat{y}_t)}{q_t(\hat{y}_t)}\right). \tag{24}$$

In contrast, sample-corrected TA-OPD's per-token loss is given by:

$$\ell_t^{\text{SC-TA}} = \underbrace{\sum_{v\in S_t^+} p_t(v)\,\mathrm{sg}\left(\log \frac{p_t(v)}{q_t(v)}\right)}_{=\ \ell_t^{\mathrm{TA}} \text{ in value}} + \mathbf{1}[\hat{y}_t \notin S_t^k]\,\frac{p_t(\hat{y}_t)}{\mathrm{sg}(p_t(\hat{y}_t))}\,\mathrm{sg}\left(\log \frac{p_t(\hat{y}_t)/p_t^{\mathrm{tail}}}{q_t(\hat{y}_t)/q_t^{\mathrm{tail}}}\right).$$

Both estimators are unbiased in value and in gradient for the full-vocabulary reverse KL divergence, but they differ in where the truncation is placed and in what the sampled token is used to estimate. EMA-PG uses the sampled token to estimate $\sum_{v\notin S_t^k} p_t(v)\log\frac{p_t(v)}{q_t(v)}$. In contrast, our estimator starts from ℓ_t^{TA} and uses the sampled token to estimate the residual $p_t^{\mathrm{tail}} D_{\mathrm{KL}}(\tilde{p}_t\|\tilde{q}_t)$ identified in Proposition 4. In other words, our estimator is obtained by debiasing TA-OPD, whereas EMA-PG debiases the unnormalized top-k objective. In the following, we show that our estimator provably achieves lower variance than EMA-PG under mild assumptions.

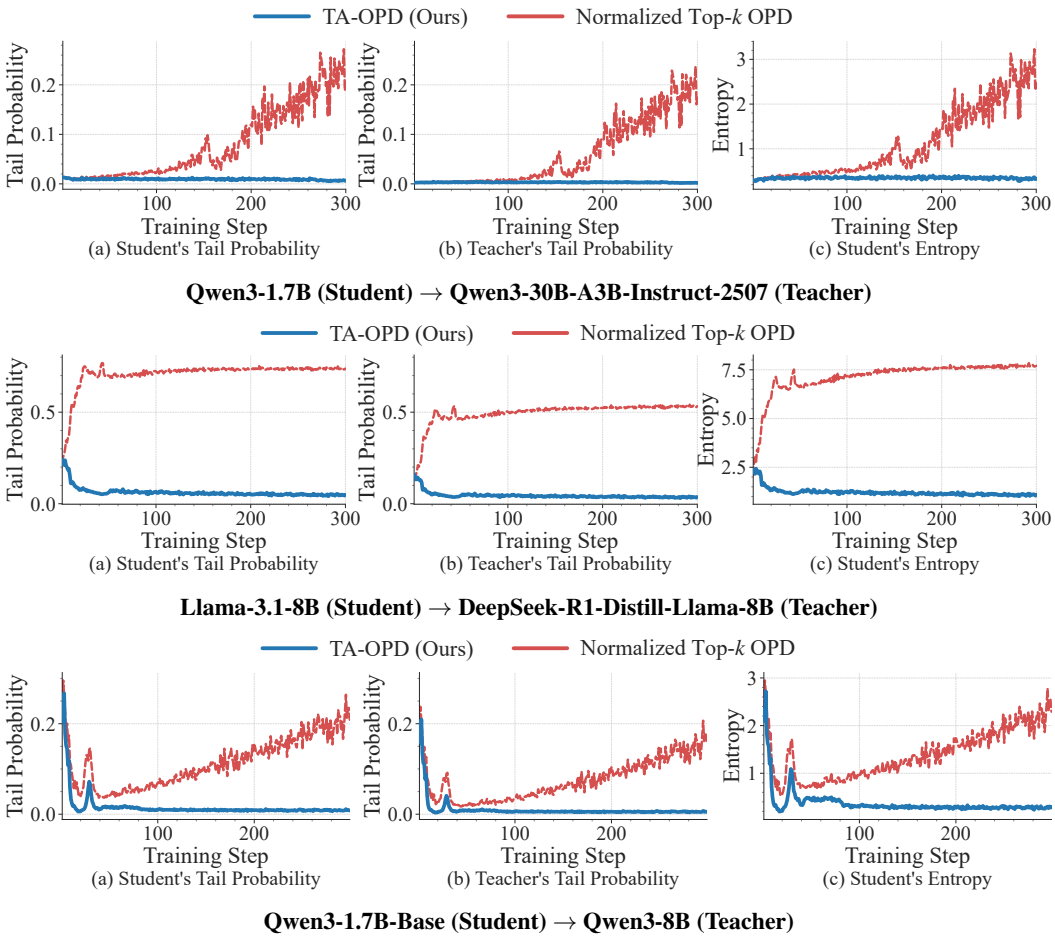


Figure 19: **Tail probability and token-level entropy over training for TA-OPD and Normalized top-k OPD.** Each row corresponds to a different student–teacher pair.

