OPD+: Rethinking the Advantage Design for On-Policy Distillation

Anonymous ACL submission

Abstract

On-policy distillation (OPD) is a widely used technique to transfer capabilities from capable teacher language models to the base student models, and can be formulated in a reinforcement learning style objective using student generated rollouts. Yet, despite the divergence reward being dependent on student model likelihood, existing works usually adopt a stop gradient design primarily for stability, which makes the resulting advantage estimation questionable. In this work, we provide a generic optimization framework based on f-divergence between the student and teacher, and mathematically revisit whether such design space is valid. We prove that general stop-gradient operation would lead to biased estimates of the reward objective and corresponding gradient for general divergence functions. We propose OPD+, the corrected version of OPD that demonstrates improved performance over the baseline KL approach and also supports the choice of various f-divergence. We validate our findings on mathematical reasoning and tool-use benchmarks.

1 Introduction

On-policy distillation (OPD) (Agarwal et al., 2024; Lu and Lab, 2025) is a powerful technique to let student models learn from capable teacher models through dense token-level supervisions. It has also become a crucial part of the post training recipes of modern large language models (LLMs), e.g. in Deepseek V4 (DeepSeek-AI, 2026), to learn from teacher models without performance degeneracy.

In the design space of OPD[1], existing baselines like Tinker (Lu and Lab, 2025) suggest to adopt the *KL divergence* between student and teacher to achieve the best performance. To derive the loss function, common practice also *disables the gradient propagation* of log ratio difference reward between student and teacher primarily for numerical stability, motivated by e.g., target networks in DQN (Mnih et al., 2013).

However, in this paper, we revisit these two key designs for OPD. Our contributions are mainly two-folded: **Firstly,** we provide the framework of f-divergence on policy distillation and derive the corresponding gradient and loss functions dependent on whether gradients on the reward function is enabled. Our framework enriches the design space of on policy distillation with various f-divergence; **Secondly,** we demonstrate that stopping gradient in reward not only yields mathematically incorrectly gradient estimates but also harms the performance for various f-divergence functions. The corrected algorithm, which we name as OPD+, yields strong empirical gains over the baseline in math reasoning and tool use scenarios, and challenges the common belief of the advantage of KL divergence over other statistical distances in on policy distillation.

2 Preliminaries

We first revisit the preliminaries of LLMs and OPD as a Reinforcement Learning (RL) objective. More discussions on some other relevant works are provided in Appendix A.

LLMs. Denote by $y \sim p_\theta(\cdot \mid x)$ a full sentence generated by language model p_θ given the context/prompt x. For auto-regressive LLMs, we use p_θ for a sequential model parameterized by θ, whose likelihood can be factorized as the product of the next-token distribution over the vocabulary, conditioning on the prompt x and already generated tokens $y_{<n}$, i.e., $p_\theta(y \mid x) = \prod_{n=1}^{L_y} p_\theta(y_n \mid y_{<n}, x)$.

OPD. For OPD, we set the behavior policy the same as the student policy/model, and we are in-

[1]For OPD implementation, existing approaches either adopt a logit (top-k) distillation or the single token likelihood. We focus on the simpler second approach without the need of additional infrastructure modification as in DeepSeek-AI (2026).

terested in optimizing a per-token level difference between student p_θ and teacher policy q:

$$\mathbb{E}_{x\sim X,\, y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} D(p_\theta(\cdot \mid y_{<n}, x), q(\cdot \mid y_{<n}, x)),$$

where $D(p, q)$ denotes a chosen statistical divergence between two probability distributions, and L_y is the length of the generation y. When choosing D as the KL divergence, i.e., $D(P, Q) := D_{\mathrm{KL}}(P\|Q) = \sum_x p(x)(\log p(x) - \log q(x))$, we have an equivalent loss objective which can be simply estimated by student sample rollouts:

$$\mathbb{E}_{x\sim X,\, y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \log \frac{p_\theta(y_n \mid y_{<n}, x)}{q(y_n \mid y_{<n}, x)}.$$

