Escaping the KL Agreement Trap in On-Policy Distillation

Haoran Xin[1] **Anhao Zhao**[3,4] **Ying Sun**[1*] **Jin Li**[1] **Xiaoyu Shen**[4] **Hui Xiong**[1,2*]
[1]Thrust of Artificial Intelligence,
The Hong Kong University of Science and Technology (Guangzhou)
[2]Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong SAR, China
[3]The Hong Kong Polytechnic University [4]Eastern Institute of Technology, Ningbo
hxin883@connect.hkust-gz.edu.cn sunyinggilly@gmail.com xionghui@ust.hk
anhao.zhao@connect.polyu.hk jslijin2015@outlook.com xyshen@eitech.edu.cn

Abstract

On-policy distillation (OPD) provides dense token-level supervision by asking a teacher to score student-generated rollouts. However, when the student drifts into an unrecoverable prefix, the teacher may locally agree with the degraded state, producing low reverse KL but little corrective training signal. We identify this persistent regime as a *low-KL agreement trap*. Further analyses show that tokens during and after such traps produce less useful supervision signals. We propose KAT (*KL Agreement Trap Termination*), an online OPD termination rule that detects persistent low-KL agreement with a dynamic training-adaptive threshold. By filtering weak supervision from degenerate agreement, KAT improves avg@k accuracy by 2.66% and pass@k by 3.43% across four mathematical benchmarks, while reducing average rollout length by 59.73%. We release our code at xhran2010/KAT.

1 Introduction

Post-training (Gong et al., 2026; Kumar et al., 2025) has become a decisive stage in the modern large language model (LLM) pipeline, where capabilities such as reasoning (Chen et al., 2025; Guo et al., 2025; Kumar et al., 2025; Zhang et al., 2026c; Gong and Sun, 2024; Xin et al., 2025) and tool use (Qi et al., 2025; Wang et al., 2025) are largely shaped. A central challenge in this stage is how to transfer the capabilities of strong but expensive models to smaller, deployable ones. LLM distillation (Xu et al., 2024; Zhang et al., 2026b) addresses this by using teacher models to provide richer supervision than ground-truth labels alone, improving smaller models' reasoning ability.

Within this landscape, on-policy distillation (OPD) (Li et al., 2026; Lu and Lab, 2025; Agarwal et al., 2024) has recently emerged as a promising recipe for training students on states closer to their own inference-time behavior, and now appears in the post-training pipelines of several flagship models (DeepSeek-AI, 2026; Xiao et al., 2026; Yang et al., 2025). At its core, OPD lets the student sample its own rollouts and asks the teacher to score every visited state with a dense, token-level signal. This mitigates the exposure bias that off-policy distillation (Kim and Rush, 2016) typically suffers from, while yielding far higher sample efficiency than sparse outcome-reward reinforcement learning (Shao et al., 2024; Schulman et al., 2017).

However, existing OPD methods often treat the full student rollout as useful supervision. We identify an underexplored failure mode: as rollouts progress, the student may drift into a corrupted prefix, such as committing an early reasoning error or entering a repetitive degeneration pattern, as illustrated in Figure 1. Conditioned on this degraded state, the teacher's token-level distribution may carry little corrective information, instead producing continuations that are plausible under the erroneous prefix and therefore close to the student's distribution (Lu and Lab, 2025). In this case, low KL can reflect local teacher–student agreement in a corrupted state rather than progress toward a correct solution. This gives low teacher–student KL two contrastive identities: **benign agreement**, where the student follows a correct reasoning path, and **degenerate agreement**, where the teacher locally follows a corrupted prefix with little corrective signal, providing weak training signal and wasting computation on uninformative tokens. This makes the central challenge distinguishing when low KL marks degenerate supervision that can misdirect and dilute OPD updates.

Our key observation is that degenerate agreement leaves a temporal signature beyond token-level low KL: after a certain rollout depth, the teacher–student KL can remain low across consecutive local regions. We call this persistent regime a **low-KL agreement trap**. This persistence is

*Corresponding authors.