Variance analysis. Since both estimators are unbiased, a natural criterion for comparing them is the variance of the estimate. A direct computation gives

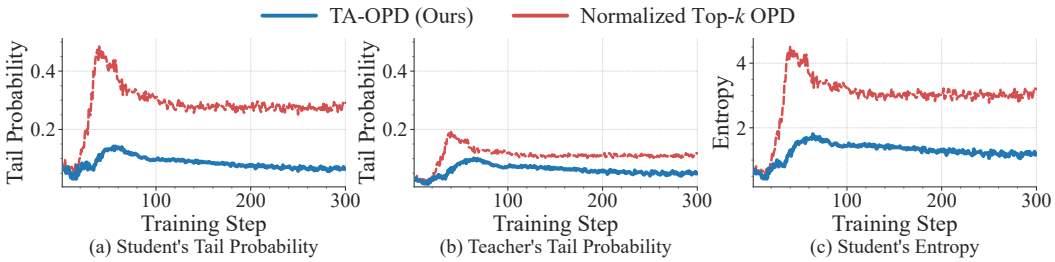
$$\mathrm{Var}\big(\ell_t^{\mathrm{EMA}}\big) - \mathrm{Var}\big(\ell_t^{\text{SC-TA}}\big) = p_t^{\mathrm{tail}}\big(1 - p_t^{\mathrm{tail}}\big)\,\log\frac{p_t^{\mathrm{tail}}}{q_t^{\mathrm{tail}}}\left(\log\frac{p_t^{\mathrm{tail}}}{q_t^{\mathrm{tail}}} + 2\,D_{\mathrm{KL}}(\tilde{p}_t\|\tilde{q}_t)\right). \tag{25}$$

Since $D_{\mathrm{KL}}(\tilde{p}_t\|\tilde{q}_t) \geq 0$, the gap is strictly positive whenever $p_t^{\mathrm{tail}} > q_t^{\mathrm{tail}}$, i.e., whenever the student's tail probability exceeds the teacher's, which is likely to happen when using the teacher's top-k support. Figures 4 and 19 confirm this empirically, where q_t^{tail} is below p_t^{tail} throughout training across all model pairs. Overall, our loss objective tends to have lower variance than EMA-PG.

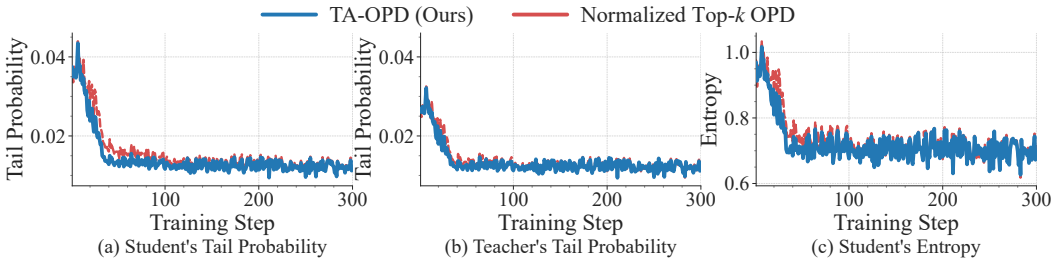
G Detailed Results

G.1 Training Dynamics of Additional Model Pairs

Figure 19 presents additional comparisons between TA-OPD and normalized top-k OPD in terms of tail probability and token-level entropy during training. Results are shown on three student–teacher pairs: Qwen3-1.7B paired with Qwen3-30B-A3B-Instruct-2507, Llama-3.1-8B paired with DeepSeek-R1-Distill-Llama-8B, and Qwen3-1.7B-Base paired with Qwen3-8B.



Qwen2.5-Math-1.5B (Student) → JustRL-DeepSeek-1.5B (Teacher)



DeepSeek-R1-Distill-Qwen-1.5B (Student) → JustRL-DeepSeek-1.5B (Teacher)

Figure 20: **Tail probability and token-level entropy over training for TA-OPD and Normalized top-k OPD under different student–teacher capability gaps.** The tail probability increase occurs only for Qwen2.5-Math-1.5B, not for DeepSeek-R1-Distill-Qwen-1.5B.