In practice, the log ratio difference is often 'stop grad'ed as a *fixed* reward. Using a zero discount factor for the advantage function yields the PPO-like loss $\mathcal{J}_{\mathrm{opd}}(\theta) :=$

$$\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \rho_\theta(y_n \mid y_{<n}, x) \log \frac{p_{\theta_0}(y_n \mid y_{<n}, x)}{q(y_n \mid y_{<n}, x)},$$

where $\rho_\theta = p_\theta(y_n \mid y_{<n}, x)/p_{\theta_0}(y_n \mid y_{<n}, x)$ is the importance sampling ratio. We provide a more detailed full derivation in Appendix B.

3 OPD+: Correcting the Advantage Estimation for On Policy Distillaion

Note that in our derivation of OPD loss objective for KL divergence, it is assumed that the reward $\log \frac{p_\theta(y_n|y_{<n},x)}{q(y_n|y_{<n},x)}$ is fixed, and hence independent of θ. Yet this leads to a potential mismatch between the true target and our loss objective: the gradient can be biased due to the stop gradient operations.

3.1 Beyond KL: f-divergence optimization for on-policy distillation

We first present a generic framework of f-divergence optimization between the student and teacher policy, and derive the true gradient and the loss objective compared to those obtained from stop gradient operations.

f-divergence objective. Let $f : (0, \infty) \to \mathbb{R}$ be a convex function satisfying $f(1) = 0$, and define the token-level density ratio as $u_{\theta,n} := q(y_n \,|\, y_{<n}, x)/p_\theta(y_n \,|\, y_{<n}, x)$. Consider the f-divergence objective sampling from the student model:

$$J^f(\theta) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} f(u_{\theta,n})\,. \quad (1)$$

Equivalently, minimizing $J^f(\theta)$ corresponds to minimizing the f-divergence between the student's and the teacher's next-token distribution along prefixes sampled from the student.

Stop-gradient policy-gradient estimator. Define the reward as the negative divergence value:

$$r^f_\theta(y_{\le n}, x) := -f(u_{\theta,n})\,, \quad (2)$$

As in the earlier reverse-KL case, suppose we stop-gradient the reward term (denote θ_- for preventing gradient back propagation), the objective is:

$$J^f_{\mathrm{sg}}(\theta) = \mathbb{E}_{x\sim X,\, y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \gamma^n r^f_{\theta_-}(y_{\le n}, x).$$

By REINFORCE (Williams, 1992), $\nabla_\theta J^f_{\mathrm{sg}}(\theta) =$

$$\mathbb{E}_{x\sim X,\, y\sim p_\theta(\cdot|x)} \left[\sum_{k=1}^{L_y} \nabla_\theta \log p_\theta(y_k \mid y_{<k}, x) A^f_k \right],$$

where the reward-to-go advantage is $A^f_k := \sum_{n=k}^{L_y} \gamma^{n-k} r^f_{\theta_-}(y_{\le n}, x)$. Taking the discount factor $\gamma = 0$ yields the immediate token-level reward $A^f_k = r^f_{\theta_-}(y_{\le k}, x)$. Thus, the resulting gradient estimator of J^f_{sg} is:

$$\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x) r^f_{\theta_-}(y_{\le n}, x). \quad (3)$$

3.2 OPD+: Corrected Formula for f-divergence Optimization.

We derive our earlier formula by applying stop gradient to the reward term. However, the true reward $r^f_\theta(y_{\le n}, x) = -f(u_{\theta,n})$ does depend on θ. So if we differentiate the true objective, we get two terms – the *score-function term* and the *direct reward-gradient term*: $\nabla_\theta \mathbb{E}_{y_n\sim p_\theta(\cdot|y_{<n},x)} \left[r^f_\theta(y_{\le n}, x) \right] =$

$$\mathbb{E}[r^f_\theta(y_{\le n}, x) \nabla_\theta \log p_\theta(y_n|y_{<n}, x) + \nabla_\theta r^f_\theta(y_{\le n}, x)].$$

