TRAIN WHERE THE MODEL IS UNCERTAIN: OFF-POLICY TRACES FOR SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) has emerged as an effective approach for improving large language models. Motivated by the success of on-policy data in reinforcement learning from verifiable rewards (RLVR) and external-teacher distillation, prior work has largely assumed that self-distillation should likewise train on rollouts from the current student policy, leaving off-policy and offline alternatives underexplored. Yet offline training is often more practical. Privileged information such as human feedback may be available only for a fixed corpus of previously collected trajectories and impossible to obtain on live rollouts from the current student policy. Moreover, maintaining a live rollout loop adds substantial computational cost. We study self-distillation when rollouts are generated by policies other than the current student. Surprisingly, offline rollouts outperform on-policy rollouts. We trace this effect to entropy collapse: self-distillation progressively reduces the student's predictive entropy, causing its own rollouts to contain fewer high-entropy tokens—the positions where the privileged-context teacher can provide useful corrections. Offline traces preserve more of these high-entropy positions because they were generated independently of the student's evolving policy. Building on this observation, we introduce two approaches that control rollout staleness to preserve opportunities for the teacher to correct the student. In ***EMA OPSD***, rollouts are generated by an exponential moving average of the student, providing a stale but continually updated rollout policy. In ***offline OPSD***, rollouts come from a fixed corpus generated by other models before training, eliminating rollout generation during training entirely. We additionally introduce high-entropy token masking to focus distillation on uncertain positions. Across mathematical reasoning benchmarks and multi-step Knights and Knaves problems, both approaches substantially outperform standard online OPSD.

1 INTRODUCTION

On-policy external-teacher distillation (OPD) (Agarwal et al., 2024; Gu et al., 2026) has recently emerged as an effective alternative to reinforcement learning from verifiable rewards (RLVR). In OPD, a weaker student policy is trained to match the outputs of a stronger teacher model on the trajectories generated by the student itself. Compared with the sparse trajectory-level reward signal used in RLVR, OPD provides token-level corrections throughout the trajectory, yielding a denser learning signal. Prior work (Agarwal et al., 2024; Gu et al., 2026; Lu & Thinking Machines Lab, 2025) has found distillation to be particularly effective in the on-policy setting, where the teacher directly corrects mistakes made in the student's own generations.

A key limitation of OPD is its reliance on a teacher that is stronger than the student. Such a teacher may be unavailable, particularly for frontier models that outperform comparable alternatives. OPD can also be brittle when teacher and student follow substantially different reasoning patterns, potentially causing destructive interference (Li et al., 2025; Kim et al., 2025). On-policy self-distillation (OPSD) (Song et al., 2026; Zhao et al., 2026a; Shenfeld et al., 2026; Hübotter et al., 2026; Ye et al., 2026) has therefore been proposed as an alternative when no suitable external teacher is available.

Unlike OPD, OPSD constructs its teacher by conditioning the student on privileged information, such as a reference solution to a math problem or textual human feedback. It then uses the same distillation objective as OPD. Recent work has identified failure modes including reasoning collapse

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

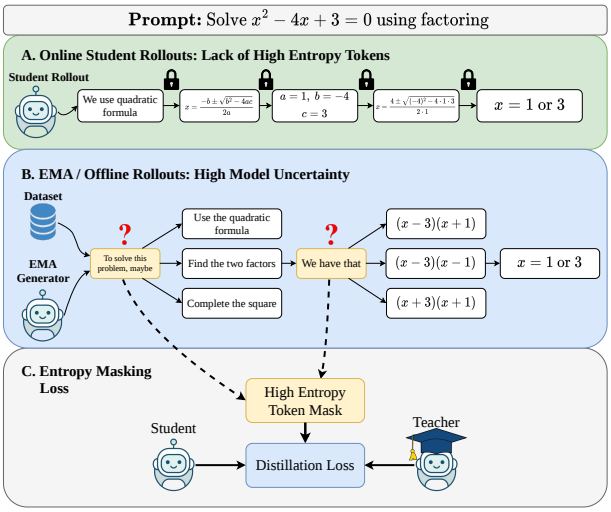

Figure 1: Illustration of how the rollout source affects the available training signal. As the student becomes more confident, its online rollouts contain fewer high-entropy positions. EMA and offline traces can preserve positions where the current student remains uncertain. In the illustration, the online trace is locked into using the quadratic formula, which does not follow the approach requested by the prompt, and the teacher cannot correct the student. On the other hand, the offline traces are more uncertain, so the teacher can encourage the student to perform factoring. We additionally apply a token mask to focus the loss on the uncertain tokens.

and epistemic verbalization, in which the model becomes overconfident and prematurely commits to incorrect answers (Kaur et al., 2026; Peng et al., 2026; Kim et al., 2026b).

Our work is complementary to these efforts. Prior work on self-distillation has largely focused on the training objective: how to assign different weights for tokens, when to gate the teacher, and how to route supervision based on entropy. Instead, we ask which rollouts and tokens the model should train on. This question has practical consequences because it determines whether self-distillation requires a live rollout loop at all. Offline training offers two advantages. First, privileged information such as human textual feedback may not be available during train time because it would require a human in the loop, and could only be acquired in an offline setting. Second, offline training removes rollout generation from the training loop entirely, eliminating the substantial inference cost of an on-policy step in our setup. Our central claim is that the failure of online self-distillation is not only a loss problem, but also a data problem: as training progresses, the student's own rollouts contain progressively fewer high-entropy positions where the privileged-context teacher can provide useful corrections. Motivated by this observation, we introduce two complementary methods: EMA self-distillation, which generates rollouts from a delayed copy of the student, and offline self-distillation, which trains on a fixed corpus generated before training begins. We also provide a token-level mask to further focus on the tokens that are most informative for training.

In summary, our contributions are as follows:

1. We analyze self-distillation through the lens of entropy and show that, unlike external-teacher distillation in our comparison, self-distillation sharply reduces the student's entropy, causing high-entropy positions that carry substantial training signal to disappear from the student's own rollouts. In contrast, we characterize offline traces as a persistent source of high-entropy positions.
2. We introduce two methods that control rollout staleness while applying the same high-entropy token mask. *EMA OPSD* generates rollouts from an exponential moving average of the student, while *offline OPSD* trains on a fixed corpus generated before training begins. Both methods retain the same live teacher and distillation objective.
3. We evaluate our methods on mathematical reasoning and on Knights and Knaves logic puzzles. Both methods outperform standard online self-distillation, while applying the same token-selection rule to on-policy rollouts degrades performance.

2 Related Work

2.1 External Teacher Distillation

Knowledge distillation (Hinton et al., 2015) is a widely used technique in which a student model is trained to imitate a stronger teacher. In the LLM setting, early work on distillation (Zhong et al., 2024) proposed training smaller models to match pretrained larger models. Subsequent work explored alternative divergence objectives for transferring knowledge from teacher to student (Wen et al., 2023).

Modern LLM distillation is often performed on-policy, allowing the teacher to correct mistakes in the student's own generations (Agarwal et al., 2024; Gu et al., 2026). Subsequent work introduces KL regularization (Yang et al., 2026b), interpolates between forward and reverse KL objectives (Jin et al., 2026a; Jia et al., 2026), and considers settings without access to teacher logits (Xiong et al., 2026). Efficiency improvements include early student termination (Zhang et al., 2026a) and offline caching (Wu et al., 2026a). Other work introduces REINFORCE-style baselines for variance reduction (Gu et al., 2026; Oh et al., 2026), identifies tokens amenable to learning (Wang et al., 2026d; Tavor et al., 2026; Sheng et al., 2026), and studies tokens resistant to change under persistent OPD training (Jiang et al., 2026).

2.2 Self-Distillation

Self-distillation uses the student policy conditioned on privileged context as the teacher (Song et al., 2026; Zhao et al., 2026a; Shenfeld et al., 2026; Hübotter et al., 2026; Ye et al., 2026). Applications span language and reasoning tasks (Buening et al., 2026; Qin et al., 2026; Liu et al., 2026c; Zhang et al., 2026d; Rezaei et al., 2026; Sang et al., 2026), agentic tasks (Zhang et al., 2026c; Zhao et al., 2026b; Ma et al., 2026), and other settings, including long-context retrieval, multi-token prediction, visual reasoning, and diffusion language models (Zhang et al., 2026b; Kirchenbauer et al., 2026; Yuan et al., 2026; Su et al., 2026a).