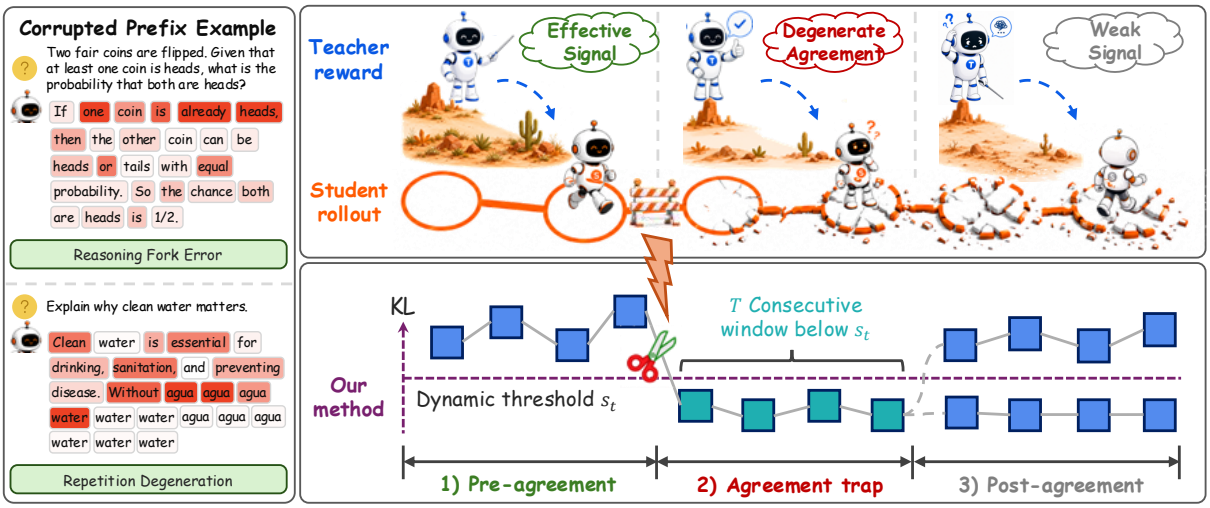


Figure 1: Overview of KAT. A student rollout can drift into a corrupted prefix, e.g., an incorrect reasoning fork or repetitive degeneration, where the teacher locally agrees with the degraded state. Darker blocks denote higher KL. Sustained low KL indicates a low-KL agreement trap, after which the teacher provides weak corrective supervision. KAT reuses OPD's reverse KL, tracks a sliding-window statistic with a training-adaptive threshold, and terminates rollouts once persistent low-KL agreement is detected, focusing updates on informative pre-trap tokens.

crucial for distinguishing degenerate agreement from benign agreement, since both can exhibit low KL locally, but only the former indicates sustained agreement on a corrupted prefix. Our analyses show that supervision in the agreement region is weak, and the subsequent post-agreement suffix does not recover effective teacher signal. Thus, once a rollout enters the agreement trap, continuing generation is unlikely to provide useful supervision and may dilute earlier learning signals.

Building on this diagnosis, we propose **KAT** (*KL Agreement Trap Termination*), a simple training-time rollout termination method for OPD (Fig. 1). KAT reuses the teacher–student reverse KL already computed by OPD and monitors whether it remains low across consecutive local regions. Once this persistent low-KL pattern is detected, KAT terminates the rollout online. The stopping threshold is dynamic, calibrated from recent rollout statistics and adapted to the evolving KL scale throughout training. By removing suffix tokens after degenerate agreement, KAT shifts the effective OPD training signal toward more informative prefixes and improves rollout efficiency without introducing a new loss, auxiliary model, or more than $O(1)$ bookkeeping per token.

Empirically, across four mathematical benchmarks and two student scales, KAT improves avg@k accuracy by 2.66% and pass@k by 3.43%, while reducing average rollout length by 59.73%. Particularly, it also outperforms random termination and fixed-prefix truncation, showing that low-KL agreement traps provide an effective online signal for filtering weak supervision rather than merely shortening rollouts. Our contributions are:

- We identify low-KL agreement traps in OPD, where persistent teacher–student agreement indicates weak rather than useful supervision.
- We analyze that supervision remains low-quality after entering the trap, including in the post-agreement suffix.
- We propose KAT, an adaptive online termination rule that removes uninformative suffixes without changing the OPD objective.