The direct reward-gradient term is computed as:

$$\nabla_\theta r^f_\theta(y_{\le n}, x) = u_{\theta,n} f'(u_{\theta,n}) \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x).$$

So the correct immediate token-level gradient is:

$$\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x) w_f(u_{\theta,n}),$$

where $w_f(u_{\theta,n}) = -f(u_{\theta,n}) + u_{\theta,n} f'(u_{\theta,n})$. Thus, for a general f-divergence, the correct advantage function is not simply $r_\theta^f(y_{\leq n}, x) = -f(u_{\theta,n})$: there is an additional term of $u_{\theta,n} f'(u_{\theta,n})$ due to the reward dependency on policy parameter θ, and we name our method as OPD+. To implement OPD+, only one line change of code is needed.

What is special about Reverse KL? So is the exisiting formula of reverse KL misused when we do not apply stop grad? Theoretically, the answer is **no**. Reverse KL is a special instance of f-divergence with $f(u) = -\log u$. As a result,

$$w_f(u) = f(u) - uf'(u) = -\log u + 1.$$

The additional $+1$ term is a constant baseline, and hence vanishes under the score identity $\mathbb{E}_{y_n\sim p_\theta}\left[\nabla_\theta \log p_\theta(y_n \mid y_{<n}, x)\right] = 0$. Thus, the direct reward-gradient correction for reverse KL reduces to a constant baseline, and is *hidden* in the stop-gradient policy optimization. For a general f, the term $uf'(u)$ is not a constant, so the stop gradient operation leads to a bias. See Table 1 for comparisons of different divergences.

Divergence	Stop Grad Adv.	Corrected Adv.
Forward KL	$-u \ln u$	u
Reverse KL	$\ln u$	$\ln u - 1$
JSD	$-\frac{1}{2}\left[u \ln u - (1+u) \ln \frac{1+u}{2}\right]$	$\frac{1}{2} \ln \frac{1+u}{2}$

Table 1: Advantage Design for On-Policy Distillation.

In addition, despite the theoretical coincidence, we observe the empirical performance gain in adding the baseline term even for KL divergence.

4 Experiments

We conduct experiments to showcase the advantage of our OPD+. We test our method across two distinct training paradigms spanning ***competition mathematics*** and ***agentic tool use***, and use either a separate, stronger model (OPD) or the model itself with privileged ground-truth solutions as the teacher (OPSD, (Zhao et al., 2026)).

Models, Tasks and Metrics. We implement OPD and OPD+ across multiple scales of the QWEN3 model family (QWEN3-1.7B and 8B). We benchmark our approach on two demanding application domains, evaluating performance via tasks that assess distinct capability dimensions. For ***competition mathematics***, we distill on DeepMath rollouts and evaluate on high-bar reasoning benchmarks, specifically AIME 24, AIME 25, and HMMT 25. For ***tool use***, we distill on the ToolUse task and evaluate on its 68-problem test split, where a teacher reference score is also available for baseline comparison. For both tasks, the evaluation metric is `avg@32`, which is the mean per-sample score as an estimate of single-sample accuracy. More details could be found in Appendix C.

Main Results

We report in Table 2, the best-checkpoint performance achieved by each method across the evaluated tasks. Our results present a striking challenge to the prevailing post-training paradigm, which universally favors reverse KL due to its empirical stability while avoiding forward KL and JSD.

The Salvation of Forward KL and JSD. Under the standard bare-cost advantage $-f(u)$, both Forward KL and JSD exhibit catastrophic collapse, failing to match even the SFT-initialized baseline on competition mathematics, and dropping to 0.00% accuracy on tool use. In contrast, our analytic correction, OPD+, completely eliminates this divergence-dependent failure mode. By replacing $-f(u)$ with the gradient-faithful weight $w(u)$, OPD+ rescues Forward KL and JSD from collapse, elevating their performance to be fully competitive with or superior to the Reverse KL benchmark.