Other approaches combine GRPO (Shao et al., 2024) with self-distillation through improved credit assignment (Yang et al., 2026a; Kim et al., 2026a), selective gating (Li et al., 2026b; Wang et al., 2026c; Ding, 2026), and hybrid training objectives (Lu et al., 2026; Shen et al., 2026; Liang et al., 2026). Related work also combines DPO (Rafailov et al., 2024) with self-distillation (Yu et al., 2026), explores self-reflection (He et al., 2026; Zhang et al., 2026e), and studies self-distillation without an external verification signal (Li et al., 2026c). Additional methods reweight OPD and OPSD trajectories using the geometric mean of the student-teacher probability ratio (Lin et al., 2026) or introduce KL regularization into OPSD (Xu et al., 2026a). We leave an extended related work on self-distillation, entropy, fork tokens, and offline LLM training in the Appendix.

3 Preliminaries

3.1 Self-Distillation

On-policy self-distillation (OPSD) assumes access to a prompt x, a student policy $\pi_\theta(\cdot \mid x)$, and privileged context c, such as the solution to a math problem or textual human feedback. The teacher policy π_T is obtained by conditioning the student on c, yielding $\pi_T = \pi_\theta(\cdot \mid x, c)$.

Let $s_n = \pi_\theta(\cdot \mid x, y_{<n})$ and $t_n = \pi_\theta(\cdot \mid x, c, y_{<n})$ denote the student and teacher next token distributions at position n. Following OPSD, we use the generalized Jensen–Shannon divergence at the forward-KL endpoint ($\beta = 0$), with each vocabulary contribution clipped at ϵ:

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\sum_{n=1}^{T} D(s_n \| t_n) \right], \qquad D(s\|t) = \sum_{v \in \mathcal{V}} \min \left\{ t(v) \log \frac{t(v)}{s(v)}, \epsilon \right\}.$$

The clipping threshold prevents overly confident teacher corrections. Note that the loss assumes that the rollout y is sampled from the current policy $y \sim \pi_\theta(\cdot|x)$. In later sections, we explore alternative rollout sources while keeping the teacher and distillation objective unchanged.

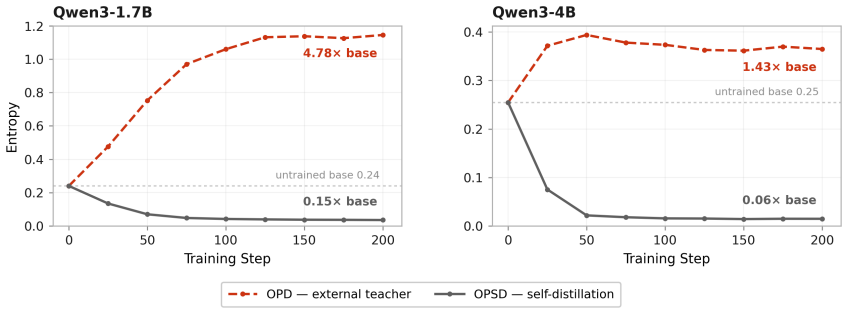

Figure 3: Entropy during self-distillation and external-teacher distillation for Qwen3-1.7B and Qwen3-4B. As seen in the plot, external distillation does not exhibit substantial entropy collapse as training progresses. On the other hand, self-distillation causes intense entropy collapse.

3.2 Rollout Sources and Terminology

Online, *EMA*, and *offline* refer only to the source of the trajectories y. We consider three rollout generators: the current student policy (online), an EMA copy of the student that lags behind the current policy (EMA), and a fixed corpus of traces produced by other models before training begins (offline). In all three settings, the teacher and distillation objective remain unchanged. Only the distribution of states $y_{<n}$ on which the student receives corrections differs. Throughout this paper, when we call a token *uncertain*, we mean that it has high entropy. We do **not** mean that it is a word explicitly expressing uncertainty, such as *maybe* or *possibly*.

4 Motivating Findings

4.1 Entropy Collapse

We begin by comparing the entropy dynamics of external-teacher distillation (OPD) and on-policy self-distillation (OPSD). In OPD, the teacher is an independent model with no access to privileged context, whereas in OPSD the teacher is the student itself conditioned on privileged information. We use Qwen3-1.7B and Qwen3-4B as the students and Qwen3-30B-A3B-Instruct as the external teacher, which substantially outperforms both students. Both self-distillation and external-distillation runs are on-policy, use the same prompts, and sample at the same temperature.

We measure entropy on each model's own generations at temperature 1.0. As shown in Figure 3, under external-teacher distillation, the model does not exhibit a decrease in entropy. In our setup, the entropy of the student model actually increases to a final entropy that is 4.78× and 1.43× the original student entropy for 1.7B and 4B respectively. Under self-distillation, entropy instead drops sharply to between 6% and 15% of the original student entropy.

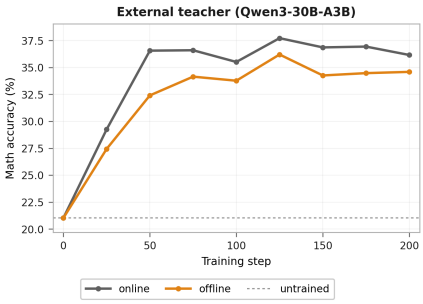

Figure 2: Online versus offline rollouts under external-teacher distillation with a Qwen3-30B-A3B-Instruct teacher and Qwen3-1.7B student. The offline arm replays the eight-source trace pool from Section 4.2, while the teacher and distillation objective remain unchanged. Math accuracy is the mean over AIME24, AIME25, and OlympiadBench. The dotted line denotes the untrained base model.

4.2 Offline Traces

Offline traces are trajectories generated independently of the current student policy. For the math experiments, we construct a pool from 8 source models spanning several model families: Qwen3-0.6B, Qwen3-4B, Qwen3-8B, Llama-3.1-8B-Instruct, Olmo-3-7B-Instruct, Mistral-7B-Instruct-

Table 1: Entropy of offline traces relative to online rollouts. We report the offline-to-online ratio of mean per-token entropy (H) and the fraction of supervised positions exceeding 1 or 2 nats. Ratios above 1 indicate higher presence of high entropy for offline traces. Online rollouts are generated at each checkpoint, while the offline corpus remains fixed. All entropies are measured using the same untrained base model. As training progresses, offline traces retain many more high-entropy positions.

	Qwen3-1.7B			Qwen3-4B			Qwen3-8B		
step	H	>1 nat	>2 nats	H	>1 nat	>2 nats	H	>1 nat	>2 nats
base	1.16	1.18	1.38	1.11	1.12	1.38	1.07	1.07	1.24
50	1.37	1.47	1.76	1.76	2.04	2.88	1.45	1.59	1.99
100	1.36	1.49	1.62	1.73	2.00	3.03	1.55	1.71	2.35
150	1.34	1.44	1.61	1.67	1.91	2.99	1.55	1.70	2.41
200	1.40	1.53	1.74	1.70	1.96	3.12	1.59	1.76	2.47

v0.3, DeepSeek-Math-7B-Instruct, and InternLM2.5-7B-Chat. The resulting pool contains a diverse set of reasoning styles and errors that the student would not necessarily produce on its own.

Given the sharp decrease in entropy during training in self-distillation, we next ask how offline traces differ from the student's own rollouts. We compare the offline dataset with the online rollouts created during online OPSD training. We report mean per-token entropy and the fraction of token entropies above 1 and 2 nats. Following Wang et al. (2025), we refer to these high-entropy positions as *fork tokens*. These positions often occur at transitions between reasoning steps, including words such as *so*, *can*, *since*, *but*, and *now*. See the Appendix for common fork words.

As shown in Table 1, the offline and online sources have nearly the same entropy before training: the untrained model's own rollouts are only 1.07–$1.16\times$ less entropy than the traces. By step 100, the ratio reaches 1.36–$1.73\times$ and it holds there throughout the end of training, indicating that as training continues, online generated trajectories continually have less entropy relative to the fixed offline trajectories. If we look at the concentration of tokens above 1 nat, the offline traces contain 1.5–$2.0\times$ as many positions above 1 nat as online. Moreover, if we look at the concentration of tokens above 2 nats, the offline has 1.7–$3.1\times$ as many as online.

