Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes

Yuqian Fu[1,2*] **Haohuan Huang**[1,2*] **Kaiwen Jiang**[1,2] **Jiacai Liu**[3]
Zhuo Jiang[4] **Yuanheng Zhu**[1,2†] **Dongbin Zhao**[1,2]
[1] State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA
[2] School of Artificial Intelligence, UCAS [3] Fudan University [4] Independent Researcher
{fuyuqian2022, yuanheng.zhu}@ia.ac.cn

Abstract

On-policy distillation (OPD) is increasingly used in LLM post-training because it can leverage a teacher model to provide dense supervision on student rollouts. The standard implementation, however, usually reduces distribution matching to a sampled-token log-ratio, which can make the learning signal fragile on long rollouts whose prefixes drift away from the teacher's typical support. We revisit this formulation from both theoretical and implementation perspectives. Theoretically, token-level OPD is biased relative to sequence-level reverse-KL minimization, but admits a substantially tighter worst-case variance bound; a controlled synthetic study further shows that stronger future-reward coupling increases gradient variance and destabilizes training. Empirically, we identify three failure modes of sampled-token OPD: imbalanced token-level supervision, unreliable teacher guidance on student-generated prefixes, and tokenizer or special-token mismatch. These findings motivate teacher top-K local support matching, a truncated reverse-KL objective that compares teacher and student distributions over a teacher-supported token set at each prefix, together with top-p rollout sampling and special-token masking. Across single-task reasoning and multi-task benchmarks spanning agentic and reasoning settings, this objective improves optimization stability and yields a **+19.8%** performance gain over standard sampled-token OPD baselines, providing a practical recipe for more stable on-policy distillation.

1 Introduction

On-policy distillation (OPD) is becoming an increasingly common component of LLM post-training, especially for reasoning and agentic models. Recent public reports from Thinking Machines Lab (Lu & Lab, 2025), Qwen3 (Yang et al., 2025), MiMo-V2-Flash (Xiao et al., 2026), and GLM-5 (Zeng et al., 2026) suggest a broader shift toward supervision on model-generated trajectories, or closely related on-policy variants, alongside off-policy distillation and reinforcement learning. By training on student rollouts while evaluating them with a stronger teacher, OPD combines on-policy data collection with dense token-level feedback at relatively low cost (Agarwal et al., 2024; Gu et al., 2024). This profile is attractive in practical post-training pipelines, where training efficiency matters and models often need to combine or recover capabilities across domains and training stages (Xiao et al., 2026; Zeng et al., 2026; Wang et al., 2026a; DeepSeek-AI, 2026).

Most OPD implementations in current LLM pipelines use a token-level estimator, even though it is biased relative to sequence-level reverse-KL (Lu & Lab, 2025). A basic reason is that sequence-level objectives couple each token update to many future rewards, which can make optimization substantially noisier in long-horizon settings. We make this trade-off explicit: token-level OPD removes future-reward coupling and is therefore biased, yet it admits more favorable worst-case variance scaling. Our toy experiment shows the same

*Equal contribution. †Corresponding authors.

pattern empirically, with stronger future coupling leading to higher gradient variance and less stable optimization. This suggests a practical design principle for long-horizon training: *keep supervision token-level to control variance.*

In current LLM pipelines, this token-level objective is usually instantiated by sampled-token comparison: at each training step, the update is driven by the teacher–student log-probability difference on the sampled token (Lu & Lab, 2025; Xiao et al., 2026). This implementation is simple and efficient, but its learning signal can become brittle once rollouts grow long. Gu et al. (2024) report degraded outputs such as repetition, which is consistent with our observations. Recent work also reports entropy collapse under sampled-token OPD in certain cases (Ko et al., 2026; Jin et al., 2026b). More recently, full-vocabulary distillation has been reported to outperform sampled-token variants in some settings (Zhao et al., 2026; DeepSeek-AI, 2026), suggesting that the one-token formulation can leave useful teacher information unused.

Our empirical analysis shows that this brittleness stems from several recurring failure modes. In particular, we identify two **objective-level** issues: the one-token signal is often highly imbalanced, and teacher guidance can become unreliable on student-generated prefixes. We also observe an additional **implementation-level** issue from tokenizer or special-token mismatch, which can further distort one-token comparisons. Taken together, these results point to a practical question: *how can we retain the variance advantage of token-level OPD while making its supervision signal less brittle in practice?*

The analysis above suggests a targeted modification of the standard sampled-token objective. We replace one-token supervision with teacher top-K local support matching, where teacher and student are compared over a teacher-supported token subset at each prefix rather than only on the sampled token. We implement this objective as truncated reverse-KL, together with top-p rollout sampling and special-token masking. The resulting update remains local and inexpensive, while providing a less brittle training signal.

Overall, **our main contributions** are as follows:

- We clarify the theoretical trade-off in OPD: token-level OPD is biased relative to sequence-level OPD, but has substantially better worst-case variance scaling with sequence length, making it attractive for long-horizon LLM post-training.
- We present an empirical analysis of why sampled-token OPD can be unstable in practice, highlighting two recurring objective-level issues–imbalanced one-token supervision and unreliable teacher guidance on student-generated prefixes–along with an additional implementation issue from tokenizer or special-token mismatch.
- We propose teacher top-K local support matching as an analysis-driven revision of sampled-token OPD, implemented with truncated reverse-KL, top-p rollouts, and special-token masking, and show that it yields more stable optimization and stronger empirical performance than sampled-token OPD in single-task math reasoning and multi-task agentic-plus-reasoning training.

2 Understanding Sampled-Token OPD: Trade-offs and Failure Modes

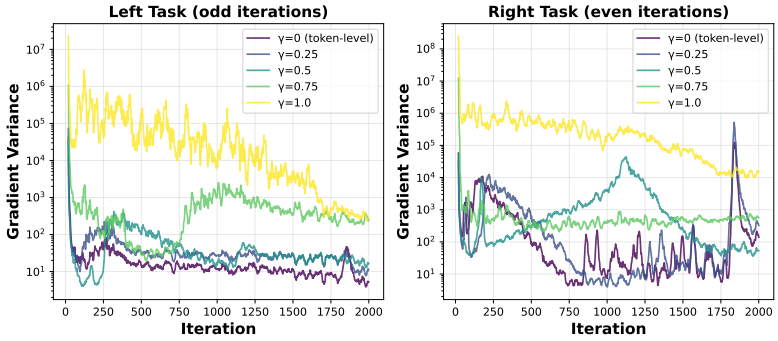
2.1 From reverse-KL to token-level OPD

On-policy distillation aims to transfer the capabilities of a stronger teacher model into a student model by minimizing the reverse-KL from the student to the teacher. For a prompt x, the OPD objective is

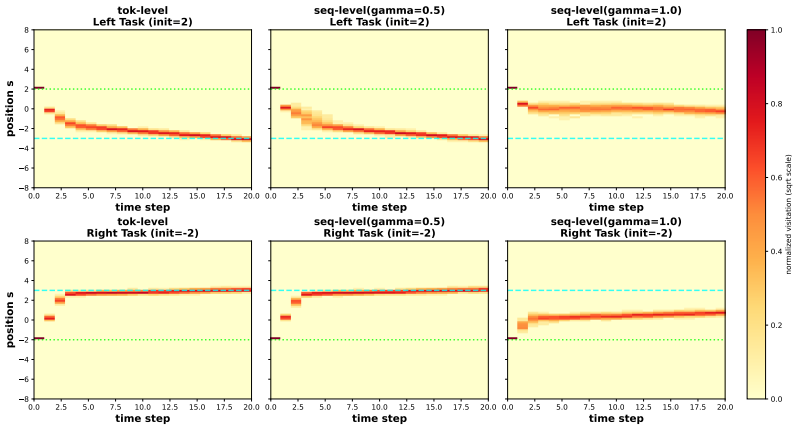
$$J_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim D}\left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, q(\cdot \mid x)\right)\right],$$

where π_θ and q denote the student and teacher models, respectively. Its gradient can be written as

$$\nabla_\theta J_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x,y\sim\pi_\theta(\cdot|x)}\left[\left(\log \pi_\theta(y \mid x) - \log q(y \mid x)\right) \nabla_\theta \log \pi_\theta(y \mid x)\right].$$



(a) Gradient variance in the toy experiment. Larger γ generally yields higher variance in both tasks.



(b) State visitation under $\gamma \in \{0.0, 0.5, 1.0\}$ in the toy environment. For $\gamma = 1.0$, the policy model fails to consistently move toward the target, and instead exhibits drifting behavior.

Figure 1: Effect of increasing γ in the toy experiment. Larger γ yields a higher and more persistent variance regime and, in the sequence-level limit, drifting policies in state space.

For each decoding step t, let $c_t = (x, y_{<t})$ denote the prefix context, and define

$$s_t = \nabla_\theta \log \pi_\theta(y_t \mid c_t), \qquad r_t = \log \frac{\pi_\theta(y_t \mid c_t)}{q(y_t \mid c_t)}.$$

Using the autoregressive factorization

$$\log \pi_\theta(y \mid x) - \log q(y \mid x) = \sum_{t'=1}^{T} r_{t'}, \qquad \nabla_\theta \log \pi_\theta(y \mid x) = \sum_{t=1}^{T} s_t,$$

we obtain the sequence-level estimator

$$\hat{g}_{\text{seq}} = \sum_{t=1}^{T} \left(\sum_{t'=1}^{T} r_{t'} \right) s_t. \tag{1}$$

For $t' < t$, we have $\mathbb{E}[r_{t'} s_t] = 0$, because $r_{t'}$ depends only on the prefix before step t, while

$$\mathbb{E}[s_t \mid x, y_{<t}] = \sum_{y_t} \pi_\theta(y_t \mid c_t)\, \nabla_\theta \log \pi_\theta(y_t \mid c_t) = 0.$$

The same gradient can therefore be expressed in causal reward-to-go form

$$\mathbb{E}[\hat{g}_{\text{seq}}] = \mathbb{E}\left[\sum_{t=1}^{T} \left(\sum_{t'=t}^{T} r_{t'} \right) s_t \right],$$

where each token update is coupled to all future rewards along the trajectory.