Surpassing the Established Paradigm. Crucially, the empirical gains of OPD+ extend beyond merely stabilizing previously un-trainable objectives; it fundamentally expands the performance ceiling of on-policy distillation:

$\diamondsuit$ **The Superiority of JSD$^+$**: While JSD was previously considered completely non-viable on-policy, JSD$^+$ not only matches the conventional Reverse KL paradigm on mathematical averages but substantially outperforms it on the challenging AIME 25 benchmark (50.42% vs. 48.54%). More importantly, on the Tool Use task, JSD$^+$ establishes a new performance peak of 60.29%, surpassing the standard Reverse KL baseline (59.97%).

$\diamondsuit$ **Bootstrapping Reverse KL Itself**: Even within the native domain of Reverse KL, injecting our derived formulation yields strict empirical improvements. Compared to the standard Reverse

Div. Type	Advantage	Competition Math				Agentic	Self-Distillation
		AIME 24	AIME 25	HMMT 25	Avg.	Tool Use	AIME 24
SFT-init student		50.94	38.61	25.52	38.36	24.63	48.89
Forward KL	OPD: $-f(u)$	50.94	38.61	25.52	38.36	24.63	48.89
	OPD+: $w(u)$	59.79	46.88	32.50	46.39	59.01	<u>50.83</u>
JSD	OPD: $-f(u)$	50.94	38.61	25.52	38.36	24.63	49.17
	OPD+: $w(u)$	63.12	**50.42**	33.02	48.85	<u>60.29</u>	49.17
Reverse KL	OPD: $-f(u)$	<u>64.27</u>	48.54	**34.06**	<u>48.96</u>	59.97	50.00
	OPD+: $w(u)$	**64.69**	<u>48.85</u>	<u>33.44</u>	**48.99**	**61.44**	**53.33**

Table 2: Best-checkpoint avg@32 accuracy (%, ↑) across math, tool-use, and self-distillation tasks. We compare standard OPD using $-f(u)$ with the corrected OPD+ using $w(u)$ across different f-divergences. Bold/underline mark the best/second-best results within each task. Grey cells indicate runs whose performance fell below the SFT-init student baseline; their displayed value is clipped to the baseline performance. OPD+ eliminates the training collapse seen in standard Forward KL and JSD. Self-Distillation corresponds to results of QWEN3-1.7B.

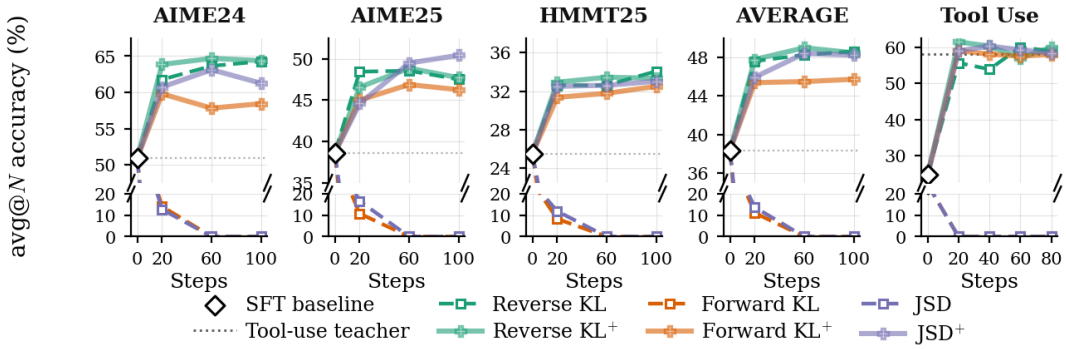

Figure 1: Training dynamics of OPD vs. OPD+ across different steps: we report average accuracy `avg@32` (↑) on three math benchmarks (AIME 24, AIME 25, HMMT 25), their average, and an agentic Tool Use task (rightmost panel). Solid lines with markers (+) represent our corrected OPD+ variants ($w(u)$), while dashed lines (□) denote standard OPD baselines ($-f(u)$). OPD+ consistently accelerates early convergence or prevents catastrophic training collapse.