Implication for the rollout source. On-policy rollouts are usually preferred because they allow the teacher to correct the student on the states it actually visits (Agarwal et al., 2024; Gu et al., 2026). In self-distillation, however, this advantage can erode as training progresses and the student's own trajectories contain fewer positions where it is uncertain and can benefit from correction. Under external-teacher distillation, the student's entropy increases, and replacing its rollouts with the offline pool reduces final accuracy from 36.2 to 34.6 (Figure 2). Self-distillation presents a different situation: as entropy decreases, the student's own rollouts contain progressively fewer high-entropy positions, while offline traces continue to contain such uncertain positions. This motivates the hypothesis that offline rollouts may become more useful as high-entropy positions become less prevalent in on-policy self-distillation trajectories.

5 METHODOLOGY

The findings above suggest that self-distillation can benefit from rollout sources that are sufficiently stale for the current student to remain uncertain on their trajectories. At the same time, the training objective should focus on positions where this uncertainty is highest. We therefore introduce two rollout settings, EMA OPSD and offline OPSD, together with a high-entropy token-selection rule that can be applied to either. Algorithm 1 summarizes the full procedure.

5.1 EMA OPSD

EMA OPSD generates trajectories from a delayed copy of the student. During the training, suppose the parameters of the student model and the rollout generator at step k are θ_k and ϕ_k respectively. The generator parameters at step $k+1$ will be

$$\phi_{k+1} = \alpha\phi_k + (1-\alpha)\theta_k,$$

with rollouts sampled as $y \sim \pi_\phi(\cdot \mid x)$. The teacher remains the current student conditioned on privileged context, $\pi_\theta(\cdot \mid x, c)$. Unlike Hübotter et al. (2026), which applies EMA to the teacher, our method applies EMA only to the rollout generator.

5.2 Offline OPSD

Offline OPSD replaces online generation with a fixed corpus $\mathcal{D}$ of previously generated traces. At each step, we sample a prompt x and uniformly sample an associated trace $y \sim \mathcal{D}(x)$. As in EMA OPSD, the teacher and distillation objective remain unchanged; only the rollout distribution differs.

Removing online generation eliminates the need for a rollout loop or separate inference engine, and the same corpus can be reused across runs and student models. It also enables self-distillation when privileged context is available only for previously collected trajectories, such as human feedback on stored conversations.

Algorithm 1 Our Algorithm.

Require: student parameters θ; offline trace dataset $\mathcal{D}$ or EMA decay α; keep fraction k

$\phi \leftarrow \theta$ ▷ ϕ are the rollout-generator parameters
for each training pair (x, c) **do**
  Sample $y \sim \mathcal{D}(x)$ if offline, else $y \sim \pi_\phi(\cdot \mid x)$
  Compute token entropy $H_n = H(\pi_\theta(\cdot \mid x, y_{<n}))$
  $\tau_k \leftarrow$ threshold for top-k fraction entropy positions in y
  $m_n \leftarrow \mathbf{1}[H_n \geq \tau_k]$
  $L \leftarrow \sum_n m_n \, D(\pi_\theta(\cdot \mid x, y_{<n}) \,\|\, \mathrm{sg}[\pi_\theta(\cdot \mid x, c, y_{<n})])$
  Update θ using $\nabla_\theta L$ ▷ batched in practice
  if EMA Generator **then**
    $\phi \leftarrow \alpha\phi + (1-\alpha)\theta$
  end if
end for

5.3 High Entropy Token Masking

Previous papers have observed that high-entropy tokens primarily drive the training of both RLVR (Wang et al., 2025) and SFT (Liu et al., 2026b; Chen et al., 2026). To focus distillation on uncertain positions, we select the top k fraction of tokens by student entropy in each trace $y_{1:T}$, where k is a fixed constant. Formally, let $s_n = \pi_\theta(\cdot \mid x, y_{<n})$ and $t_n = \pi_\theta(\cdot \mid x, c, y_{<n})$ denote the student and privileged-context teacher distributions. We define the mask and objective as

$$L_{\text{mask}}(y) = \sum_{n=1}^{T} \mathbf{1}[H(s_n) \text{ is in the top } k \text{ fraction of token entropies}] \cdot D(s_n \| t_n).$$

We use $k = 0.1$ throughout, so only the 10% most uncertain positions contribute to the loss. The mask changes only which tokens are optimized; the teacher and distillation objective remain unchanged. Trajectories are supplied by the current student, EMA generator, or fixed offline corpus, depending on the training arm.

6 Experiments

6.1 Setup

Models and training. We train Qwen3-1.7B, Qwen3-4B, and Qwen3-8B using LoRA (Hu et al., 2021). Most experiments are run on two NVIDIA GH200 GPUs. Full training hyperparameters are provided in the Appendix. At each model size, all training arms use the same optimization recipe and differ only in the rollout source and whether high-entropy token masking is applied.

Math. We follow OPSD and train on the same OpenThoughts (Guha et al., 2025) math subset with gold solutions. Online and EMA arms sample rollouts, while the offline arm uses the eight-source trace pool from Section 4.2. We evaluate on AIME 2024 and AIME 2025 (30 problems each, avg@12) and OlympiadBench (He et al., 2024) (675 problems, avg@2) using a 16,384-token budget and temperature 1.0, and report the mean across the three benchmarks. For each arm, we report both the best checkpoint and the final checkpoint to measure peak performance and degradation over training.

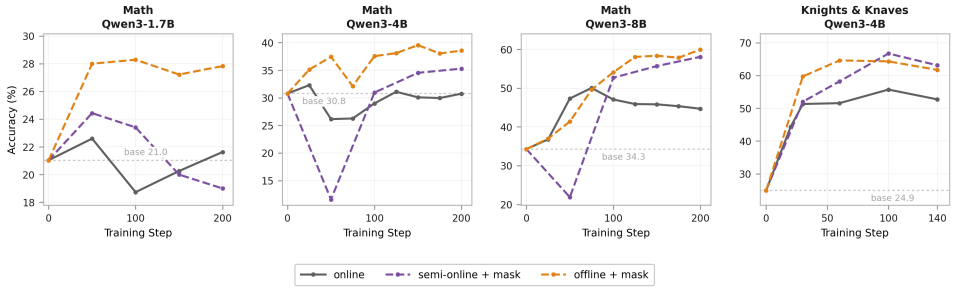

Figure 4: Performance of the three configurations central to our main comparison: online OPSD, EMA OPSD with high-entropy masking, and offline OPSD with high-entropy masking. Columns show Qwen3-1.7B, Qwen3-4B, and Qwen3-8B on math, followed by Qwen3-4B on Knights and Knaves.

Table 2: Peak and final accuracy for all six training configurations. Math is the mean of AIME24, AIME25, and OlympiadBench; Final corresponds to the last checkpoint evaluated for every arm. Best performance in each column is shown in bold and runner-up is underlined.

	Math 1.7B		Math 4B		Math 8B		KK 4B	
	Peak	Final	Peak	Final	Peak	Final	Peak	Final
untrained base	21.0	–	30.8	–	34.3	–	24.9	–
online OPSD	22.6	<u>21.6</u>	32.3	30.7	50.0	44.7	55.7	52.7
online OPSD + mask	24.0	1.8	33.3	12.7	46.0	40.2	52.0	38.6
EMA OPSD	23.4	21.0	32.3	32.3	50.9	49.9	57.5	57.0
EMA OPSD + mask	<u>24.4</u>	19.0	35.3	<u>35.3</u>	<u>58.1</u>	<u>58.1</u>	**66.7**	**63.1**
offline OPSD	21.7	20.4	<u>35.9</u>	34.8	55.9	54.8	63.7	58.6
offline OPSD + mask	**28.3**	**27.8**	**39.5**	**38.6**	**59.9**	**59.9**	<u>64.6</u>	<u>61.7</u>

Knights and Knaves. We additionally evaluate on Knights and Knaves (KK), a second multi-step reasoning environment. Each puzzle describes a set of islanders, where knights always tell the truth and knaves always lie; given their statements about one another, the model must infer each islander's identity. A puzzle is parameterized by the number of people p, depth d, and width w. We train Qwen3-4B on 44,623 puzzles with p=5, d=3, and w=3, and evaluate on 1,000 held-out puzzles with p=5, d=5, and w=4, making the evaluation set harder than the training distribution. All reported KK results use this held-out set. The offline pool contains 5,000 traces from four source models. We procedurally generate the KK problems and use Qwen3-30B-A3B-Instruct to produce solutions, retaining only responses with the correct final answer.