Another approximation in LLM training retains only the immediate term at each position

$$\hat{g}_{\text{tok}} = \sum_{t=1}^{T} r_t s_t. \tag{2}$$

We refer to Eq. (2) as the token-level estimator. This approximation removes future-reward coupling, so the update for token y_t depends only on its immediate reward. Consequently, it is biased relative to the sequence-level reverse-KL estimator, but exhibits lower variance in long-horizon settings. Under bounded rewards and bounded gradients, the worst-case variance upper bound of token-level OPD scales as $O(T^2)$, whereas the sequence-level estimator scales as $O(T^4)$. We provide a detailed derivation in Appendix E. To interpolate between these extremes, we consider the discounted return-to-go estimator

$$\hat{g}_\gamma = \sum_{t=1}^{T} \left(\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \right) s_t. \qquad \gamma \in [0, 1] \tag{3}$$

The case $\gamma = 0$ recovers token-level OPD, while $\gamma = 1$ recovers the causal sequence-level estimator. We further validate this trade-off in a two-task toy experiment (Figure 1): stronger future coupling leads to substantially higher gradient variance and less stable optimization. This motivates our focus on token-level supervision in the remainder of the paper, where the main question becomes how to improve its local training signal in practical LLM settings. Additional experimental details are provided in Appendix F.

2.2 Why sampled-token OPD is brittle in practice

Although token-level OPD is appealing from a bias–variance perspective, the standard sampled-token formulation can be brittle in practice. We isolate three failure modes: (1) a highly imbalanced token-level distillation signal, (2) unreliable teacher guidance on student-generated prefixes, and (3) distortions introduced by tokenizer or special-token mismatch. These observations come from sampled-token OPD experiments on math reasoning, using Qwen2.5-7B-Instruct (Qwen et al., 2024) as the student and OpenThinker3-7B (Guha et al., 2025), an SFT model built on Qwen2.5-7B-Instruct, as the teacher.

A highly imbalanced sampled-token signal. In sampled-token OPD, the update at step t is driven by the log-ratio on a single sampled token:

$$\log q(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t).$$

Negative rewards arise whenever the student assigns higher probability to a sampled token than the teacher. As shown in Figure 2, most sampled tokens receive negative reward. This leaves optimization dominated by a small subset of locally positive tokens, so training becomes sensitive to high-frequency fillers and short continuations that can receive favorable local scores while contributing little to trajectory-level quality. The horizontal bands in Figure 2 appear to reflect recurring teacher-side probability levels across different prefixes, while the student probabilities of the same sampled tokens vary more widely. Appendix B.1 discusses this diagnostic pattern; it does not affect the main observation that most points fall below the diagonal.

The teacher signal can become unreliable on student-generated prefixes. Sampled-token OPD assumes that teacher probability on a student-generated token is a useful proxy for trajectory quality. This proxy becomes unreliable on prefixes that are common under the student but uncommon for the teacher. In such regions, tokens with high teacher probability can remain rewarded even after the trajectory has drifted into repetition, self-resetting reasoning, or other meaningless continuations (Figure 3; Appendix I). This creates an objective mismatch between token-level teacher agreement and trajectory-level quality.

We hypothesize that two factors amplify this issue: sharp teacher distributions, where modest teacher–student mismatch can produce large log-ratio values, and growing teacher–student divergence along long rollouts. Consistent with this view, Figure 4 shows that the distribution of teacher–student gaps becomes wider later in the sequence.

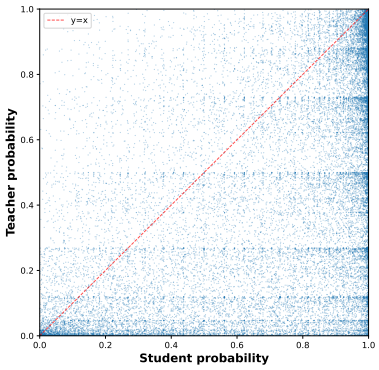


Figure 2: Scatter of token probabilities (teacher vs. student) at the first iteration of sampled-token OPD. The sampled-token signal is heavily skewed toward penalizing the current token.

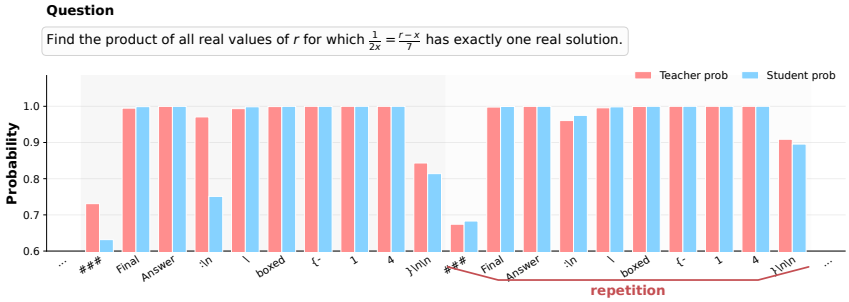


Figure 3: Example of unreliable teacher guidance. The student falls into a repetition loop, yet the teacher remains locally aligned with the student on repeated tokens, indicating that sampled-token OPD may fail to penalize this behavior.

Tokenizer and special-token mismatch. Sampled-token OPD compares the exact token generated by the student using the teacher distribution. When the two models use different tokenizations, the same raw text can be segmented differently, so a student-generated token may not correspond to a natural token under the teacher (Boizard et al., 2025; Patiño et al., 2025; Minixhofer et al., 2025). For example, the student may generate `'<think>'` as `'<'`, `'think'`, `'>'`, while the teacher expects `'<th'`, `'ink'`, `'>'`. Then token `'<'` receives low probability from the teacher, even though both models produce the same semantic content. Similar mismatches arise for special tokens such as end-of-sequence markers. In this setting, a one-token comparison confuses semantic disagreement with tokenizer mismatch. Since supervision is applied on a single token, such mismatch can distort the reward signal.

These observations motivate moving beyond one-token supervision: instead of comparing only the sampled token, we compare teacher and student over a set of teacher-supported next-token continuations at each prefix, while retaining token-level updates for stability.

3 Method

Our method is designed as a direct response to the failure modes above: it retains token-level OPD, but replaces one-token supervision with a distribution-level comparison over a teacher-selected support set at each prefix. This yields a truncated reverse-KL objective that preserves the efficiency of local updates while reducing dependence on any single sampled token. Section 3.1 introduces the objective, and Section 3.2 describes the practical choices that ensure stable training.

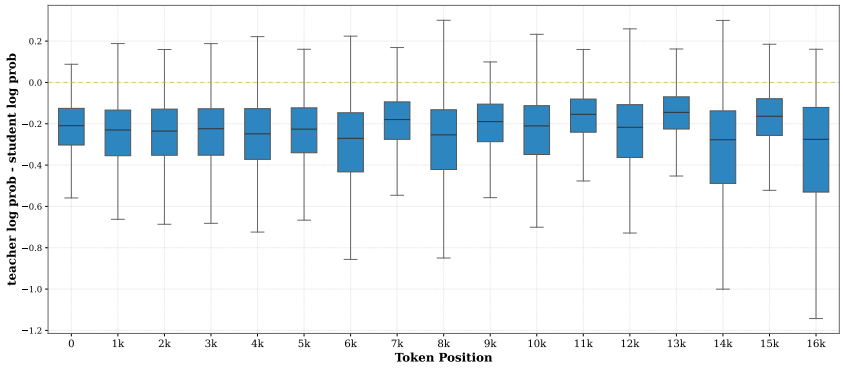


Figure 4: Distribution of teacher–student log-probability gaps across token positions. Several later position buckets exhibit wider lower tails and more extreme values, indicating a noisier teacher signal on long-horizon student-generated rollouts.

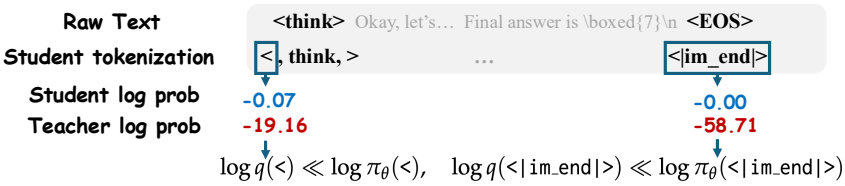


Figure 5: Token-level comparison can penalize semantically correct outputs due to tokenizer mismatch between the teacher and student.

3.1 Teacher top-K local support matching

Instead of comparing teacher and student on a single sampled token, we compare their next-token distributions over a teacher-defined local support set. A natural starting point is the full-vocabulary reverse-KL at prefix c_t:

$$\mathcal{L}_{\text{full}}(c_t) = \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)}. \tag{4}$$

Sampled-token OPD can be viewed as a Monte Carlo approximation to this quantity:

$$\mathcal{L}_{\text{sample}}(c_t, y_t) = \log \frac{\pi_\theta(y_t \mid c_t)}{q(y_t \mid c_t)}, \qquad y_t \sim \pi_\theta(\cdot \mid c_t). \tag{5}$$

This approximation is computationally attractive, but it concentrates the entire update on a single sampled token. We instead compare teacher and student over a teacher-supported candidate set at each prefix.