KL baseline, Reverse KL$^+$ achieves consistent performance boosts across three out of the four evaluated benchmarks: AIME 24 (64.69% vs. 64.27%), AIME 25 (48.85% vs. 48.54%), and Tool Use (61.44% vs. 59.97%). These findings demonstrate that the historical underperformance of alternative f-divergences is entirely a side-effect of a misspecified gradient coefficient, rather than an intrinsic limitation of the divergences themselves. When optimized faithfully, they offer powerful alternatives that outcompete the current industry standard.

OPSD+ also outperforms OPSD baselines. We observe that the our method still demonstrate better performance compared to the baseline approaches in the OPSD setup when the teacher model is model itself with privileged information. As in the last row of Table 2, when using QWEN3 1.7B INSTRUCT as the base model, the improved versions all outperforms the vanilla stop gradient version, and the improved reverse KL showcases the best performance among different divergences.

5 Conclusions and Discussion

In this paper, we investigate on the two key parts among the design space of OPD: the divergence and the choice of reward/advantage. We provide OPD+, the generic framework upon f-divergence optimization and the corrected formulas compared to vanilla policy gradient based on the stop gradient operations. Our techniques provide both more mathematically correct and empirically well-performed algorithms, challenging the existing practice of divergence choice and design space of the reward. We believe that our paper could trigger further research on better design of advantage function of OPD type methods.

Limitations

In this paper, we did not do a detailed model scaling analysis to evaluate the sensitivity of our algorithms and baseline methods to the model scale. We also limit the analysis of our experiments to the Qwen3 model families, so the results beyond Qwen model families will be helpful to see the generalization capability of our findings. We also do not compare the performance of single token performance with logit distillation performance for various divergences, which we would pursue as future work to better understand the tradeoffs.

Ethical Considerations

Our work primarily targets at advancing the field of Machine Learning and NLP. There is no ethical considerations that we believe are worth mentioning.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. 2026. Flow density control: generative optimization beyond entropy-regularized fine-tuning. In *Advances in neural information processing systems*, volume 38, pages 11056–11088.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Zhengyi Guo, Jiayuan Sheng, David D Yao, and Wenpin Tang. 2026. Improved techniques for fine-tuning flow models via adjoint matching: a deterministic control pipeline. *arXiv preprint arXiv:2605.06583*.

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

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. 2025. Correlated proxies: a new definition and improved mitigation for reward hacking. In *International Conference on Learning Representations*, volume 2025, pages 88379–88421.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.

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

Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J Andrew Bagnell, Aarti Singh, and Andrea Zanette. 2026. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint arXiv:2602.02482*.

Wenpin Tang and Fuzhong Zhou. 2024. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. *arXiv preprint arXiv:2403.06279*.

Yunhao Tang and Rémi Munos. 2025. On a few pitfalls in kl divergence gradient estimation for rl. *arXiv preprint arXiv:2506.09477*.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. 2024. Beyond reverse KL: generalizing direct preference optimization with diverse divergence constraints. In *International Conference on Learning Representations*, volume 2024, pages 10450–10480.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. 2025. On the design of KL-regularized policy gradient algorithms for LLM reasoning. *arXiv preprint arXiv:2505.17508*.

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

A Related Works

f-divergence in post training generative models. Statistical divergence beyond KL has been studied in the context of RLHF as regularizers for post-training LLMs (Wang et al., 2024; Laidlaw et al., 2025; Zhang et al., 2025). Relevant extensions has also been studied for fine-tuning diffusion/flow models (Tang and Zhou, 2024; De Santi et al., 2026; Guo et al., 2026). The possible biasedness of directly differentiated KL gradient estimator has been discussed in (Tang and Munos, 2025; Shah et al., 2025). Comparatively in OPD, divergence choice acts as the primary reward, which is of more crucial importance for downstream tasks than for alignment/reasoning regularizors.