2 Preliminaries

2.1 Notation

Let x denote a prompt and $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T)$ a student rollout sampled from the student policy $\pi_\theta(\cdot \mid x)$. We write $h_t = (x, \hat{y}_{<t})$ for the prefix at step t, and use $p_t(v) = \pi_\theta(v \mid h_t)$ and $q_t(v) = \pi_T(v \mid h_t)$ for the student and teacher next-token distributions, respectively. The token-level reverse KL divergence is

$$d_t = D_{\mathrm{KL}}(p_t \,\|\, q_t) = \sum_{v \in \mathcal{V}} p_t(v) \log \frac{p_t(v)}{q_t(v)}. \quad (1)$$

2.2 On-Policy Distillation Objective

We adopt the standard OPD objective (Agarwal et al., 2024; Li et al., 2026):

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x,\hat{y}\sim\pi_\theta(\cdot|x)}\Bigg[\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x, \hat{y}_{<t}) \,\big\|\, \pi_T(\cdot \mid x, \hat{y}_{<t})\big)\Bigg].$$

The expectation is taken over student-generated trajectories: supervision is dense (every generated token receives a teacher signal) and is anchored on the student's own state distribution. This combination is what gives OPD its sample-efficiency advantage over outcome-only RL methods, and its robustness advantage over off-policy KD that trains only on a static dataset.

3 Low-KL Agreement in OPD Rollouts

We motivate KAT by analyzing student rollouts during OPD training and identifying a recurring low-KL agreement pattern (§3.1): at certain trajectory depth, the student and teacher often maintain sustained low reverse KL. This pattern naturally partitions each rollout into three phases: *pre-agreement*, *agreement*, and *post-agreement*. We study these phases from an optimization perspective (§3.2), using gradient projections to assess their contribution to effective learning, and further validate the teacher supervision through a token-level visualization (§3.3). Together, these analyses show that the agreement and post-agreement phases provide little useful supervision or parameter movement, motivating our proposed method.

3.1 Low-KL Dynamics

Setup. We sample $N = 100$ student rollouts at each of two training stages—*early* (within the first 10% of optimizer steps) and *late* (near the end of training). For each rollout, we record the per-token reverse KL d_t and visualize it as a heatmap in Fig. 2a, with rollout position on the horizontal axis and rollout index on the vertical axis; color intensity encodes $\log d_t$. Low-KL regions are marked using the calibrated threshold adopted by KAT.

Observations. The heatmap reveals three robust phenomena. First, **low-KL regions are prevalent**: more than 60% of rollouts exhibit at least one extended low-KL segment. Second, **early-training rollouts enter low-KL earlier**: the median onset position of the first low-KL segment in early-stage rollouts is markedly smaller than in late-stage rollouts, indicating that an under-trained student more quickly drifts into agreements with the teacher. Third, **low-KL regions tend to be followed by weaker KL signals**: after a rollout enters a low-KL segment, the subsequent reverse KL often remains lower than its pre-segment level rather than fully returning to earlier high-KL behavior. This trend is especially pronounced in early training, where many rollouts exhibit extremely small KL values after entering such regions, indicating that the token-level OPD signal has largely lost update strength. This observation suggests that low-KL regions may mark a qualitative change in the usefulness of later rollout tokens, motivating a closer analysis of the optimization signal they provide. A complementary teacher-recoverability analysis in Appendix C supports this interpretation: recoverability drops by 15.9 percentage points across detected trap boundaries, compared with 5.6 points across position-matched permuted boundaries, yielding a 10.3-point difference-in-differences.