For each prompt x, we sample a group of outputs $\{o_i\}_{i=1}^{G}$ using the student inference policy. Let $c_{i,t} = (x, y_{i,<t})$ be the prefix at position t of output o_i, and define the teacher support set

$$S(c_{i,t}) = \text{TopK}_q(c_{i,t}), \tag{6}$$

which contains the K highest-probability tokens under the teacher at that prefix.

Within this support, we renormalize both teacher and student distributions:

$$\hat{\pi}_\theta(v \mid c_{i,t}) = \frac{\pi_\theta(v \mid c_{i,t})}{\sum_{u \in S(c_{i,t})} \pi_\theta(u \mid c_{i,t})}, \qquad \hat{q}(v \mid c_{i,t}) = \frac{q(v \mid c_{i,t})}{\sum_{u \in S(c_{i,t})} q(u \mid c_{i,t})}. \tag{7}$$

Our local support matching (LSM) objective averages the truncated reverse-KL over all rollout positions:

$$\mathcal{L}_{\text{LSM}} = \mathbb{E}_{x, \{o_i\} \sim \pi_{\theta,\text{infer}}} \left[\frac{1}{\sum_{i=1}^{G} |o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \sum_{v \in S(c_{i,t})} \hat{\pi}_\theta(v \mid c_{i,t}) \log \frac{\hat{\pi}_\theta(v \mid c_{i,t})}{\hat{q}(v \mid c_{i,t})} \right]. \tag{8}$$

Relative to sampled-token OPD, this replaces a one-token point estimate with a distribution-level comparison over teacher-supported candidates at the same prefix. The update is therefore no longer determined by the sign and magnitude of a single sampled-token log-ratio, while remaining much cheaper than full-vocabulary KL. Local support matching reuses the teacher forward pass already required by sampled-token OPD, but increases teacher-target storage and support-wise KL computation from $O(T)$ to $O(TK)$. Appendix B.2 details this trade-off and compares it with full-vocabulary matching.

3.2 Practical stabilization choices

Support-set renormalization. Renormalization is necessary because the objective is evaluated on a truncated support rather than the full vocabulary. Specifically, we apply a separate softmax over the logits inside the support set, so gradients do not directly propagate to tokens outside this set. Without this step, optimization can become unstable because the teacher and student probability masses inside the support are not directly comparable.

Top-p rollout sampling. We generate rollouts with top-p sampling. Unconstrained sampling occasionally produces very low-probability tokens, creating prefixes on which the teacher signal becomes less informative and optimization less stable. Top-p sampling keeps trajectories closer to typical continuations and makes the teacher signal more reliable.

Special-token masking. We mask problematic special tokens to reduce false negatives caused by incompatible tokenization conventions. This is an orthogonal practical fix: it materially helps sampled-token OPD in our experiments, while our local support objective is much less sensitive to it. In principle, one could also merge multi-token marker variants or average over equivalent tokenizations, but we do not pursue those tokenizer-specific remedies here because masking is the simplest model-agnostic correction.

4 Experiments

We evaluate local support matching in three settings: single-task math reasoning (Section 4.2), alternating multi-task training over math and agentic tasks (Section 4.3), and an additional single-task agentic setting on a smaller student model (Appendix H.1). We also present ablations in Section 4.4 and provide training-dynamics analysis in Appendix H.2.

4.1 Setup

We implement local support matching on top of the verl-agent framework (Feng et al., 2025), using Qwen2.5-Instruct models as students. We consider two main settings. The first is single-task math reasoning, where OpenThinker3-7B (Guha et al., 2025) serves as the teacher and training uses the English portion of DAPO-Math-17K (Yu et al., 2025) with a maximum context length of 16K. The second is a multi-task setting that alternates batches between math reasoning and a multi-turn agentic task based on ALFWorld (Shridhar et al., 2021). In this setting, math uses OpenThinker3-7B (Guha et al., 2025) as the teacher, while the agentic side uses the released GiGPO-Qwen2.5-7B-Instruct-ALFWorld checkpoint (Feng et al., 2025).

For math reasoning, we report pass@1 on five benchmarks: Math500 (Hendrycks et al., 2021), AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For ALFWorld (Shridhar et al., 2021), we report success rate by default. In a small number of cases, we additionally report avg@32 on the math benchmarks. More details on experimental setups can be found in Appendix G.

4.2 Single-task math reasoning

Table 1 shows that local support matching improves over sampled-token OPD in single-task math reasoning. Sampled-token OPD already improves the student from 28.2 to 36.4 average score, but still remains substantially below the teacher. Applying special-token masking to

Method	Math500	AIME24	AIME25	Minerva	OlympiadBench	Avg.
Qwen2.5-7B-It	68.2	13.3	0.0	26.5	32.9	28.2
OpenThinker3-7B	92.2	53.3	40.0	39.0	55.6	56.0
Sampled-token OPD	80.0	10.0	16.7	32.4	43.1	36.4
Sampled-token OPD w/ mask	81.4	**26.7**	16.7	34.2	**44.7**	40.7
Ours w/o mask	80.4	23.3	**26.7**	34.2	43.9	**41.7**
Ours w/ mask	**82.0**	23.3	23.3	**34.9**	43.9	41.5

Table 1: Results on single-task math reasoning.

sampled-token OPD further improves the average to 40.7, indicating that tokenizer-related mismatch is a meaningful part of the failure.

Both variants of our method outperform sampled-token OPD and its masked variant in average score. This shows that the gain is not solely attributable to mismatch handling, and instead supports the role of a stronger distribution-level distillation signal. In addition, masking changes our method only modestly (41.7 vs. 41.5), consistent with the view that local support matching is less sensitive to tokenizer mismatch than one-token supervision. Additional evidence on WebShop (Yao et al., 2022) is shown in Appendix H.1.

4.3 Multi-task agentic-plus-reasoning training

Table 2 shows that the effect of local support matching differs across the two task families in alternating multi-task training. The unmasked version raises the average math score from 34.8 to 41.7 (+19.8%), while maintaining competitive ALFWorld performance. The masked version achieves the best ALFWorld score at 97.7, but gives up part of the math improvement. This pattern suggests that local support matching is especially helpful for the reasoning side of the mixture, where sampled-token signals are more exposed to prefix drift; masking, in contrast, mainly shifts the trade-off toward the agentic task in this run. Beyond evaluation performance, our objective also yields consistently better optimization dynamics. We defer the full learning curves and diagnostic plots to Appendix H.2.

Method	Agentic	Reasoning					
	ALFWorld	MATH500	AIME24	AIME25	Minerva	OlympiadBench	Avg.
Qwen2.5-7B-It	21.9	68.2	13.3	0.0	26.5	32.9	28.2
GiGPO-Qwen2.5-7B-It-Alfworld	95.3	–	–	–	–	–	–
OpenThinker3-7B	–	92.2	53.3	40.0	39.0	55.6	56.0
Sampled-token OPD	90.6	74.8	13.3	13.3	32.1	40.5	34.8
Sampled-token OPD w/ mask	93.8	76.0	20.0	13.3	33.5	40.4	36.6
Ours w/o mask	95.3	**82.0**	**33.3**	**16.7**	32.7	**44.0**	**41.7**
Ours w/ mask	**97.7**	79.0	20.0	16.7	**34.6**	42.5	38.6

Table 2: Results on batch-alternating multi-task training over ALFWorld and math reasoning.

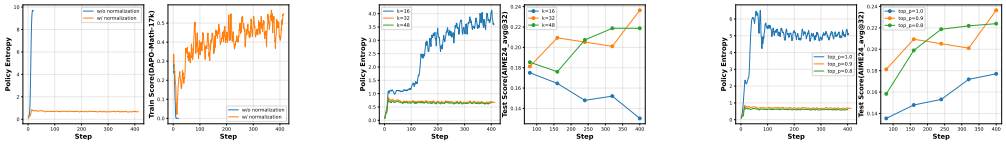
4.4 Ablations

Table 3 and Figure 6 suggest that the gains arise from several design choices rather than any single modification. Teacher top-K comparison alone is not sufficient: the rollout policy must also remain in a stable region, and adding top-p sampling turns an initially weaker top-K variant into a stronger configuration. Under the same top-p rollout conditions, teacher top-K local support matching still improves AIME24 avg@32 from 21.6 to 23.6. Renormalization inside the truncated support is essential, as removing it leads to rapid collapse. Performance is not especially sensitive to the exact support size once K is large enough, but training becomes unstable when the support is too small or rollouts are fully unconstrained.

Top-K support variants. Our main experiments define the truncated expectation on the teacher's top-K support. Alternative objectives can compute the expectation over teacher top-K (used in the main results), student top-K, or teacher top-K augmented with the student-sampled token. We provide their formulations and a preliminary comparison under both the single-task and multi-task settings in Appendix H.3 and Table A3.

Method	AIME24 avg@32
Qwen2.5-7B-Instruct	10.0
OpenThinker3-7B	63.3
Sampled-token OPD	20.4
Sampled-token OPD + top-p	21.6
Teacher top-K	17.7
Teacher top-K + top-p	23.6

Table 3: Component ablation under single-task math training.



(a) Support renormalization. (b) Support size K. (c) Rollout top-p.

Figure 6: Ablations of the main design choices. Renormalization is required for stability, very small support sets hurt learning, and unconstrained rollout sampling degrades optimization.

In the single-task setting, the three variants achieve broadly comparable results. The teacher top-K + sampled-token variant attains the highest average pass@1 score, while student top-K performs competitively on several individual benchmarks, and the original teacher top-K gives the best AIME24 avg@32 among the three. The multi-task results are less uniform. In that setting, the original teacher top-K variant performs substantially better than the other two alternatives, while both student top-K and teacher top-K + sampled-token variants degrade considerably on the math benchmarks. We therefore treat these results as evidence that support construction matters, but do not over-interpret the ranking among variants.