OPD. (Agarwal et al., 2024) proposed GKD for LLMs, which corresponds to a supervised learning form of OPD and also proposes using various f-divergence in the same off-policy setup. (Lu and Lab, 2025) discussed the RL like variant of OPD for KL divergence, including the application in LLM reasoning and personalizations. See (Song and Zheng, 2026) for a comprehensive survey on the followup works and recent progress. More recently, OPSD (Zhao et al., 2026) type of methods generalize the setups of OPD to using the models itself with priviledge information as the teacher model. Concurrent work like SDPO (Hübotter et al., 2026) and SDFT (Shenfeld et al., 2026) proposed similar ideas by letting the model condition on environment feedback. In the same spirit (Song et al., 2026) also discussed using the modified solutions from LLMs conditioned on text feedback to provide similar dense supervision.

B OPD Derivations

We provide a more detailed derivation of OPD, and comparison of RL like loss in our paper to a supervised learning which is used in practice among different code repositories.

RL Loss. To simplify the notation, we now define the per-token reward as the negative sampled reverse-KL term:

$$r_\theta(y_{\le n}, x) = \log q(y_n \mid y_{<n}, x) - \log p_\theta(y_n \mid y_{<n}, x).$$

Therefore, minimizing the reverse-KL objective

$$J_{\mathrm{KL-rev}}(\theta) = \mathbb{E}_{x\sim X, y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \log \frac{p_\theta(y_n \mid y_{<n}, x)}{q(y_n \mid y_{<n}, x)}$$

is equivalent to maximizing

$$J_{\mathrm{reward}}(\theta) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} r_\theta(y_{\le n}, x).$$

As a common practice, the reward term is usually stop-gradient'ed in the policy-gradient estimator. We write

$$r_{\theta_-}(y_{\le n}, x) := \log q(y_n \mid y_{<n}, x) - \log p_{\theta_-}(y_n \mid y_{<n}, x),$$

where θ_- denotes the stop-gradient version of θ. The corresponding RL-style objective is

$$J_{\mathrm{sg}}(\theta) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} r_{\theta_-}(y_{\le n}, x).$$

By the REINFORCE formula, its gradient is

$$\nabla_\theta J_{\mathrm{sg}}(\theta) = \mathbb{E}_{y\sim p_\theta(\cdot|x)} \left[\sum_{k=1}^{L_y} \nabla_\theta \log p_\theta(y_k \mid y_{<k}, x) A_k \right].$$

in which the advantage function is defined as: $A_k = \sum_{n=k}^{L_y} r_{\theta_-}(y_{\le n}, x)$. However, in token-level on-policy distillation, one commonly uses discount factor $\gamma = 0$, which is reported by (Lu and Lab, 2025) that there is no improved performance from $\gamma > 0$, despite that they are more mathematically correct. This thus replaces the full future reward-to-go by the immediate reward:

$$A_k = \sum_{n=k}^{L_y} \gamma^{n-k} r_{\theta_-}(y_{\le n}, x) \Longrightarrow A_k = r_{\theta_-}(y_{\le k}, x),$$

when $\gamma = 0$. Therefore the $\gamma = 0$ policy-gradient estimator becomes

$$\nabla_\theta J_{\mathrm{sg}}^{\gamma=0}(\theta) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x) r_{\theta_-}(y_{\le n}, x).$$

Equivalently, the corresponding minimization loss is

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_\theta(\cdot|x)} \sum_{n=1}^{L_y} r_{\theta_-}(y_{\le n}, x),$$

In practice, trajectories are usually sampled from a frozen behavior model p_{θ_0}. For each sampled token $y_n \sim p_{\theta_0}(\cdot \mid y_{<n}, x)$, define

$$r_{\theta_0}(y_{\le n}, x) = \log q(y_n \mid y_{<n}, x) - \log p_{\theta_0}(y_n \mid y_{<n}, x),$$

and the token-level importance ratio

$$\rho_n(\theta) = \frac{p_\theta(y_n \mid y_{<n}, x)}{p_{\theta_0}(y_n \mid y_{<n}, x)}.$$

The practical importance-sampling RL loss is then

$$\mathcal{L}_{\mathrm{RL}}(\theta;\theta_0) = - \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \rho_n(\theta) r_{\theta_0}(y_{\le n}, x),$$

with its gradient being:

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta;\theta_0) = -\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \rho_n(\theta) r_{\theta_0}(y_{\le n}, x) \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x).$$

At $\theta = \theta_0$, we have $\rho_n(\theta_0) = 1$, and hence

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta;\theta_0)\Big|_{\theta=\theta_0} = -\mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \nabla_\theta \log p_\theta(y_n \mid y_{<n}, x) \left[\log \frac{q(y_n \mid y_{<n}, x)}{p_{\theta_0}(y_n \mid y_{<n}, x)}\right].$$

Supervised Learning Loss. Next we consider the supervised-learning style loss on the same frozen on-policy prefix data. After sampling $y \sim p_{\theta_0}(\cdot \mid x)$, we freeze the prefixes $(y_{<n}, x)$ and directly minimize the full-vocabulary reverse KL:

$$\mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} D_{\mathrm{KL}}\left(p_\theta(\cdot \mid y_{<n}, x) \| q(\cdot \mid y_{<n}, x)\right).$$

Expanding over the vocabulary $\mathcal{V}$,

$$\mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \sum_{a\in\mathcal{V}} p_\theta(a \mid y_{<n}, x) \log \frac{p_\theta(a \mid y_{<n}, x)}{q(a \mid y_{<n}, x)}.$$

Its gradient is $\nabla_\theta \mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0) =$

$$\mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \mathbb{E}_{a\sim p_\theta(\cdot|y_{<n},x)} \left[\nabla_\theta \log p_\theta(a \mid y_{<n}, x) \log \frac{p_\theta(a \mid y_{<n}, x)}{q(a \mid y_{<n}, x)}\right].$$

The extra $+1$ term from differentiating $p_\theta \log p_\theta$ theoretically vanishes because of score identity. Evaluating the supervised-KL gradient at $\theta = \theta_0$ gives $\nabla_\theta \mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0)\Big|_{\theta=\theta_0} =$

$$\mathbb{E}_{x\sim X}\mathbb{E}_{y\sim p_{\theta_0}(\cdot|x)} \sum_{n=1}^{L_y} \mathbb{E}_{a\sim p_{\theta_0}(\cdot|y_{<n},x)} \left[\nabla_\theta \log p_\theta(a \mid y_{<n}, x) \left(\log \frac{p_{\theta_0}(a \mid y_{<n}, x)}{q(a \mid y_{<n}, x)}\right)\right]_{\theta=\theta_0}.$$

The RL surrogate above is exactly a Monte Carlo estimator of this gradient, using the sampled token $a = y_n$. Therefore,

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta;\theta_0)\Big|_{\theta=\theta_0} = \nabla_\theta \mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0)\Big|_{\theta=\theta_0}$$

in expectation over the sampled token $y_n \sim p_{\theta_0}(\cdot \mid y_{<n}, x)$.

However, the two losses are not globally identical. Away from θ_0, the supervised KL gradient uses the current coefficient

$$\log p_\theta(a \mid y_{<n}, x) - \log q(a \mid y_{<n}, x),$$

whereas the RL surrogate uses the frozen rollout coefficient

$$\log p_{\theta_0}(a \mid y_{<n}, x) - \log q(a \mid y_{<n}, x).$$

Thus, in general,

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta;\theta_0) \neq \nabla_\theta \mathcal{L}_{\mathrm{supKL}}(\theta;\theta_0) \qquad \text{for } \theta \neq \theta_0.$$

The equivalence is only a local gradient equivalence at the rollout policy θ_0.

C More Experimental Details and Analysis

C.1 More details on Experimental Setup

In ***OPD***, we distill a Qwen3-8B teacher into a Qwen3-8B-Base student model which has been initialized by supervised fine-tuning (SFT). For competition mathematics, the student trains for 100 on-policy steps (with a total batch size of 64 groups). For tool use, distillation proceeds for 80 steps (256 rollouts per step) under identical architecture settings. Notably for math reasoning tasks, we used the default hyperparameter setup of the KL divergence baseline already tuned and recommended by tinker of the Thinking Machine Labs, and our OPD+ achieves better results with simple one line change of code under the same

hyperparameter setups. It is possible that better results can be achieved through more detailed sweeping of hyperparameters.