6.2 Main Results

Our empirical evaluation addresses five questions:

- **Q1**: Do off-policy and offline rollout sources outperform standard on-policy sampling?
- **Q2**: How stable are the different training configurations?
- **Q3**: How does offline OPSD compare with online OPSD in computational efficiency?
- **Q4**: How does high-entropy token masking affect performance across different rollout sources?
- **Q5**: Does the correctness of the traces matter?

Table 2 reports all six training configurations, while Appendix A.7 provides results for each math benchmark. Figure 4 highlights our three main configurations: online OPSD, EMA OPSD with masking, and offline OPSD with masking. Offline OPSD with masking achieves the highest peak and final math accuracy at every model size. Its peak advantage over online OPSD grows from 5.7 points at 1.7B to 9.9 points at 8B. On Knights and Knaves, EMA OPSD with masking achieves the highest peak and final accuracy, improving peak accuracy over online OPSD by 11.0 points; offline OPSD with masking also improves over online OPSD. We examine the separate contributions of rollout source and masking in Section 6.3.

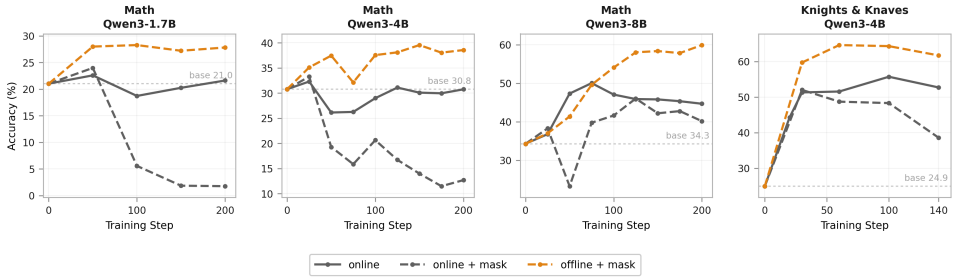

Figure 5: The same high-entropy token-selection rule produces opposite outcomes depending on the rollout source. Masking has higher performance when applied to a fixed offline corpus but degrades performance when applied to the student's own on-policy rollouts.

Stability. We measure stability by the decline from peak to final accuracy. On math, offline OPSD with masking finishes above online OPSD at every model size and drops by at most 0.9 points from peak to final, compared with a drop of up to 5.3 points for online OPSD. EMA OPSD with masking also finishes above online OPSD and exhibits no peak-to-final decline at 4B and 8B. The 1.7B EMA run is an exception: it drops more than online OPSD and finishes below it. On Knights and Knaves, both masked offline and masked EMA finish above online OPSD, although all three configurations exhibit similar peak-to-final declines.

Compute. Offline OPSD is substantially faster than online OPSD at both measured model sizes because it removes rollout generation from the training loop. It provides a $4.3\times$ speedup at 4B and a $4.0\times$ speedup at 8B, as shown in Table 3. These measurements exclude the one-time cost of constructing the offline trace pool, which can be reused across training runs and student models.

Table 3: Wall-clock training time for 200 steps with batch size 256 on two NVIDIA GH200 GPUs. Offline OPSD excludes one-time offline trace generation, since the same trace pool can be reused across different runs.

Method	Qwen3-4B (hours ↓)	Qwen3-8B (hours ↓)
Offline OPSD	1.54	2.23
Online OPSD	6.62	8.86

6.3 Ablations: Rollout Source and Masking

We compare the six configurations in Table 2 to examine the effects of changing the rollout source and selecting high-entropy token positions.

Rollout source. We first compare online, EMA OPSD without masking, and offline OPSD without masking. Compared to online OPSD, offline rollouts improve both peak and final accuracy on math for the two larger models and on Knights and Knaves (by 8.0 points). EMA OPSD matches or modestly improves upon online OPSD in peak accuracy across the evaluated settings. Thus, the alternative rollout sources generally match or improve upon online OPSD.

Masking and rollout source. We next compare masked and unmasked training while holding each rollout source fixed. Masking improves offline OPSD's peak and final math accuracy at every model size, with peak gains of 3.6–6.6 points. On Knights and Knaves, masking also improves peak accuracy by 0.9 points. Crucially, the offline traces are identical in masked and unmasked offline training; only the tokens selected for training change. These results provide evidence that high-entropy tokens carry a particularly useful training signal. As training progresses, such tokens become less prevalent in the student's own rollouts. We also compare masking with no masking on EMA-generated trajectories. Masking improves EMA OPSD's peak accuracy across all evaluated settings, with a 7.2-point gain on 8B math and a 9.2-point gain on Knights and Knaves.

In contrast, the same entropy mask behaves differently on online rollouts. At 1.7B, the masked run drops 22.2 points from peak to final, compared with a 1.0-point drop without masking. Also, in online OPSD, high-entropy masking lowers peak accuracy by 4.0 and 3.7 points on 8B math and Knights and Knaves respectively, and reduces final accuracy in every evaluated setting. As

illustrated in figure 5, the effect of focusing updates on high-entropy positions depends strongly on the offline/online nature of the trajectories.

These results show that the benefit of masking depends strongly on the rollout source: it is beneficial for offline and EMA OPSD but can be harmful for online OPSD. One possible explanation is that offline or EMA rollouts have higher entropy, so the top-k fraction of selected tokens remain sufficiently uncertain for focused distillation to be useful. In contrast, as the online rollouts become increasingly low-entropy, the top-k fraction of selected tokens become less uncertain, making the updates increasingly concentrated and potentially harmful to training.

6.4 Entropy

As mentioned before, self-distillation naturally causes entropy collapse during training and evaluation. A natural question is how our method compares to the entropy collapse of online self-distillation. As seen in Figure 6, we find that our method also exhibits entropy decline. However, compared to online, offline with mask and EMA with mask have significantly less entropy decline across all benchmarks and models. At step 100, the entropies of our methods are 1.67× to 6.15× times those of their online counterparts. The final entropies of all runs are also 1.35× to 6.86× times those of their online counterparts. For space, we leave pass@k discussion to the Appendix.

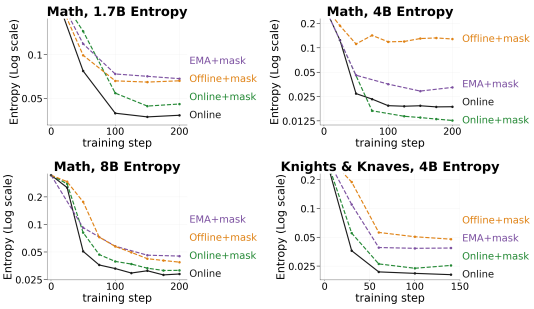

Figure 6: Entropy of evaluation generations under each configuration, measured on AIME for math and on the held-out set for KK. Across models, our method preserves more entropy than online self-distillation.

6.5 Does the Correctness of the Traces Matter?

A natural concern is that offline traces may only be useful when they are predominantly correct. To test this, we train separate Qwen3-8B students on traces from Olmo-3-7B-Instruct and Llama-3.1-8B-Instruct. Table 4 compares each source's trace accuracy with the peak AIME performance of the student trained on that source.

Table 4: Trace correctness does not determine how useful a rollout source is for self-distillation. Results of training Qwen3-8B on different trace sources, evaluated with avg@12 on AIME 2024 and AIME 2025. †Because the online arm has no fixed trace pool, we measure the trace accuracy on the untrained student's own rollouts.