Overall, these results suggest that the choice of where the KL expectation is computed can matter in nontrivial ways, especially in the multi-task setting. We treat this as a partial ablation and a preliminary exploration, and discuss possible causes, including support-set construction and remaining off-policy effects, in Appendix A.

5 Conclusion

In this work, we revisit on-policy distillation (OPD) for LLM post-training from both theoretical and implementation perspectives. Our theoretical analysis shows why token-level OPD is an attractive approximation for long-horizon training: it is biased relative to sequence-level reverse-KL, but avoids future-reward coupling and has substantially better worst-case variance scaling. At the same time, our empirical study shows that the standard sampled-token implementation can provide brittle supervision because its signal is imbalanced, can remain misleading on student-drifted prefixes, and is sensitive to tokenizer or special-token mismatch. Teacher top-K local support matching addresses these issues by preserving local token-level updates while replacing one-token supervision with a truncated distribution-level comparison. Across single-task math reasoning and alternating agentic-plus-reasoning training, this simple modification improves optimization stability and downstream performance over sampled-token OPD, while also clarifying where teacher matching remains an imperfect proxy for task success.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 62136008 and 62293541 and in part by Beijing Nova Program under Grant 20240484514.

References

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

Nicolas Boizard, Kevin El Haddad, CELINE HUDELOT, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=bwRxXiGO9A.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

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

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QXEhBMNrCW.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. SRFT: A single-stage method with supervised and reinforcement fine-tuning for reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=n6E0r6kQWQ.

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

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

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

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

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

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026a. URL https://openreview.net/forum?id=WSRQ37tzk1.

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*, 2026b.

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.

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

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.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang 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.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Zhuo Jiang. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, September 2025. URL https://richardli.xyz/rl-collapse.

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.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DxKP2E0xK2.

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.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family, 2025.

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. *arXiv preprint arXiv:2412.15115*, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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

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*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. UCOB: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026. URL https://arxiv.org/abs/2606.29502.

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

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-RL: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026b.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. MiMo-V2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

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.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026a.

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

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient RL framework secretly brings you off-policy RL training, August 2025. URL https://fengyao.notion.site/off-policy-rl.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

Chenlu Ye, Xuanchang Zhang, Yifan Hao, Zhou Yu, Ziji Zhang, Abhinav Gullapalli, Hao Chen, Jing Huang, and Tong Zhang. Adaptive layerwise perturbation: Unifying off-policy corrections for LLM RL. *arXiv preprint arXiv:2603.19470*, 2026a.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

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

Juzheng Zhang, Abhimanyu Hans, John Kirchenbauer, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Learning from mixed rollouts: Logit fusion as a bridge between imitation and exploration. *Notion Blog*, 2026a. URL https://juzhengz.notion.site/logit-fusion.

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.

Yaocheng Zhang, Jiajun Chai, Yuqian Fu, Songjun Tu, Xiaohan Wang, Wei Lin, Guojun Yin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Are full rollouts necessary for on-policy distillation? *arXiv preprint arXiv:2605.31490*, 2026b. URL https://arxiv.org/abs/2605.31490.

Yaocheng Zhang, Yuanheng Zhu, Wenyue Chong, Songjun Tu, Qichao Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. π-play: Multi-agent self-play via privileged self-distillation without external data. *arXiv preprint arXiv:2604.14054*, 2026c.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

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

Appendix

A Discussion and Limitations

Sampled-token augmentation of the teacher top-K support. In our current implementation, the support is $S(c_t) = \mathrm{TopK}_q(c_t)$: teacher and student distributions are renormalized on this teacher-selected set before computing the KL term. However, this is not the only possible design. One can instead use an augmented support $S^+(c_t) = S(c_t) \cup \{y_t\}$, or include the sampled token through an importance-weighted correction that more explicitly preserves unbiasedness with respect to the full-vocabulary reverse-KL (Zhang & Ba, 2026). We provide formal definitions of several support-augmentation variants, including alternatives based on teacher- or student-centered supports and EMA-PG-style corrections, and evaluate them in Appendix H.3. Overall, our default renormalized formulation remains competitive across both single-task and multi-task settings, which is why we adopt it in the main experiments.

Support-set KL is a truncated objective. Our formulation computes the KL term only over a restricted token subset, i.e., the teacher top-K support, rather than over the full vocabulary. This means that the expected gradient is formed within that subset: tokens outside the support do not receive gradient contributions from this objective. Relative to full-vocabulary reverse-KL, this introduces bias; more generally, it changes where the expectation is taken and which parts of the vocabulary participate in the update. We describe this here as a property of the estimator, rather than as a settled benefit or drawback.

Training–inference mismatch. Prefixes are generated under a rollout policy, e.g., top-p sampling in the vLLM engine, while the training engine updates the model without correcting for this sampling process. This may introduce a training–inference mismatch (Liu et al., 2025; Yao et al., 2025), which we leave to ongoing work.

Teacher matching remains an imperfect proxy for task success. Even when OPD is well defined as a teacher-matching objective, the resulting reward can still diverge from the underlying notion of successful behavior. Our reward-hacking cases make this gap concrete: locally teacher-preferred continuations can remain rewardable even when the overall trajectory is already unhelpful or harmful. A noticeable gap to the teacher also remains in our experiments, which suggests that better local supervision is only one part of the distillation problem, especially when teacher and student differ substantially. Closing that gap may require stronger rollout control, better handling of distribution shift, better use of teacher uncertainty, and combinations with outcome-verifiable rewards.

B Additional clarifications

B.1 Interpreting the probability bands in Figure 2

The horizontal bands in Figure 2 appear when many distinct prefixes assign similar teacher probability to the token sampled by the student. The vertical axis records $q(y_t \mid c_t)$ for a student-sampled token y_t, whereas the horizontal axis records $\pi_\theta(y_t \mid c_t)$. Consequently, recurring teacher-side probability levels can extend into horizontal bands when the corresponding student probabilities vary across prefixes.

One plausible source of these recurring levels is locally similar teacher uncertainty. Two comparably likely next tokens can place the sampled-token probability near $1/2$; formatting, option, or digit-like continuations can yield other recurring levels. Because tokens are sampled from the student, their probabilities can still vary within each teacher-side regime. This is an interpretation of the visual pattern, not a separate empirical claim. The main conclusion remains unchanged: most samples lie below the diagonal, so the signal is skewed toward penalizing the current token.

B.2 Computational profile of local support matching

In the white-box setting considered in this work, local support matching does not require an additional teacher forward pass relative to sampled-token OPD. Both objectives evaluate the teacher on the same student-generated prefixes. Sampled-token OPD retains only the teacher log-probability of the sampled token, whereas local support matching extracts the teacher's top-K tokens and log-probabilities from the same logits. Its additional work therefore comes from top-K extraction, retaining the selected targets, and evaluating the support-wise KL.

For a rollout of length T, sampled-token OPD stores one teacher target per position and evaluates an $O(T)$ sampled-token loss. Local support matching retains K teacher-supported targets per position, giving $O(TK)$ target storage and support-wise KL computation. Full-vocabulary matching instead requires $O(T|\mathcal{V}|)$ targets and computation. With $K = 32$ in our experiments and $|\mathcal{V}| \approx 150{,}000$ for Qwen2.5, local support matching occupies an intermediate point: it uses richer teacher information than sampled-token OPD without materializing the full teacher distribution at every rollout position. The resulting wall-clock and memory costs remain implementation- and hardware-dependent, so we state this algorithmic trade-off without relying on a system-specific overhead measurement.

C Future Directions

Top-p truncation as an adaptive support. Another route toward a more compute-efficient reverse-KL objective is top-p truncation instead of top-K truncation, where the KL divergence is computed on the subset of tokens whose cumulative probability mass reaches a prescribed top-p threshold.

OPD versus RL in multi-task settings. Our multi-task results motivate a more direct comparison between OPD and RL as transfer mechanisms. In RL, positive or negative transfer can be read directly from environment reward across tasks. In OPD, the optimization target remains teacher-derived, so transfer is filtered through what the teacher regards as locally preferable behavior. This distinction may help explain why our multi-task gains are strongest on the math side and why nearby support-set definitions become less uniform in that setting. A matched-task, matched-compute comparison between OPD and RL would help clarify when teacher-guided transfer tracks environment-level generalization and when the teacher–reward gap becomes the bottleneck.

Continual learning as a testbed. Continual learning is another natural setting for OPD. A teacher-guided on-policy objective could act as a retention mechanism while the student adapts to new tasks, but that regime would also stress exactly the issues surfaced in this paper: distribution shift, teacher staleness, and the accumulation of approximation error over long adaptation horizons. Testing OPD there would therefore probe not only whether local support matching mitigates forgetting, but also whether teacher-based objectives remain useful once the student keeps moving away from the teacher's original domain.

Relation to other stabilization directions. This work is complementary to directions such as reward-hacking mitigation, EMA-anchor stabilization with top-K KL (Zhang & Ba, 2026), perturbation-based off-policy correction (Ye et al., 2026a), and logit-level fusion between teacher and student rollouts (Zhang et al., 2026a). These methods address different parts of the same broader problem: how to keep teacher-derived signals useful once teacher and student policies begin to diverge (Li et al., 2026). We view our method as one component in that larger toolbox, rather than as a replacement for those stabilization strategies.