3.2 Gradient Geometry

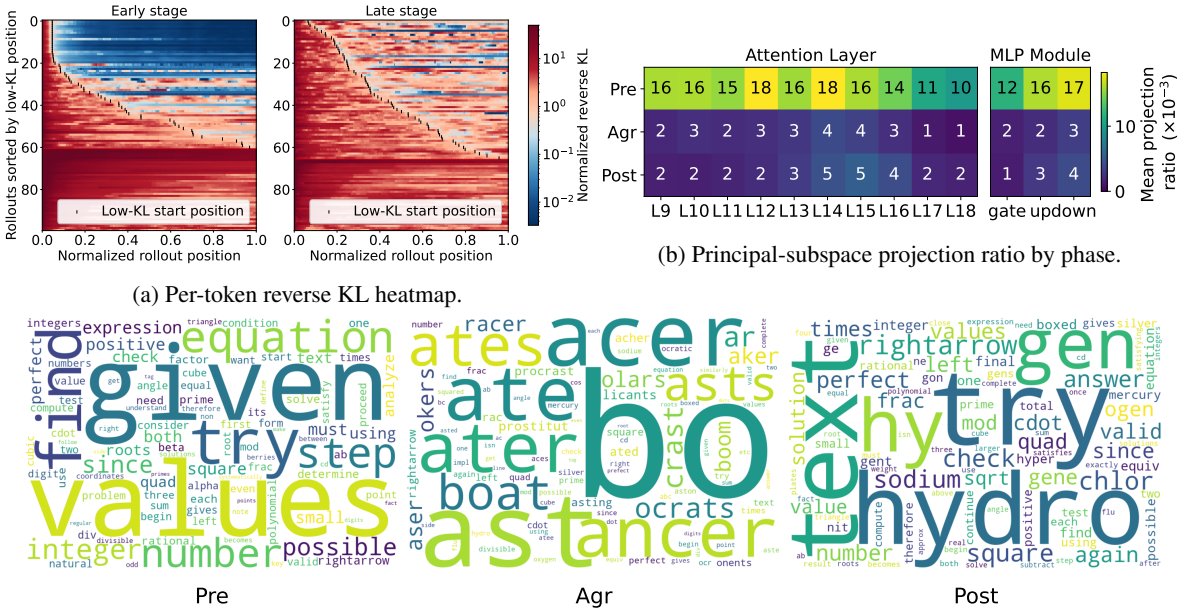
The observations of §3.1 induce a three-phase decomposition of each rollout: *pre-agreement* positions before the first extended low-KL segment, *agreement* positions inside the segment, and *post-agreement* positions after it. We now ask whether the OPD updates from these phases align with the principal parameter directions during training.

Principal Update Subspace via Top-k SVD. Let θ_0 denote the parameters at the start of training and $\theta^\star$ that of the final analyzed checkpoint. Inspired by (Cai et al., 2026), for each layer, we form the net learned update $\Delta W = \theta^\star - \theta_0$ and compute its rank-k truncated SVD (Koren et al., 2009),

$$\Delta W \approx U_k \Sigma_k V_k^\top. \tag{2}$$

The top-k singular directions define a principal update subspace: the dominant directions along which the model parameters changed over training. Projection onto this subspace therefore provides a proxy for how much an update aligns with the main parameter movement learned by the model.

Projection Magnitude. At a selected analysis checkpoint θ_k, and for each rollout phase $\phi \in \{\mathrm{pre}, \mathrm{agr}, \mathrm{post}\}$, we form the phase-conditional gradient $g_k^{(\phi)}$ by accumulating per-token OPD gradients only at positions belonging to phase ϕ on a



(a) Per-token reverse KL heatmap.

(b) Principal-subspace projection ratio by phase.

(c) Teacher top-token word clouds by phase.

Figure 2: Evidence for low-KL agreement and weakened update alignment in OPD rollouts. **(a)** Per-token KL at early and late training stages, with rollouts sorted by the first low-KL position. Low-KL regions appear in many rollouts and occur earlier in early training. **(b)** OPD gradients in respective rollout stages are projected onto the principal update subspace. Pre-agreement gradients align much more strongly with the principal update directions than agreement and post-agreement gradients, suggesting that low-KL agreement marks a transition to less effective optimization signal. **(c)** Word clouds of the teacher's high-probability tokens in each phase: pre-agreement is dominated by reasoning-related vocabulary, while (post-)agreement shifts toward local fragments.

held-out probe set. We then report the fraction of $g_k^{(\phi)}$'s squared norm that lies inside the layer's principal update subspace,

$$\rho_k^{(\phi)} = \frac{\|P_{U_k} g_k^{(\phi)}\|^2}{\|g_k^{(\phi)}\|^2}, \tag{3}$$

where $P_{U_k} = U_k U_k^\top$ projects onto the principal update subspace. A larger $\rho_k^{(\phi)}$ indicates that phase ϕ contributes gradient signal more aligned with the model's principal parameter update, while a smaller value indicates weaker alignment.