Rollout source	Trace accuracy (%)	Student peak accuracy (%)
Olmo-3-7B-Instruct	87.7	**56.5**
Llama-3.1-8B-Instruct	30.7	<u>53.5</u>
Online OPSD	76.4†	42.1

As shown in Table 4, trace correctness does not determine peak accuracy. The two offline pools lie at opposite ends of the correctness range: 87.7% of Olmo traces are correct, compared with only 30.7% of Llama traces, which is substantially lower than the accuracy of the online rollouts. Despite this large difference, the resulting students peak within 3.0 points of each other, and both substantially outperform online OPSD. These results suggest that high accuracy of offline traces is not necessary for high performance of the resulting student.

7 Conclusion

Self-distillation drives a substantial reduction in the student's entropy, causing its on-policy rollouts to contain progressively fewer high-entropy positions and become less useful for continued training. Stale and offline rollout sources preserve more of these uncertain positions, and we find that their usefulness is not determined by trace correctness alone. By training on trajectories where the current student remains uncertain, EMA and offline OPSD provide stronger and more stable learning signals than standard on-policy self-distillation, while offline OPSD additionally removes the cost of rollout generation during training.

AI USE STATEMENT

As part of our method, we used LLMs to generate synthetic data: the offline trace pools were generated using 8 different LLM models for math and 4 different LLM models for KK (see Sections 4.2 and 6.1). The Knights and Knaves reference solutions were produced by Qwen3-30B-A3B-Instruct (Section 6.1). We also used AI coding assistants to help write supporting code that was checked by a human. The methods themselves were implemented by the authors. LLMs were also used for proofreading and polishing the paper. We have reviewed all AI-assisted work, including all AI-generated code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

We did not use AI to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Rishabh Agrawal, Jacob Fein-Ashley, and Paria Rashidinejad. Reinforcement learning from rich feedback with distributional dagger, 2026. URL https://arxiv.org/abs/2606.05152.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions, 2026. URL https://arxiv.org/abs/2603.12273.

Souradip Chakraborty, Noah Ziems, Furong Huang, Meng Jiang, Amrit Singh Bedi, and Omar Khattab. Pedagogical rl: Teaching models to teach themselves from privileged information. 2026. https://noahziems.com/pedagogical-rl.

Yijie Chen, Yijin Liu, and Fandong Meng. Sed-sft: Selectively encouraging diversity in supervised fine-tuning, 2026. URL https://arxiv.org/abs/2602.07464.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective, 2025. URL https://arxiv.org/abs/2506.14758.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback, 2024. URL https://arxiv.org/abs/2403.00409.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation, 2026. URL https://arxiv.org/abs/2603.23871.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization, 2025. URL https://arxiv.org/abs/2510.14545.

Zitian Gao, Lynx Chen, Haoming Luo, Joey Zhou, and Bryan Dai. One-shot entropy minimization, 2025. URL https://arxiv.org/abs/2505.20282.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models, 2026. URL https://arxiv.org/abs/2306.08543.

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

Zihao Han, Tiangang Zhang, Huaibin Wang, and Yilun Sun. Adaptive teacher exposure for self-distillation in llm reasoning, 2026. URL https://arxiv.org/abs/2605.11458.

Yuto Harada, Yusuke Yamauchi, Yusuke Oda, Yohei Oseki, Yusuke Miyao, and Yu Takagi. Massive supervised fine-tuning experiments reveal how data, layer, and training factors shape llm alignment quality, 2025. URL https://arxiv.org/abs/2506.14681.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision, 2026. URL https://arxiv.org/abs/2604.12002.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search, 2025. URL https://arxiv.org/abs/2506.11902.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Jiazhen Huang, Xiao Chen, Xiao Luo, Yong Dai, Senkang Hu, and Yuzhi Zhao. Skill-conditioned gated self-distillation for llm reasoning, 2026a. URL https://arxiv.org/abs/2605.28791.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M. Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling, 2026b. URL https://arxiv.org/abs/2507.01679.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 2026. URL https://arxiv.org/abs/2605.06387.

Chenyang Jiang and Changhan Huang. Smopd: Selective token-entropy masking for dirty-history multi-turn on-policy self-distillation, 2026. URL https://arxiv.org/abs/2608.14647.

Yuxuan Jiang, Runchao Li, Shubhashis Roy Dipta, Dawei Li, and Zhao Yang. Cornerstones or stumbling blocks? deciphering the rock tokens in on-policy distillation, 2026. URL https://arxiv.org/abs/2605.09253.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models, 2026a. URL https://arxiv.org/abs/2603.07079.

Yiqiao Jin, Yiyang Wang, Lucheng Fu, Yijia Xiao, Yinyi Luo, Haoxin Liu, B. Aditya Prakash, Josiah Hester, Jindong Wang, and Srijan Kumar. Unisd: Towards a unified self-distillation framework for large language models, 2026b. URL https://arxiv.org/abs/2605.06597.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models, 2026. URL https://arxiv.org/abs/2607.05184.

Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting self-uncertainty in on-policy self-distillation for efficient llm reasoning, 2026. URL https://arxiv.org/abs/2605.13255.

Jaehoon Kim, Kwangwook Seo, and Dongha Lee. In their own words: Reasoning traces tailored for small models make them better reasoners, 2025. URL https://arxiv.org/abs/2509.22230.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled rlvr, 2026a. URL https://arxiv.org/abs/2605.10781.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms?, 2026b. URL https://arxiv.org/abs/2603.24472.

John Kirchenbauer, Abhimanyu Hans, Brian Bartoldson, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Multi-token prediction via self-distillation, 2026. URL https://arxiv.org/abs/2602.06019.

Gaotang Li, Ruizhong Qiu, Xiusi Chen, Heng Ji, and Hanghang Tong. Beyond log likelihood: Probability-based objectives for supervised fine-tuning across the model capability continuum, 2026a. URL https://arxiv.org/abs/2510.00526.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing, 2026b. URL https://arxiv.org/abs/2604.02288.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision, 2026c. URL https://arxiv.org/abs/2608.06296.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners, 2025. URL https://arxiv.org/abs/2502.12143.

Zihan Liang, Yufei Ma, Ben Chen, Zhipeng Qian, Xuxin Zhang, Huangyu Dai, and Lingtao Mao. Search-e1: Self-distillation drives self-evolution in search-augmented reasoning, 2026. URL https://arxiv.org/abs/2605.22511.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay, 2026. URL https://arxiv.org/abs/2607.04763.

Chen Lin, Kedi Chen, and Wei Zhang. Renio: Reweighting negative trajectory importance for llm on-policy distillation, 2026. URL https://arxiv.org/abs/2606.23104.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need, 2025. URL https://arxiv.org/abs/2404.07965.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip S. Yu, and Meng Cao. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights, 2025a. URL https://arxiv.org/abs/2410.04350.

Haoran Liu, Yuwei Zhang, Xiyao Li, Bohan Lyu, and Jingbo Shang. Hero: Hindsight-enhanced reflection from environment observations for agentic self-distillation, 2026a. URL https://arxiv.org/abs/2606.11559.

Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, and ShaoGuo Liu. Ettrl: Balancing exploration and exploitation in llm test-time reinforcement learning via entropy mechanism, 2025b. URL https://arxiv.org/abs/2508.11356.

Qi Liu, Mingdi Sun, Yongyi He, Zhi Zheng, Tong Xu, Yi Zheng, Zhefeng Wang, and Enhong Chen. Entropy-kl divergence-based token masking: A novel approach for selective fine-tuning of large language models, 2026b. URL https://arxiv.org/abs/2605.29303.

Yihong Liu, Raoyuan Zhao, Michael A. Hedderich, and Hinrich Schütze. Crosslingual on-policy self-distillation for multilingual reasoning, 2026c. URL https://arxiv.org/abs/2605.09548.

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.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2605.15155.

Yufei Ma, Zihan Liang, Ben Chen, Zhipeng Qian, Huangyu Dai, Lingtao Mao, Xuxin Zhang, Chenyi Lei, and Wenwu Ou. Sd-search: On-policy hindsight self-distillation for search-augmented reasoning, 2026. URL https://arxiv.org/abs/2605.18299.