D Related Work

On-policy distillation. Many widely used post-training methods for language models rely on off-policy supervision from fixed datasets, including supervised fine-tuning on

demonstration data (Ouyang et al., 2022; Guha et al., 2025; Fu et al., 2026) and conventional knowledge distillation on teacher-provided targets (Hinton et al., 2015; Sanh et al., 2019). In autoregressive generation, however, such fixed-prefix training creates a mismatch between the contexts seen during training and the prefixes induced by the student's own test-time rollouts, motivating subsequent work on on-policy distillation (OPD) (Agarwal et al., 2024), where supervision is computed on student-generated trajectories. Agarwal et al. (2024) formulate generalized knowledge distillation for autoregressive models, while Gu et al. (2024) derive an effective on-policy optimization procedure. More recently, OPD and closely related variants have appeared in large-scale post-training pipelines, including Qwen3 (Yang et al., 2025), Thinking Machines Lab (Lu & Lab, 2025), MiMo-V2-Flash (Xiao et al., 2026), and GLM-5 (Zeng et al., 2026), underscoring the practical value of on-policy supervision for reasoning transfer, continual adaptation, and capability recovery across training stages. Subsequent work broadens this family in several directions: Ye et al. (2025), Yang et al. (2026b), and Ko et al. (2026) extend OPD to black-box teachers, more flexible reward formulations, and relaxed imitation objectives, respectively. Ye et al. (2026b) further connect OPD with context distillation, showing that on-policy supervision can also be used to internalize context- or prompt-conditioned behaviors. Recent work also studies the rollout horizon itself (Zhang et al., 2026b), while UCOB uses credit-aware bidirectional self-distillation between skill-conditioned and no-skill views for agentic learning (Tu et al., 2026). Related self- or hindsight-distillation approaches replace the external teacher with guidance derived from privileged information or feedback (Zhao et al., 2026; Hübotter et al., 2026; Wang et al., 2026b; Yang et al., 2026a; Zhang et al., 2026c). In contrast, our work does not introduce a new supervision regime; instead, we revisit the standard sampled-token estimator itself and improve its token-level learning signal through teacher top-K local support matching.

KL computation for LLM post-training. Sampled KL estimators in LLM post-training are commonly based on forms such as K_1, K_2, and K_3 (Schulman, 2020). These estimators are lightweight, but they retain only the sampled token and therefore discard the richer next-token distribution available from teacher logits in white-box settings. When teacher logits are available, one can in principle match the full next-token distribution, but materializing full-vocabulary logits is often prohibitively memory-intensive for long responses and large models (Hinton et al., 2015; Kim & Rush, 2016; Agarwal et al., 2024). At one extreme, Zhao et al. (2026) use full-vocabulary logit distillation in a self-distillation setting. To reduce the cost of full-distribution matching, Zhang & Ba (2026) derive an unbiased top-K KL estimator that combines exact computation on a top-K set with a sampled correction for the tail. Related self-distillation methods such as SDPO and OpenClaw-RL also adopt top-K approximations for efficiency, but without explicitly incorporating the sampled token into the truncated support (Hübotter et al., 2026; Wang et al., 2026b). Our method instead computes a truncated reverse-KL objective on the teacher's local top-K support, with both teacher and student distributions renormalized within the selected set. This yields a simple distribution-level alternative to sampled-token OPD without incurring the cost of full-vocabulary matching.

KL divergence in RL for LLMs. The choice of KL divergence can substantially affect distribution matching in on-policy distillation and related post-training methods for language models. Gu et al. (2024) argue that reverse KL is often better suited than the forward-KL objective commonly used in conventional distillation because it discourages the student from overestimating the teacher's low-probability regions and tends to yield more precise generation. Agarwal et al. (2024) further show that different divergences induce a quality–diversity trade-off: moving from forward KL toward reverse KL through generalized Jensen–Shannon divergence (JSD) yields increasingly mode-seeking behavior and lower diversity. A related perspective is provided by Chen et al. (2025), who connect the reverse-KL-like, mode-seeking character of on-policy RL to reduced forgetting during post-training. More recently, Jin et al. (2026a) argue that pure reverse-KL-style OPD can become brittle when the teacher distribution has high entropy, and propose augmenting reverse KL with forward KL on high-entropy tokens to better preserve diversity. At the same time, the preferred divergence can depend on the supervision setting. In self-distillation with privileged information, Zhao et al. (2026) report that forward KL outperforms reverse KL

and JSD, whereas SDPO adopts JSD as a stability-oriented design choice (Hübotter et al., 2026). In this work, we adopt a reverse-KL formulation, but focus on a different question: how to obtain a stable local reverse-KL-style training signal under sampled-token OPD.

E Bias and variance analysis of token-level versus sequence-level OPD

E.1 Bias of the token-level estimator

Recall the sequence-level estimator in causal return-to-go form

$$\hat{g}_{\text{seq}} = \sum_{t=1}^{T} \left(\sum_{t'=t}^{T} r_{t'} \right) s_t.$$

Expanding the inner sum gives

$$\hat{g}_{\text{seq}} = \sum_{t=1}^{T} r_t s_t + \sum_{t=1}^{T} \sum_{t'=t+1}^{T} r_{t'} s_t.$$

Since the token-level estimator keeps only the first term,

$$\hat{g}_{\text{tok}} = \sum_{t=1}^{T} r_t s_t,$$

their expectation gap is

$$\mathbb{E}[\hat{g}_{\text{seq}}] - \mathbb{E}[\hat{g}_{\text{tok}}] = \mathbb{E}\left[\sum_{t=1}^{T} \sum_{t'=t+1}^{T} r_{t'} s_t \right].$$

This makes explicit that token-level OPD removes the future-reward coupling terms and is therefore generally biased with respect to the sequence-level objective.

E.2 Worst-case variance upper bounds

Assume there exist constants $B_r, B_s > 0$ such that

$$|r_t| \le B_r, \qquad \|s_t\| \le B_s \quad \text{for all } t.$$

For the token-level estimator,

$$\|\hat{g}_{\text{tok}}\| \le \sum_{t=1}^{T} |r_t| \, \|s_t\| \le T B_r B_s,$$

which implies

$$\mathbb{E}\|\hat{g}_{\text{tok}}\|^2 \le T^2 B_r^2 B_s^2.$$

Using $\mathrm{Var}(X) \le \mathbb{E}\|X\|^2$, we obtain

$$\mathrm{Var}(\hat{g}_{\text{tok}}) = O(T^2).$$

For the sequence-level estimator, define

$$R = \sum_{t=1}^{T} r_t, \qquad S = \sum_{t=1}^{T} s_t, \qquad \hat{g}_{\text{seq}} = RS.$$

Then

$$|R| \le T B_r, \qquad \|S\| \le T B_s,$$

so

$$\|\hat{g}_{\text{seq}}\| \le T^2 B_r B_s, \qquad \mathbb{E}\|\hat{g}_{\text{seq}}\|^2 \le T^4 B_r^2 B_s^2.$$

Therefore,

$$\mathrm{Var}(\hat{g}_{\text{seq}}) = O(T^4).$$

E.3 Discussion

The sequence-level estimator is closer to the exact trajectory-level objective, but it couples each score term with many future rewards. In worst-case scaling, this changes variance growth from quadratic to quartic in sequence length. The argument is conservative, but it captures why stronger reward coupling can become problematic in long-horizon training.

F Toy experiment details

F.1 Environment

We use a two-task one-dimensional continuous-control environment to visualize how stronger return coupling changes OPD optimization. The student policy is a three-layer MLP with roughly 4K parameters. Its input is a three-dimensional vector containing task identity, current position, and normalized time step. The policy outputs the mean and standard deviation of a Gaussian action distribution, and the state transition is

$$s_{t+1} = s_t + \delta, \qquad \delta \sim \mathcal{N}(\mu, \sigma).$$

The two tasks are mirror images of each other: the left task starts from $+2$ and targets -3, while the right task starts from -2 and targets $+3$. We first train separate teachers with REINFORCE and then distill them into a shared student with alternating-task OPD.

F.2 Gradient variance estimation

At each training step, we split a batch of $B = 64$ trajectories into $M = 8$ micro-batches. For each micro-batch m, we compute a loss $\mathcal{L}_m$ and the corresponding gradient vector $\mathbf{g}_m$ on the output layer parameters. We then estimate gradient variance by

$$\mathrm{Var}(\mathbf{g}) = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{g}_m - \bar{\mathbf{g}}\|^2, \qquad \bar{\mathbf{g}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{g}_m.$$

We use this quantity only as a qualitative proxy, but it is sufficient for comparing relative variance across different γ settings.

F.3 Additional Results of Toy Experiments

Figures A1, A2, and A3 report gradient-variance curves and corresponding state-visitation heatmaps for different OPD estimators ($\gamma \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$) across three random seeds. Although the exact magnitudes vary by seed, the qualitative pattern is consistent. All settings exhibit large variance spikes during early optimization, and larger γ typically remains at a higher variance level later in training. In several runs, the variance under $\gamma = 0.75$ or $\gamma = 1.0$ stays one to several orders of magnitude above that of smaller γ values. Across runs, token-level OPD ($\gamma = 0$) consistently learns trajectories that move toward the target states for both tasks. Intermediate values of γ remain qualitatively similar but become more diffuse. When γ approaches the sequence-level case ($\gamma = 1.0$), the learned trajectories often deviate from the desired direction and stabilize around sub-optimal regions.