Observations. Figure 2b shows a clear ordering across phases. The *pre-agreement* phase attains the largest projection ratio; the *agreement* and *post-agreement* phases are substantially smaller. This comparison is especially informative for attention and final MLP parameters, which carry task-relevant updates in OPD training. The result suggests that once a rollout enters a low-KL agreement region, the subsequent gradient signal becomes less aligned with the principal directions of parameter movement; moreover, the post-agreement segment does not fully recover this alignment even when the per-token KL increases again. Thus, agreement and post-agreement tokens may provide less efficient supervision than pre-agreement tokens, motivating us to examine whether continuing to roll out through these regions is worthwhile.

3.3 Teacher Vocabulary Shift

To further understand what supervision the teacher provides in different phases, we visualize the tokens to which the teacher assigns high probability.

Setup. For each rollout position, we take the teacher's top-5 tokens under $\pi_T(\cdot \mid h_t)$, filter out special characters and non-content artifacts, and aggregate the remaining tokens separately over three phases. We visualize the resulting empirical token distributions as word clouds (Ji et al., 2025) in Fig. 2c.

Observations. The word clouds show a clear shift in the teacher's high-probability vocabulary across phases. In the *pre-agreement* phase, the teacher frequently assigns mass to reasoning-related tokens such as mathematical quantities, equation terms, and discourse markers, suggesting that the supervision still reflects useful intermediate reasoning. In the *agreement* phase, the distribution becomes more concentrated on local surface con-

tinuations and problem-specific fragments, with fewer tokens that clearly indicate stepwise reasoning. The *post-agreement* phase partially recovers content words, but its vocabulary remains noticeably different from the pre-agreement phase and contains more generic continuation tokens. Together with the analysis in §3.2, this visualization suggests that low-KL regions are not merely low-disagreement regions; they also correspond to a qualitative change in the teacher's token-level supervision, motivating us to consider whether these later tokens are worth generating and training on.

4 Method: Online Termination with Low-KL Agreement

Based on the analysis in §3, we propose KAT, an online rollout termination method for OPD. KAT uses the teacher–student reverse KL already computed during OPD to detect sustained low-KL agreement at sufficient rollout depth. It calibrates a threshold from recent rollouts, monitors a sliding-window KL statistic during generation, and terminates the rollout once the statistic remains low for consecutive windows. The truncated suffix is not generated and contributes no OPD supervision.

4.1 Local KL Statistic

Consider a prompt x and a student rollout $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T)$. At token position t, let $h_t = (x, \hat{y}_{<t})$ denote the current prefix, and let $p_t = \pi_\theta(\cdot \mid h_t)$ and $q_t = \pi_T(\cdot \mid h_t)$ denote the student and teacher next-token distributions, respectively. We measure their local disagreement by the reverse KL d_t in Eq. (1). Directly thresholding d_t at a single token is unreliable, since token-level KL can fluctuate due to punctuation, formatting tokens, or local lexical choices. We therefore track a local average over a window of W tokens:

$$z_t \;=\; \frac{1}{W} \sum_{i=t-W+1}^{t} d_i, \qquad t \geq W, \tag{4}$$

where W is the sliding-window size and z_t summarizes the recent teacher–student disagreement around position t. A small z_t indicates that the teacher and student have remained close over a local region, which is the signal used by KAT to detect sustained low-KL agreement.

4.2 Adaptive Threshold Calibration

The scale of KL can vary substantially across training, so a fixed threshold is difficult to choose a priori. KAT therefore calibrates the low-KL threshold from recent rollouts. For each completed rollout $\hat{y}$, we record the smallest sliding-window KL value:

$$m(\hat{y}) \;=\; \min_{t \geq W} z_t. \tag{5}$$

Intuitively, $m(\hat{y})$ summarizes how low the rollout's local teacher–student disagreement becomes once it is eligible for early stopping. During the first K optimizer steps, KAT runs in a warmup mode: no rollout is terminated early, and each full rollout contributes its minimum score $m(\hat{y})$ to a FIFO buffer $\mathcal{B}$. The buffer stores the most recent B minimum scores, providing a rolling estimate of the current low-KL scale under the evolving student policy.