Andrei Liviu Nicolicioiu, Mohammad Pezeshki, and Aaron Courville. On-policy self-distillation with sampled demonstrations reduces output diversity, 2026. URL https://arxiv.org/abs/2606.26091.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. Kl for a kl: On-policy distillation with control variate baseline, 2026. URL https://arxiv.org/abs/2605.07865.

Motoki Omura, Yasuhiro Fujita, and Toshiki Kataoka. Entropy controllable direct preference optimization, 2025. URL https://arxiv.org/abs/2411.07595.

Keqin Peng, Chen Li, Yuanxin Ouyang, Yancheng Yuan, and Liang Ding. Diagnosing and mitigating thinking collapse in on-policy self-distillation, 2026. URL https://arxiv.org/abs/2607.10805.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning, 2025. URL https://arxiv.org/abs/2505.22660.

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved), 2025. URL https://arxiv.org/abs/2507.12856.

Ruiyang Qin, Qingzhuo Wang, Dongrui Liu, Qiang Li, Zhihua Wei, and Wen Shen. Multilingual safety alignment via self-distillation, 2026. URL https://arxiv.org/abs/2605.02971.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration, 2026. URL https://arxiv.org/abs/2601.18779.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

MohammadHossein Rezaei, Anas Mahmoud, Zihao Wang, Utkarsh Tyagi, Advait Gosai, Razvan-Gabriel Dumitru, Aakash Sabharwal, Bing Liu, and Yunzhong He. Rubric-guided self-distillation: Post-training without rubric verifiers, 2026. URL https://arxiv.org/abs/2606.12507.

Zhiwen Ruan, Yixia Li, He Zhu, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. Enhancing large language model reasoning via selective critical token fine-tuning, 2025. URL https://arxiv.org/abs/2510.10974.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation, 2026. URL https://arxiv.org/abs/2603.05433.

Amrith Setlur, Zijian Wang, Andrew Cohen, Paria Rashidinejad, and Sang Michael Xie. Reuse your flops: Scaling rl on hard problems by conditioning on very off-policy prefixes, 2026. URL https://arxiv.org/abs/2601.18795.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guobin Shen, Xiang Cheng, Chenxiao Zhao, Lei Huang, Jindong Li, Dongcheng Zhao, and Xing Yu. Anti-self-distillation for reasoning rl via pointwise mutual information, 2026. URL https://arxiv.org/abs/2605.11609.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less, 2025. URL https://arxiv.org/abs/2509.04259.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Huanxin Sheng, Zhiling Ye, Haonan Wang, Jian Wang, Jinjie Gu, and Jian Kang. 1% of tokens can be enough: On gradient estimation in on-policy distillation, 2026. URL https://arxiv.org/abs/2609.24432.

Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J. Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback, 2026. URL https://arxiv.org/abs/2602.02482.

Xingyu Su, Jacob Helwig, Shubham Parashar, Atharv Chagi, Lakshmi Jotsna, Degui Zhi, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Data-efficient autoregressive-to-diffusion language models via on-policy distillation, 2026a. URL https://arxiv.org/abs/2606.06712.

Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning, 2026b. URL https://arxiv.org/abs/2509.20712.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data, 2024. URL https://arxiv.org/abs/2404.14367.

Almog Tavor, Itay Ebenspanger, Neil Cnaan, and Mor Geva. Rethinking selective knowledge distillation, 2026. URL https://arxiv.org/abs/2602.01395.

Hao Wang, Hao Gu, Hongming Piao, Kaixiong Gong, Yuxiao Ye, Xiangyu Yue, Sirui Han, Yike Guo, and Dapeng Wu. Entropy-preserving supervised fine-tuning via adaptive self-distillation for large reasoning models, 2026a. URL https://arxiv.org/abs/2602.02244.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, Guorui Zhou, and Ling Pan. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr, 2026b. URL https://arxiv.org/abs/2507.15778.

Jiaxuan Wang, Xuan Ouyang, Zhiyu Chen, Yulan Hu, Zheng Pan, Xin Li, and Lan-Zhe Guo. Trace: Distilling where it matters via token-routed self on-policy alignment, 2026c. URL https://arxiv.org/abs/2605.10194.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation, 2026d. URL https://arxiv.org/abs/2605.26844.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation, 2023. URL https://arxiv.org/abs/2307.15190.

Chao-Chung Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, Shao-Hua Sun, and Hung yi Lee. Mitigating forgetting in llm fine-tuning via low-perplexity token learning, 2025. URL https://arxiv.org/abs/2501.14315.

Yecheng Wu, Song Han, and Hai Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation, 2026a. URL https://arxiv.org/abs/2604.13010.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification, 2026b. URL https://arxiv.org/abs/2508.05629.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, Peng Sun, Tao Gui, Qi Zhang, and Xuanjing Huang. Training large language models for reasoning through reverse curriculum reinforcement learning, 2024. URL https://arxiv.org/abs/2402.05808.

Jing Xiong, Hui Shen, Shansan Gong, Yuxin Cheng, Jianghan Shen, Chaofan Tao, Haochen Tan, Haoli Bai, Lifeng Shang, and Ngai Wong. Ovd: On-policy verbal distillation, 2026. URL https://arxiv.org/abs/2601.21968.

Jiawei Xu, Minghui Liu, Juzheng Zhang, Tom Goldstein, and Furong Huang. β-opsd: Deriving with policy optimization, training with self-distillation, 2026a. URL https://arxiv.org/abs/2607.28582.

Zaiyan Xu, Sushil Vemuri, Kishan Panaganti, Dileep Kalathil, Rahul Jain, and Deepak Ramachandran. Robust llm alignment via distributionally robust direct preference optimization, 2026b. URL https://arxiv.org/abs/2502.01930.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr, 2026a. URL https://arxiv.org/abs/2604.03128.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026b. URL https://arxiv.org/abs/2602.12125.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in rl for llms, 2025. URL https://arxiv.org/abs/2505.12929.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models, 2026. URL https://arxiv.org/abs/2602.12275.

Woongyeong Yeo, Yumin Choi, Taekyung Ki, and Sung Ju Hwang. Hint-sd: Targeted hindsight self-distillation for long-horizon agents, 2026. URL https://arxiv.org/abs/2605.17873.

Xin Yu, Liuchen Liao, Yiwen Zhang, Yingchen Yu, Lingzhou Xue, and Qinzhen Guo. Preference-based self-distillation: Beyond kl matching via reward regularization, 2026. URL https://arxiv.org/abs/2605.05040.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation, 2026. URL https://arxiv.org/abs/2605.18740.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes, 2026a. URL https://arxiv.org/abs/2602.15260.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models, 2026b. URL https://arxiv.org/abs/2604.17535.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning, 2025. URL https://arxiv.org/abs/2506.17211.

Yan Zhang, Daiqing Wu, Huawen Shen, Can Ma, and Yu Zhou. Learn where to click from yourself: On-policy self-distillation for gui grounding, 2026c. URL https://arxiv.org/abs/2605.00642.

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, 2026d. URL https://arxiv.org/abs/2604.14054.

Yuwei Zhang, Sha Li, Changlong Yu, Qin Lu, Shuowei Jin, Chengyu Dong, Haoran Liu, Ilgee Hong, Xintong Li, Zhenyu Shi, Bing Yin, and Jingbo Shang. Learning with rare success but rich feedback via reflection-enhanced self-distillation, 2026e. URL https://arxiv.org/abs/2605.12741.

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, 2026a. URL https://arxiv.org/abs/2601.18734.

Zhengyang Zhao, Lu Ma, and Wentao Zhang. Training with harnesses: On-policy harness self-distillation for complex reasoning, 2026b. URL https://arxiv.org/abs/2605.08741.

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. Rosd: Reflective on-policy self-distillation for language model reasoning across domains, 2026c. URL https://arxiv.org/abs/2605.28014.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return, entropy-eliciting explore, 2025. URL https://arxiv.org/abs/2507.07017.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models, 2024. URL https://arxiv.org/abs/2402.11890.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

Amir Zur, Atticus Geiger, Ekdeep Singh Lubana, and Eric Bigelow. Are language models aware of the road not taken? token-level uncertainty and hidden state dynamics, 2025. URL https://arxiv.org/abs/2511.04527.