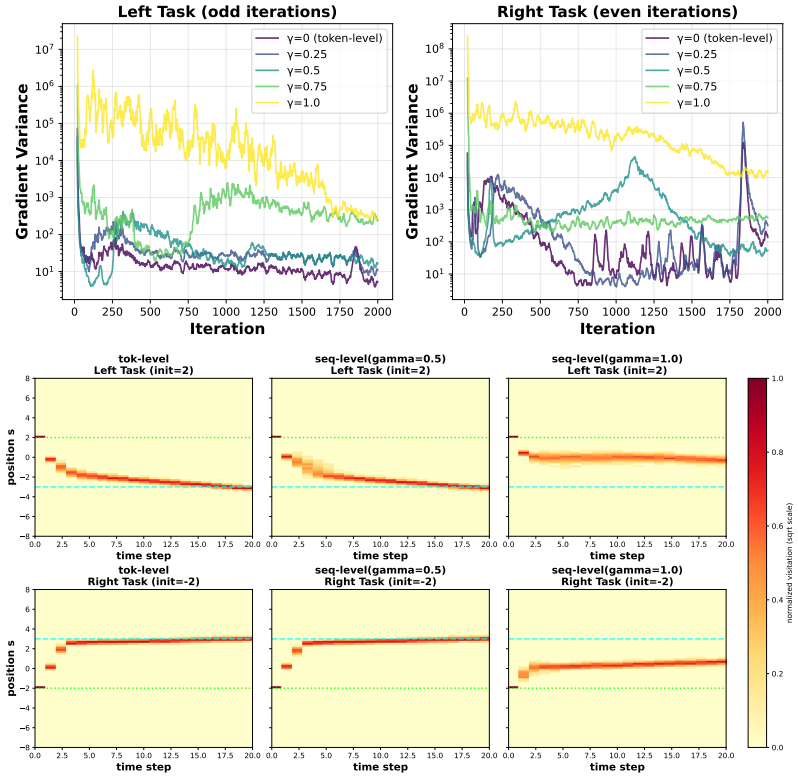


Figure A1: Toy experiment with random seed 42: gradient variance and state visitation.

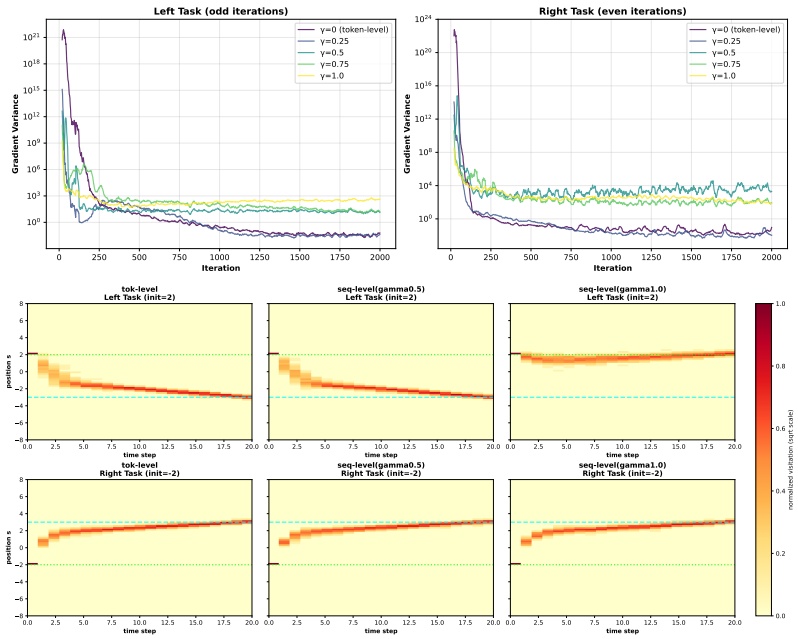


Figure A2: Toy experiment with random seed 43: gradient variance and state visitation.

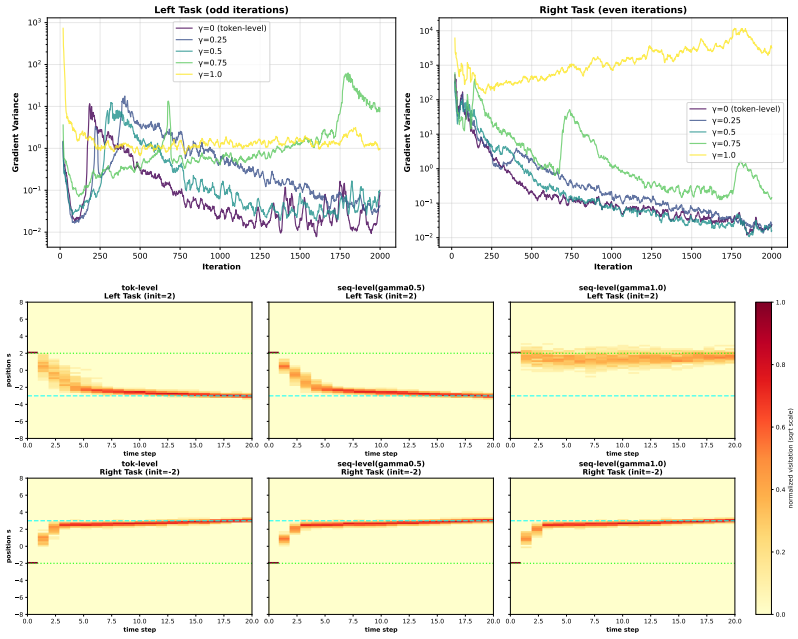


Figure A3: Toy experiment with random seed 2026: gradient variance and state visitation.

G Experiment Setups

We build our training pipeline on verl[1] (Sheng et al., 2025) and the verl-agent framework[2] (Feng et al., 2025), and conduct all experiments on a node with 8 NVIDIA H100 GPUs. We report the settings and hyperparameters used in our experiments in Table A1.

Category	Item	Math (§ 4.2,§ 4.3)	ALFWorld (§ 4.3)	WebShop (§ H.1)
Model	Student	Qwen2.5-7B-Instruct	Qwen2.5-7B-Instruct	Qwen2.5-1.5B-Instruct
	Teacher	OpenThinker3-7B	GiGPO-Qwen2.5-7B-It-ALFWorld	GiGPO-Qwen2.5-1.5B-It-WebShop
Environment	Max prompt length	2048	2048	2048
	Max response length	16384	512	512
	Max turns	–	30	15
Rollout	Group size	8	8	8
	Top-p*	0.9	0.9	0.9
	Temperature	1	1	1
Evaluation	Top-p	0.9	0.9	0.9
	Temperature	1	1	1
Training	Teacher top-K	32	32	32
	Optimizer	AdamW	AdamW	AdamW
	Total training steps†	400	400	60
	Learning rate	2×10^{-6}	2×10^{-6}	2×10^{-6}
	Warmup steps	0	0	0
	Batch size	128	128	128
	Mini batch size	64	64	64

Table A1: Training and evaluation settings. * Rollout top-p is used in local support matching, but not in the sampled-token OPD baseline. † Training steps denote the total number of training updates; in the alternating multi-task setting, 400 total steps correspond to 200 math updates and 200 ALFWorld updates.

[1] https://github.com/verl-project/verl
[2] https://github.com/langfengQ/verl-agent

H Additional Results

H.1 Additional Evidence on Single-task WebShop

To test whether the same trend transfers beyond the main math and ALFWorld settings, we additionally evaluate single-task WebShop (Yao et al., 2022) using Qwen2.5-1.5B-Instruct as the student. As shown in Table A2, local support matching improves over sampled-token OPD on both task score and success rate, increasing success rate from 50.0 to 57.8. Although the gap to the teacher remains substantial, the result suggests that the method transfers beyond the main math/agentic settings. We do not include a masking variant in this setting because the teacher is obtained by RL training from the same base model, so tokenizer or special-token mismatch is less likely to be a dominant factor here.

Method	Task score	Success rate
Qwen2.5-1.5B-It	12.7	2.3
GiGPO-Qwen2.5-1.5B-Webshop (Feng et al., 2025)	81.9	66.4
Sampled-token OPD	73.0	50.0
Ours	**75.1**	**57.8**

Table A2: Results on single-task WebShop with Qwen2.5-1.5B-Instruct as the student. Local support matching improves over sampled-token OPD on both task score and success rate, reducing the gap to the teacher. We do not include a masking variant here because the teacher is obtained by RL training from the same base model, making tokenizer-mismatch effects less central in this setting.

H.2 Training Dynamics and Alignment

Figures A4, A5, and A6 provide a more detailed view of the optimization dynamics.

Better learning curves. On math reasoning, our method improves both training reward and evaluation performance across most of training rather than only at the final checkpoint. This pattern holds in both the single-task setting and the alternating multi-task setting.

More stable optimization. Our method yields smaller gradient norms and lower clipping-boundary fractions while maintaining sufficient policy entropy, indicating more stable optimization. We also observe that special-token masking substantially reduces the clipping-boundary fraction of sampled-token OPD during early and middle training, while having only minor effects on our method.

Improved teacher–student alignment. The teacher–student log-probability gap on sampled tokens also moves closer to zero, suggesting that the truncated local support objective improves alignment even under the sampled-token diagnostic used by the baseline.

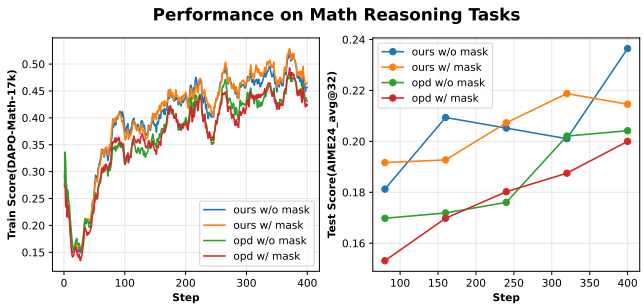


Figure A4: Single-task training curves for math reasoning. Local support matching improves both training reward and evaluation performance over the course of training.