After warmup, at the start of each step $k > K$, KAT sets the stopping threshold from the buffer:

$$s_k \;=\; Q_{1-\eta}(\mathcal{B}), \tag{6}$$

where $Q_{1-\eta}$ denotes the $(1-\eta)$-quantile and η controls the aggressiveness of stopping. Smaller η yields a more permissive rule: for example, with $\eta = 10\%$, the threshold is the 90% quantile of recent rollout minima, so a rollout is considered low-KL when its local window score falls below this more permissive threshold. This calibrated threshold makes the rule data-driven and adaptive to the changing KL scale during optimization, avoiding hand-tuning an absolute KL cutoff.

4.3 Online Trigger and Termination

After warmup, KAT applies the calibrated threshold online during rollout. Let k denote the current optimizer step, K the number of warmup steps, z_t the sliding-window KL statistic ending at position t, and s_k the threshold calibrated from the buffer. We also introduce an exemption length L_0, which excludes the initial KL region from the stopping decision. A rollout becomes eligible for termination only when $k > K$ and $t \geq L_0 + W$. This exemption is important because early in a rollout, reverse KL often drops rapidly from a high-disagreement regime to a lower one while the teacher may still provide meaningful corrective signal. KAT therefore focuses on sustained low-KL agreement that appears after this early transition.

For each eligible rollout, we maintain a counter c for consecutive low-KL windows. The counter is initialized to zero for each rollout and updated as

$$c \leftarrow \begin{cases} c+1, & z_t < s_k, \\ 0, & z_t \geq s_k. \end{cases} \tag{7}$$

Here, $z_t < s_k$ means that the window ending at position t is classified as low-KL. We define T as the consecutive trigger length: the number of consecutive low-KL windows required before termination is confirmed. This requirement prevents the rule from firing on isolated KL dips and makes termination depend on a sustained low-KL pattern.

When $c \geq T$ at position t, the trigger is confirmed. Since this confirmation occurs only after observing T consecutive low-KL windows, the effective truncation point is set by backtracking to the start of the first low-KL window in this consecutive run. No further student tokens are generated for that prompt, and tokens after the effective truncation point do not contribute OPD updates.

Remarks We defer the full KAT pseudocode and the streaming teacher-scoring implementation to Appendix D. The added online state is $O(1)$ per token, and the teacher signal already required by OPD is reused without additional rollouts.

5 Experiments

5.1 Setup

Models. We evaluate two teacher–student pairs. The teacher is Qwen3-8B, and the students are Qwen3-1.6B and Qwen3-4B, respectively. For each pair, the student is trained with OPD under the same training recipe. The implementation details are described in Appendix E.

Benchmarks and Metrics. We evaluate mathematical reasoning performance on AMC, MATH500, MinervaMath, and AIME24. We report avg@k and pass@k accuracy on each benchmark, as well as the average across benchmarks. Following common evaluation budgets, we use $k = 32$ for AIME24, and $k = 8$ for the remaining benchmarks.

Baselines. We compare KAT-OPD against the following baselines. **Teacher** and **Student** report the zero-shot performance of the original teacher and student models, respectively. **OPD** is the standard on-policy distillation baseline. We also compare against two OPD variants that reduce rollout length without using the proposed low-KL agreement signal. **Random Termination** stops each rollout at a randomly sampled position, controlling for the effect of shortening rollouts without selecting specific low-KL regions. **Fixed Prefix** caps the maximum response length at 1024, providing a simple length-based truncation baseline. **KAT-OPD** is our full method, which terminates rollouts online when sustained low-KL agreement is detected.

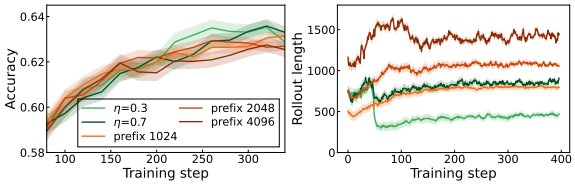


Figure 3: Training dynamics of KAT versus fixed-prefix OPD with varying lengths. KAT reaches higher accuracy earlier while maintaining substantially shorter effective rollout lengths, indicating that adaptive termination of agreement traps improves the accuracy–compute trade-off over uniform prefix truncation.