A Appendix

A.1 Extended Related Works

A.1.1 More Self-Distillation Related Works

Recent work has studied several failure modes of self-distillation, including reasoning collapse (Kaur et al., 2026; Peng et al., 2026), epistemic verbalization (Kim et al., 2026b), diversity collapse (Nicolicioiu et al., 2026), and teacher unrecoverability under flawed prefixes (Chakraborty et al., 2026). Other approaches strengthen the teacher signal through multi-teacher consensus (Jin et al., 2026b; Huang et al., 2026a), truncated privileged information (Han et al., 2026), targeted hints (Yeo et al., 2026; Liu et al., 2026a), teacher-entropy gating (Ke et al., 2026), and region-specific training (Yeo et al., 2026; Zhao et al., 2026c). Related work also shifts training away from the student distribution toward the teacher distribution (Agrawal et al., 2026) or an interpolated distribution between the two (Chakraborty et al., 2026). Concurrent work (Jiang & Huang, 2026) studies high-entropy tokens in self-distillation, but does not analyze trace content or the relationship between offline and online rollout policies, and evaluates only in the multi-turn setting. Finally, other methods modify the objective to incorporate future cumulative student-teacher disagreement rather than relying solely on token-level disagreement (Agrawal et al., 2026; Xu et al., 2026a).

A.1.2 Entropy and Fork Tokens Related Works

Fork tokens are high-entropy positions that often determine subsequent reasoning steps and therefore play a disproportionate role in the trajectory. Recent work has exploited such tokens in GRPO training in several ways. Wang et al. (2025); Yang et al. (2025) focus learning on high-entropy tokens, while Cheng et al. (2025) incorporates an explicit entropy reward into the advantage and Cui et al. (2025) introduces a covariance term between advantage and entropy. Other approaches instead encourage lower entropy during training (Prabhudesai et al., 2025; Gao et al., 2025). Selective PPO clipping and entropy-aware KL regularization have also been explored (Wang et al., 2026b; Su et al., 2026b). Several methods perform fork-token-aware sampling by branching into multiple continuations from high-entropy positions (Hou et al., 2025; Zheng et al., 2025; Liu et al., 2025b; Dong et al., 2025), while other work uses fork tokens for inference-time steering (Zur et al., 2025).

A.1.3 Offline LLM Training Related Works

A common view in LLM post-training is that online methods generally outperform offline alternatives (Tajwar et al., 2024; Shenfeld et al., 2025; Chu et al., 2025). Offline post-training methods broadly include preference-based objectives such as Direct Preference Optimization (DPO) (Rafailov et al., 2024) and supervised fine-tuning (SFT).

Numerous DPO variants have been proposed to improve entropy control (Omura et al., 2025), robustness to label noise (Chowdhury et al., 2024), robustness to distribution shift (Xu et al., 2026b), and credit assignment (Liu et al., 2025a). In SFT, prior work has studied dataset quality (Zhou et al., 2023; Harada et al., 2025), reward-aware loss reweighting (Wu et al., 2026b), importance-sampling corrections (Qin & Springenberg, 2025), and alternatives to the standard log-likelihood objective (Li et al., 2026a). A substantial body of work has also explored selective token masking in SFT, using signals such as perplexity (Lin et al., 2025; Wu et al., 2025; Wang et al., 2026a), token substitution (Ruan et al., 2025), and entropy (Liu et al., 2026b; Chen et al., 2026).

Recent work has also explored prefix-based learning, in which the model generates a completion conditioned on an off-policy solution prefix. This approach has been studied in both RLVR (Qu et al., 2026; Setlur et al., 2026; Xi et al., 2024; Huang et al., 2026b; Zhang et al., 2025) and external-teacher distillation (Liao et al., 2026).

A.2 Hyperparameters

We use the following hyperparameters for our results unless specified:

Table 5: Training hyperparameters used across experiments unless otherwise specified.

Hyperparameter	Value
Effective batch size	256
Learning rate	5×10^{-6}
Training steps	200
Loss temperature	1.1
Clip value	1×10^{-7}
Divergence	generalized JSD, $\beta = 0$ (forward KL)
Mask keep fraction k	0.1
Generator EMA decay	0.999
LoRA rank (r)	64
LoRA α	128
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
Max completion length (training)	1024
Rollout sampling temperature	1.1 (online and EMA arms)

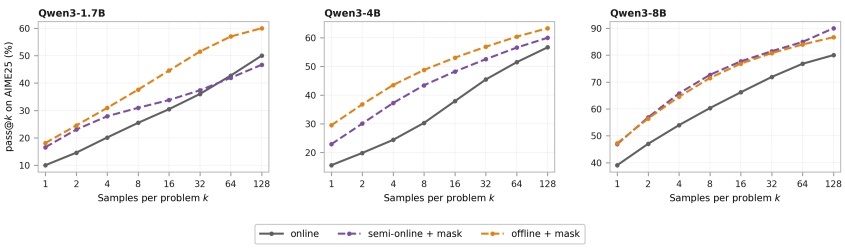

Figure 7: Pass@k on AIME25 at checkpoint 100 using 128 samples per problem. Online OPSD is included for reference. Both masked methods continue to improve as k increases, showing that additional samples solve more problems over the evaluated range.

Evaluation. For math, we use a 16,384-token generation budget with temperature 1.0 and top-p 0.8. AIME24 and AIME25 are evaluated with avg@12, while OlympiadBench uses avg@2. For Knights and Knaves, we evaluate on 1,000 held-out puzzles with the same 16,384-token budget.

A.3 Pass@k

Because reduced predictive entropy could also limit variation across sampled solutions, we examine pass@k on AIME25 in Figure 7. For both masked configurations and at every model size, pass@k continues to improve as more samples are considered, even at $k = 128$. The models therefore continue to find additional correct answers with more sampling over the evaluated range instead of exhibiting diversity collapse.

A.4 Clipping and Epistemic Verbalization

Kim et al. (2026b) report that self-distillation can make model generations verbally overconfident where the model starts using highly assertive words such as *obviously*, *clearly*, or *indeed*. Their experiments use an unclipped distillation objective. We find that in our experiments, it is not the case that our trained models become assertively overconfident, and we attribute this to the clipping mechanism in OPSD. We investigate whether clipping affects this behavior by sweeping ϵ in a Qwen3-4B offline OPSD run while holding the remaining training configuration fixed. For each setting, we count the frequency of two groups of expressions in the model's generations, measured per 1,000 words:

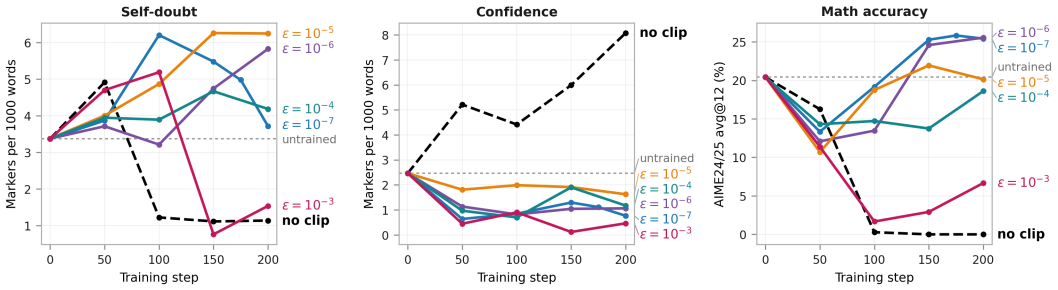

Figure 8: Effect of the distillation clip in offline Qwen3-4B training through step 200. Without clipping, confidence-associated words increase to 3.3× their base frequency, self-doubt expressions fall to roughly one third of base, and accuracy collapses to zero. Clips from 10^{-7} to 10^{-4} reverse these trends, while the looser 10^{-3} setting also fails to train effectively, with accuracy collapsing and both word counts ending below base.