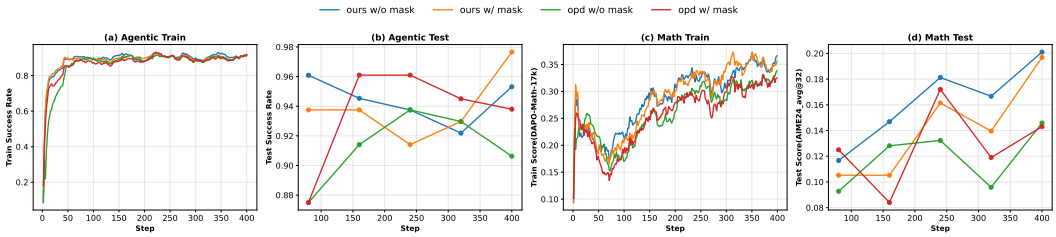
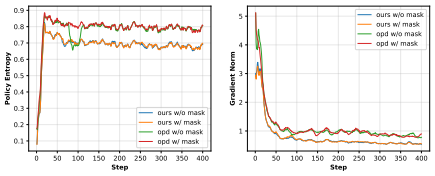
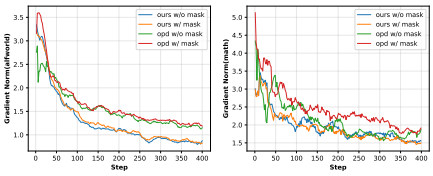


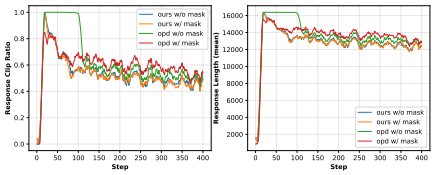
Figure A5: Multi-task learning curves for ALFWorld and math reasoning. The main gains appear on the math side while agentic performance remains strong.



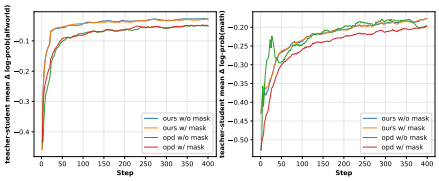
(a) Single-task optimization statistics.



(b) Multi-task gradient norms.



(c) Response length statistics.



(d) Teacher–student log-probability gaps.

Figure A6: Optimization and alignment diagnostics. Relative to sampled-token OPD, local support matching yields smaller gradient norms, fewer clipping-boundary hits, shorter responses, and teacher–student log-probability gaps closer to zero.

H.3 Additional Results of Top-K Variants

We primarily evaluate five top-K variants in our experiments. Below, we present their formulations under unified notation and evaluate them in both single-task math and multi-task settings. All experimental setups are kept the same as in the main experiments.

H.3.1 Formulations of Five Top-K Variants

Shared notation. For each prompt x, we sample a group of outputs $\{o_i\}_{i=1}^{G}$ from the student inference policy π_{infer}. Let $y_{i,t}$ denote the token at position t in output o_i, and let

$$c_{i,t} = (x, y_{i,<t})$$

be the corresponding prefix. We write

$$\mathcal{T}_K^{\pi}(c_{i,t}) = \text{TopK}_{\pi}(c_{i,t}), \qquad \mathcal{T}_K^{q}(c_{i,t}) = \text{TopK}_{q}(c_{i,t}),$$

for the student and teacher top-K token sets at prefix $c_{i,t}$.

All variants below share the same rollout-level aggregation:

$$\mathcal{L}[\ell] = \mathbb{E}_{x,\{o_i\}\sim\pi_{\text{infer}}}\left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\ell(c_{i,t}, y_{i,t})\right].$$

They differ only in how the local loss $\ell(c_{i,t}, y_{i,t})$ is defined.

Renormalized local reverse-KL. For any support set $\mathcal{S}(c_{i,t}) \subseteq \mathcal{V}$, define the renormalized student and teacher distributions on $\mathcal{S}(c_{i,t})$ by

$$\bar{\pi}_\theta(v \mid c_{i,t}; \mathcal{S}) = \frac{\pi_\theta(v \mid c_{i,t})}{\sum_{u \in \mathcal{S}(c_{i,t})} \pi_\theta(u \mid c_{i,t})}, \qquad \bar{q}(v \mid c_{i,t}; \mathcal{S}) = \frac{q(v \mid c_{i,t})}{\sum_{u \in \mathcal{S}(c_{i,t})} q(u \mid c_{i,t})}, \quad v \in \mathcal{S}(c_{i,t}).$$

The corresponding local reverse-KL is

$$\ell_{\text{renorm}}(c_{i,t}; \mathcal{S}) = \sum_{v \in \mathcal{S}(c_{i,t})} \bar{\pi}_\theta(v \mid c_{i,t}; \mathcal{S}) \log \frac{\bar{\pi}_\theta(v \mid c_{i,t}; \mathcal{S})}{\bar{q}(v \mid c_{i,t}; \mathcal{S})}.$$

Variant 1: teacher top-K renormalization (our method). Our default local support matching (LSM) method uses the teacher top-K support

$$\mathcal{S}_{\text{LSM}}(c_{i,t}) = \mathcal{T}_K^q(c_{i,t}),$$

and defines

$$\ell_{\text{LSM}}(c_{i,t}, y_{i,t}) = \ell_{\text{renorm}}(c_{i,t}; \mathcal{S}_{\text{LSM}}).$$

The overall objective is

$$\mathcal{L}_{\text{LSM}} = \mathcal{L}[\ell_{\text{LSM}}].$$

Variant 2: student top-K + sampled token + renormalization. For this variant, the support is obtained by augmenting the student top-K set with the sampled token when needed:

$$\mathcal{S}_{\pi+y}(c_{i,t}) = \begin{cases} \mathcal{T}_K^\pi(c_{i,t}), & y_{i,t} \in \mathcal{T}_K^\pi(c_{i,t}), \\ \mathcal{T}_K^\pi(c_{i,t}) \cup \{y_{i,t}\}, & y_{i,t} \notin \mathcal{T}_K^\pi(c_{i,t}). \end{cases}$$

We then define

$$\ell_{\pi+y}(c_{i,t}, y_{i,t}) = \ell_{\text{renorm}}(c_{i,t}; \mathcal{S}_{\pi+y}),$$

and

$$\mathcal{L}_{\pi+y} = \mathcal{L}[\ell_{\pi+y}].$$

Variant 3: teacher top-K + sampled token + renormalization. Similarly, this variant augments the teacher top-K set with the sampled token:

$$\mathcal{S}_{q+y}(c_{i,t}) = \begin{cases} \mathcal{T}_K^q(c_{i,t}), & y_{i,t} \in \mathcal{T}_K^q(c_{i,t}), \\ \mathcal{T}_K^q(c_{i,t}) \cup \{y_{i,t}\}, & y_{i,t} \notin \mathcal{T}_K^q(c_{i,t}). \end{cases}$$

Its local and rollout-level losses are

$$\ell_{q+y}(c_{i,t}, y_{i,t}) = \ell_{\text{renorm}}(c_{i,t}; \mathcal{S}_{q+y}), \qquad \mathcal{L}_{q+y} = \mathcal{L}[\ell_{q+y}].$$

EMA-PG-style local loss. The remaining two variants follow the EMA-PG decomposition (Zhang & Ba, 2026), where a top-K truncated head term is combined with an importance-weighted sampled-token tail correction to obtain an unbiased estimator of the full-vocabulary reverse-KL. For any support set $\mathcal{S}(c_{i,t}) \subseteq \mathcal{V}$, define the truncated head term

$$\ell_{\text{head}}(c_{i,t}; \mathcal{S}) = \sum_{v \in \mathcal{S}(c_{i,t})} \pi_\theta(v \mid c_{i,t}) \log \frac{\pi_\theta(v \mid c_{i,t})}{q(v \mid c_{i,t})}.$$

Let $\text{sg}(\cdot)$ denote stop-gradient, and define

$$r_{i,t} = \frac{\pi_\theta(y_{i,t} \mid c_{i,t})}{\text{sg}(\pi_\theta(y_{i,t} \mid c_{i,t}))}, \qquad s_{i,t} = \text{sg}\left(\frac{\pi_\theta(y_{i,t} \mid c_{i,t})}{\pi_{\text{infer}}(y_{i,t} \mid c_{i,t})}\right),$$

where $s_{i,t}$ is the sampling-policy correction (optionally clipped, as in EMA-PG). The sampled-token tail correction is

$$\ell_{\text{tail}}(c_{i,t}, y_{i,t}; \mathcal{S}) = s_{i,t}\, \mathbf{1}[y_{i,t} \notin \mathcal{S}(c_{i,t})]\; r_{i,t}\; \text{sg}\left(\log \frac{\pi_\theta(y_{i,t} \mid c_{i,t})}{q(y_{i,t} \mid c_{i,t})}\right).$$

The corresponding local EMA-PG loss is

$$\ell_{\text{EMA}}(c_{i,t}, y_{i,t}; \mathcal{S}) = \ell_{\text{head}}(c_{i,t}; \mathcal{S}) + \ell_{\text{tail}}(c_{i,t}, y_{i,t}; \mathcal{S}).$$

Variant 4: student top-K + EMA-PG correction. This variant uses the student top-K support

$$\mathcal{S}_{\pi\text{-EMA}}(c_{i,t}) = \mathcal{T}_K^{\pi}(c_{i,t}),$$

and defines

$$\ell_{\pi\text{-EMA}}(c_{i,t}, y_{i,t}) = \ell_{\text{EMA}}(c_{i,t}, y_{i,t}; \mathcal{S}_{\pi\text{-EMA}}), \qquad \mathcal{L}_{\pi\text{-EMA}} = \mathcal{L}[\ell_{\pi\text{-EMA}}].$$

Variant 5: teacher top-K + EMA-PG correction. This variant instead uses the teacher top-K support

$$\mathcal{S}_{q\text{-EMA}}(c_{i,t}) = \mathcal{T}_K^{q}(c_{i,t}),$$

and defines

$$\ell_{q\text{-EMA}}(c_{i,t}, y_{i,t}) = \ell_{\text{EMA}}(c_{i,t}, y_{i,t}; \mathcal{S}_{q\text{-EMA}}), \qquad \mathcal{L}_{q\text{-EMA}} = \mathcal{L}[\ell_{q\text{-EMA}}].$$

Summary. The five formulations differ only in how the local support is constructed and whether the sampled token is incorporated through support augmentation or through an EMA-PG-style tail correction. Variants 1–3 compute a renormalized reverse-KL on the resulting local support, while Variants 4–5 replace support renormalization by a truncated head term plus a sampled-token correction (Zhang & Ba, 2026).