In ***OPSD***, we a use the same model to act as both student and teacher simultaneously by varying the conditioning context: the teacher policy operates under an RL setting by conditioning on privileged information (the problem and its reference solution), while the student policy observes only the problem statement. We use QWEN3-1.7B with a fixed LoRA teacher (rank 64), a learning rate of 5×10^{-6}, and a 1024-token generation budget.

C.2 Other OPD Experimental Results Analysis

To understand the mechanics underlying the stability and performance improvement of OPD+, we trace the single-sample accuracy (`avg@32`) and generation properties across the full training trajectory.

Convergence Acceleration via Variance Control. While our analysis in Section 3.2 shows that the bare cost and analytic weight for Reverse KL induce identical gradients in expectation ($w = g + 1$), the training dynamics reveal a highly consequential empirical phenomenon: Reverse KL$^+$ (solid green line) exhibits significantly greater **sample efficiency and early convergence velocity** than standard Reverse KL (dashed green line).

As shown in Figure 1, on AIME 24, Reverse KL$^+$ undergoes an immediate performance leap, approaching its peak accuracy ($\sim 64\%$) by step 20, whereas the uncorrected baseline lags noticeably behind at the same interval. We observe a matching trend on HMMT 25, where the solid green curve reaches its peak plateau at step 20 while the standard paradigm slowly climbs to catch up in later stages. This acceleration is even more pronounced in the Tool Use task (last figure of Figure 1), where the solid Reverse KL$^+$ curve maintains a dominant upper bound over its dashed counterpart throughout the crucial step 20–40 training window.

In the policy gradient framework, shifting a per-token score function by a state-independent constant $+1$ acts as an exact, analytically derived variance-reduction baseline. Our training curves substantiate that this built-in baseline mechanism effectively stabilizes early gradient updates, converting directly into faster optimization rates and superior sample efficiency during post-training.

Sustained Optimization Resilience of JSD$^+$. Beyond accelerating convergence, the corrected JSD (JSD$^+$, solid purple line) exhibits exceptional optimization stability and late-stage stamina. While the stop gradient JSD baseline (dashed purple line) suffers from immediate and irreversible collapse to 0% accuracy across all tasks, JSD$^+$ not only completely averts this failure mode but continuously expands its performance ceiling as training progresses.

This behavior is most prominently illustrated on AIME 2025 (Figure 1): while other competitive variants (including Reverse KL) plateau or begin to slightly degrade after step 60, the trajectory of JSD$^+$ maintains an upward trajectory, achieving the absolute highest accuracy (50.42%) at step 100. This sustained growth highlights that symmetrically balancing the properties of both forward and reverse distributions, when mathematically formalized via the proper analytic weight $w(u)$, yields a inherently robust training signal that actively counteracts late-stage overfitting or policy degradation.

C.3 Other OPSD Experimental Figures

We also report the concrete training curves of OPSD experiments, on AIME24.

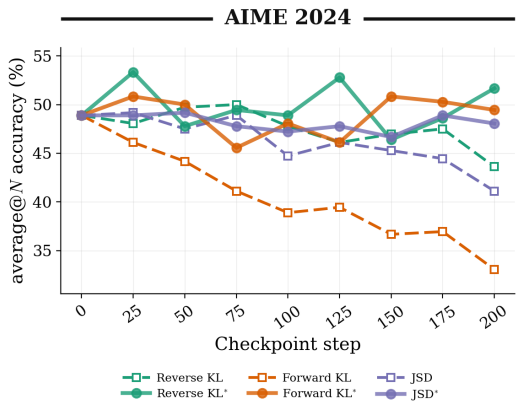

Figure 2: Training dynamics of OPSD vs. OPSD+ across different steps: OPSD+ achieves both better highest performance and clearly prevents catastrophic training collapse.