Self-doubt: *maybe, perhaps, possibly, probably, likely, might, could, seems, seem, appears, apparently, presumably, roughly, approximately, suspect, doubt, unclear, unsure, wait, hmm, actually, check, verify, confirm*

Confidence: *clearly, obviously, indeed, must, should*

Without clipping, confidence-associated words increase from 2.5 to 8.1 per 1,000 words by step 200, while self-doubt expressions fall from 3.4 to 1.1. Generation length also decreases sharply, from roughly 1,900 words to 400, and accuracy collapses to 0.3% by step 100 and 0.0% from step 150 onward. This is consistent with the findings of Kim et al. (2026b). With clips between 10^{-7} and 10^{-4}, self-doubt expressions are preserved or increase, while confidence-associated words generally remain suppressed relative to the unclipped run. Accuracy varies across clip values, with the tighter settings performing best in this sweep.

A looser clip of $\epsilon = 10^{-3}$ also fails to train effectively: accuracy falls to 1.7% by step 100 and is 6.7% at step 200. Overall, these results show the loss of uncertainty does not necessarily occur in self-distillation.

A.5 High Entropy Words

We find the highest entropy words in the offline trace, as ranked by the Qwen3-8B untrained model. The list of words can be found in figure 9. As seen in the word cloud, these words are mostly the start of reasoning clauses or sentences where the model will state what step it wants to take next.

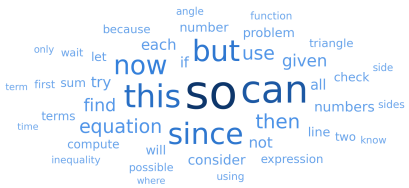

Figure 9: High-entropy words identified by the untrained Qwen3-8B on 1,000 offline traces. We visualize the top 10% highest-entropy token positions in the offline traces. Word size indicates how frequently each word appears among these positions.

A.6 Temperature

Figure 10 sweeps temperature from 0.7 to 1.2 for Qwen3-8B. Offline OPSD is consistently stronger and less sensitive to temperature, varying by 5.1 points across the sweep compared with 11.4 points for online OPSD. Across the tested range, offline performance remains comparatively stable, while online performance varies more substantially with temperature.

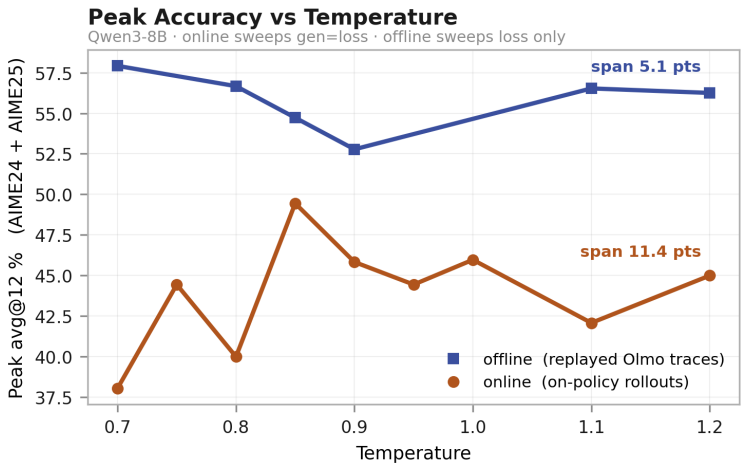

Figure 10: Peak accuracy of Qwen3-8B across temperatures from 0.7 to 1.2. For offline OPSD, the training is on the fixed Olmo trace pool. Offline OPSD outperforms online OPSD across the entire temperature range and is substantially less sensitive to the choice of temperature. Due to compute constraints, we train with effective batch size 32 for these experiments.

A.7 Math Results by Benchmark

Table 2 reports the mean across AIME24, AIME25, and OlympiadBench, while Table 6 reports the individual benchmark results. Results are grouped by checkpoint. *Peak* denotes the checkpoint with the highest mean across the three benchmarks, matching the value reported in Table 2; all benchmark scores in the Peak block are taken from that same checkpoint.

Table 6: Accuracy on AIME24, AIME25, and OlympiadBench at three checkpoints. AIME24 and AIME25 use avg@12 over 30 problems each, while OlympiadBench uses avg@2 over 675 problems. Within each checkpoint group, we report the three benchmark scores, their mean, and the change in mean accuracy relative to the untrained base model of the same size (Δ). *Peak* denotes the checkpoint with the highest mean across the three benchmarks; all benchmark values in that group are taken from that same checkpoint. Step 100 and step 200 are fixed checkpoints shared across all arms. Best performance in each column is shown in bold and runner-up is underlined, excluding the untrained base.

	Peak					Step 100					Step 200				
	A24	A25	Olymp	Mean	Δ	A24	A25	Olymp	Mean	Δ	A24	A25	Olymp	Mean	Δ
Qwen3-1.7B															
untrained base	12.8	10.8	39.5	21.0	–	–	–	–	–	–	–	–	–	–	–
online OPSD	14.2	12.5	41.1	22.6	+1.6	9.7	10.8	35.6	18.7	−2.3	12.5	14.4	37.9	21.6	+0.6
online OPSD + mask	15.6	13.9	42.5	24.0	+3.0	2.2	0.8	13.7	5.6	−15.4	0.3	0.0	5.0	1.8	−19.2
EMA OPSD	15.6	11.9	42.7	23.4	+2.4	14.4	8.9	38.5	20.6	−0.4	13.3	11.4	38.3	21.0	+0.0
EMA OPSD + mask	15.3	16.4	41.6	24.4	+3.4	11.9	17.2	41.0	23.4	+2.4	9.4	12.5	35.0	19.0	−2.0
offline OPSD	12.8	13.1	39.3	21.7	+0.7	12.5	9.2	40.5	20.7	−0.3	11.9	8.1	41.1	20.4	−0.6
offline OPSD + mask	**20.6**	**17.8**	**46.5**	**28.3**	**+7.3**	**20.6**	**17.8**	**46.5**	**28.3**	**+7.3**	**20.6**	**18.3**	**44.6**	**27.8**	**+6.8**
Qwen3-4B															
untrained base	22.2	18.6	51.5	30.8	–	–	–	–	–	–	–	–	–	–	–
online OPSD	23.9	21.7	51.3	32.3	+1.5	22.2	15.6	49.2	29.0	−1.8	22.2	18.6	51.4	30.7	−0.1
online OPSD + mask	26.4	21.7	51.8	33.3	+2.5	13.6	6.7	41.6	20.6	−10.2	4.7	3.9	29.5	12.7	−18.1
EMA OPSD	23.6	22.2	51.0	32.3	+1.5	20.3	19.2	49.6	29.7	−1.1	23.6	22.2	51.0	32.3	+1.5
EMA OPSD + mask	26.9	27.2	51.7	35.3	+4.5	21.7	20.8	50.4	31.0	+0.2	26.9	27.2	51.7	35.3	+4.5
offline OPSD	27.2	25.3	55.3	35.9	+5.1	**30.0**	23.3	53.3	35.5	+4.7	23.3	26.9	54.1	34.8	+4.0
offline OPSD + mask	**31.4**	**31.4**	**55.9**	**39.5**	**+8.7**	26.4	**30.6**	**55.7**	**37.5**	**+6.7**	**29.7**	**29.7**	**56.2**	**38.6**	**+7.8**
Qwen3-8B															
untrained base	28.1	21.7	53.1	34.3	–	–	–	–	–	–	–	–	–	–	–
online OPSD	50.6	38.9	60.7	50.0	+15.7	43.3	36.7	61.2	47.1	+12.8	41.9	33.1	59.0	44.7	+10.4
online OPSD + mask	44.4	36.4	57.3	46.0	+11.7	35.3	33.9	55.7	41.6	+7.3	34.7	31.4	54.4	40.2	+5.9
EMA OPSD	56.4	36.1	60.3	50.9	+16.6	56.4	36.1	60.3	50.9	+16.6	50.6	37.8	61.4	49.9	+15.6
EMA OPSD + mask	59.4	49.7	65.0	58.1	+23.8	53.6	43.3	61.1	52.7	+18.4	59.4	49.7	65.0	58.1	+23.8
offline OPSD	55.6	46.1	66.0	55.9	+21.6	**58.3**	**46.9**	62.2	**55.8**	**+21.5**	54.2	46.4	63.8	54.8	+20.5
offline OPSD + mask	**62.2**	**51.1**	**66.4**	**59.9**	**+25.6**	54.2	45.0	**63.1**	54.1	+19.8	**62.2**	**51.1**	**66.4**	**59.9**	**+25.6**