H.3.2 Results under Single-task and Multi-task Settings

Method	avg@32	pass@1					
	AIME24	MATH500	AIME24	AIME25	Minerva	OlympiadBench	Avg.
teacher top-K	**23.6**	80.4	23.3	**26.7**	34.2	43.9	41.7
student top-K w/ sampled token (renorm)	22.3	**82.4**	**30.0**	16.7	35.7	44.9	41.9
teacher top-K w/ sampled token (renorm)	22.4	81.6	26.7	23.3	**36.4**	**46.7**	**42.9**
student top-K w/ sampled token (EMA-PG)	21.1	82.0	26.7	23.3	32.4	41.8	41.2
teacher top-K w/ sampled token (EMA-PG)	20.7	81.4	16.7	16.7	32.7	43.0	38.1

(a) Five top-K variants in the single-task setting.

Method	Agentic	Reasoning					
	ALFWorld	MATH500	AIME24	AIME25	Minerva	OlympiadBench	Avg.
teacher top-K	95.3	**82.0**	**33.3**	**16.7**	**32.7**	**44.0**	**41.7**
student top-K w/ sampled (renorm)	95.3	65.6	10.0	10.0	25.0	31.6	28.4
teacher top-K w/ sampled (renorm)	94.5	63.2	10.0	10.0	21.0	30.1	26.9
student top-K w/ sampled token (EMA-PG)	93.8	76.0	20.0	**16.7**	29.4	39.1	36.2
teacher top-K w/ sampled token (EMA-PG)	**96.1**	72.8	16.7	13.3	26.8	38.7	33.7

(b) Five top-K variants in the multi-task setting. We report pass@1 for all math reasoning benchmarks. The final column averages pass@1 metrics only.

Table A3: Ablation of five top-K variants under single-task and multi-task settings.

As shown in Table A3, teacher top-K performs reasonably well in the single-task setting and remains strong in the multi-task setting. By contrast, the EMA-PG variants, despite their unbiasedness motivation, do not translate into better empirical performance here. We therefore treat the variant comparison as a diagnostic result rather than a definitive ranking of support-set objectives.

I Qualitative OPD reward-hacking case study

To complement the representative failures in the main text, we summarize a longer trajectory from multi-task training under sampled-token OPD. Read chronologically, the case exhibits the same pattern in several forms: the model continues reasoning after a valid answer has already been reached, falls into repetition loops such as `wait`, drifts into malformed continuations, and still receives high local teacher probability on those tokens. In these visualizations, uncolored tokens indicate positions where the displayed teacher and student probabilities are approximately matched.

The failure first appears as *over-continuation*. Even after the answer is effectively available, the local signal continues to place substantial mass on generic reasoning fillers and connective tokens, encouraging the model to keep going instead of stopping cleanly. The same pattern later appears on prefixes such as `confirm`, where the local signal still favors additional verification rather than termination. Some of this behavior may also reflect the teacher's own output habits. Figure A7 illustrates several representative cases.

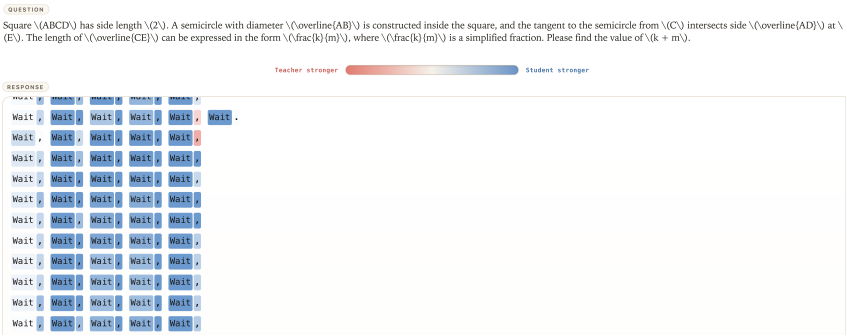
(a) High teacher probability on generic reasoning fillers (`implies`) at step 5.

(b) Teacher and student remain well aligned even after the answer is effectively available.

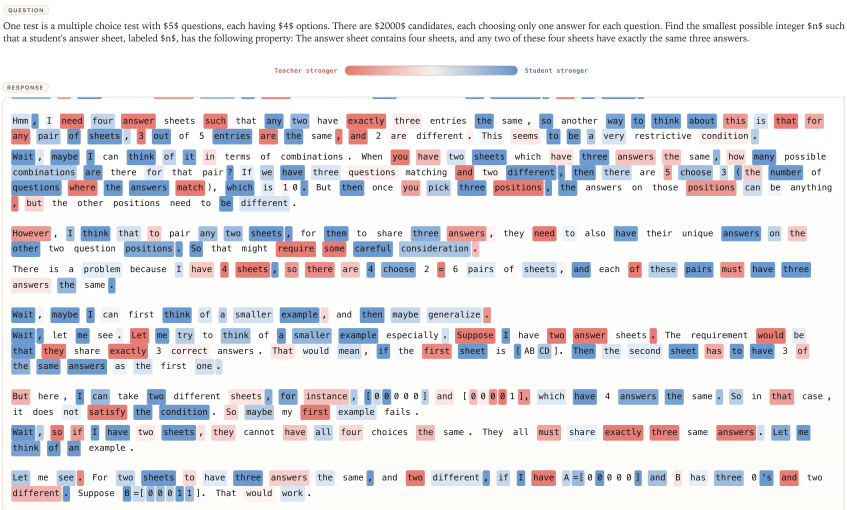
Figure A7: Even after the student has effectively reached an answer, the teacher can still assign high conditional probability to low-information continuations.

The trajectory then develops into *hesitation loops and low-information continuations*. Repeated `wait` tokens, punctuation-heavy continuations, and other semantically weak fillers can remain locally rewardable even after the overall trajectory has become unproductive. This is consistent with the repetition-loop discussion in Section 2.2. We provide two similar cases in Figure A8.

Finally, once the student drifts further off-distribution, the local signal can remain misleadingly positive rather than self-correcting. In the case study, this appears as degenerate continuations and malformed non-English text, yet many tokens still receive high teacher probability. An example is shown in Figure A9.



(a) The teacher may fail to penalize, and sometimes even reinforce, repetitive generation.



(b) Training can also produce overlong chain-of-thought traces with substantial low-quality content, a pattern that may partly reflect the teacher's output style.

Figure A8: Loops, overlong chain-of-thought traces, and punctuation-heavy continuations can remain locally rewardable.

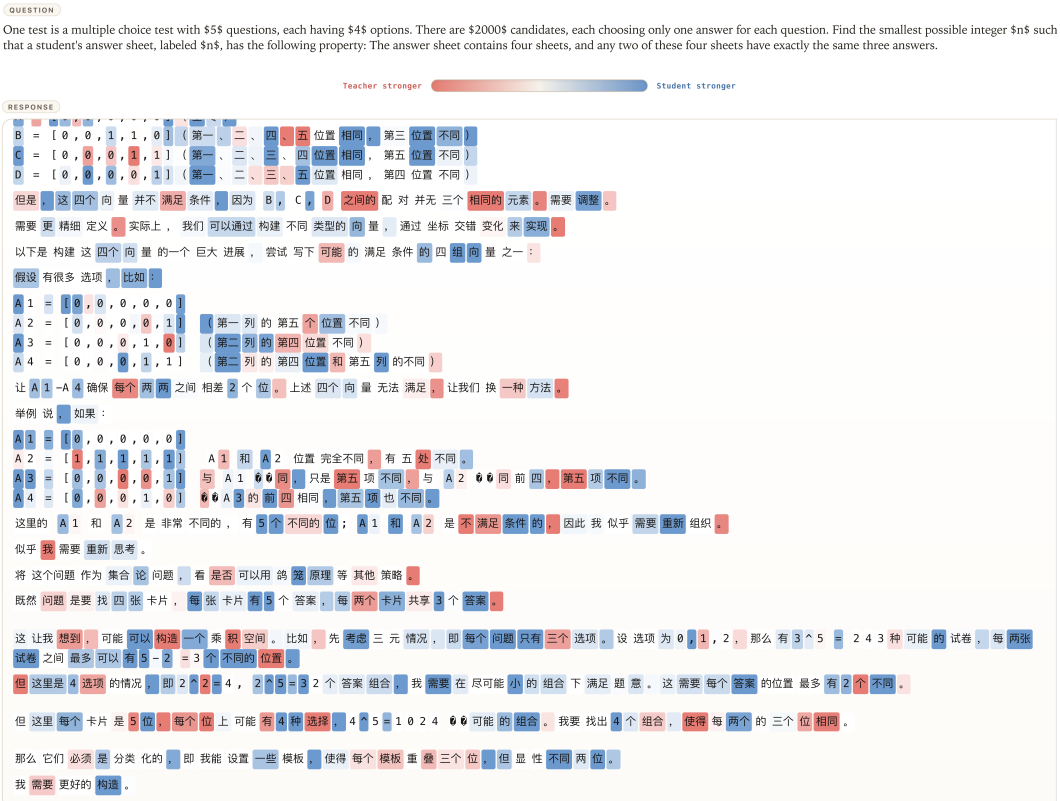


Figure A9: The teacher still assigns high probability to several tokens after the student drifts into malformed non-English output.