5.2 Main Results

Table 1 reports avg@k and pass@k accuracy across benchmarks. Overall, KAT-OPD improves both accuracy and rollout efficiency over standard OPD. Averaged over the two student settings, KAT increases avg@k from 30.27 to 31.08, a relative improvement of 2.66%, and improves pass@k from 54.74 to 56.62, a relative gain of 3.43%. Meanwhile, the average rollout length is reduced from 1480.0 to 596.0, saving 59.73% of rollout tokens.

The comparison with length-reduction baselines further highlights the advantage of KAT. Standard OPD keeps the full rollout and can therefore train on late tokens where the teacher signal has become weak. Fixed Prefix truncates every rollout at a fixed length, ignoring the quality and importance of the current trajectory state. Random Termination shortens rollouts at random positions and cannot distinguish useful reasoning prefixes from low-quality suffixes. In contrast, KAT makes prefix-dependent online decisions based on sustained low-KL agreement between the student and teacher. This allows it to preserve useful early supervision while terminating rollouts once the teacher signal becomes less informative, explaining why it achieves higher accuracy with substantially fewer rollout tokens.

5.3 Analysis

Training Dynamics Analysis. Figure 3 compares KAT with fixed-prefix OPD variants of different prefix lengths, reporting validation accuracy and effective rollout length. For both quantile settings, $\eta = 0.3$ and $\eta = 0.7$, KAT reaches the converged accuracy region earlier and achieves higher final accuracy, while maintaining substantially shorter effective rollouts. This supports our central motivation: useful OPD supervision is con-

Method	AMC		MATH500		Minerva		AIME24		Avg		Rollout Length
	avg@k	pass@k	avg@k	pass@k	avg@k	pass@k	avg@k	pass@k	avg@k	pass@k	
Teacher	58.28	77.11	83.18	94.20	29.23	41.18	27.92	63.33	49.65	68.96	-
Qwen3-1.7B-Base											
Student	10.09	40.96	20.55	65.60	5.38	24.27	1.67	13.33	9.42	36.04	-
OPD	20.33	53.01	41.55	80.00	10.71	30.52	3.33	23.33	18.98	46.72	1013
- *Random Termination*	19.88	46.99	**44.50**	79.20	10.98	28.68	3.54	**33.33**	<u>19.73</u>	47.05	682
- *Fixed Prefix*	20.63	49.40	40.78	78.00	10.39	33.46	4.27	30.00	19.02	<u>47.72</u>	730
KAT-OPD	**21.24**	**56.63**	<u>43.33</u>	**81.40**	**11.49**	**33.82**	**4.48**	<u>30.00</u>	**20.14**	**50.46**	**456**
Qwen3-4B-Base											
Student	14.91	50.60	21.90	74.80	6.34	30.15	5.21	40.00	12.09	48.89	-
OPD	45.48	73.49	<u>78.80</u>	91.20	25.60	43.02	16.35	<u>43.33</u>	<u>41.56</u>	62.76	1947
- *Random Termination*	44.13	68.68	**79.00**	91.40	24.95	45.59	14.38	33.33	40.62	59.75	863
- *Fixed Prefix*	45.19	72.31	78.05	90.60	25.41	44.49	15.88	**46.67**	41.13	**63.52**	987
KAT-OPD	**46.69**	**75.90**	78.23	**91.80**	**26.54**	**46.75**	**16.56**	36.67	**42.01**	<u>62.78</u>	**736**

Table 1: avg@k and pass@k accuracy on four benchmarks for two teacher–student pairs. We compare KAT-OPD against the zero-shot Student/Teacher endpoints, standard OPD, and two length-reduction baselines: Random Termination and Fixed Prefix. The rightmost column reports the average rollout length in tokens. Best per-column results in **bold**, second-best <u>underlined</u>.

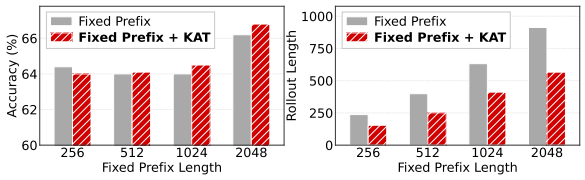


Figure 4: Applying KAT on top of fixed-prefix OPD across prefix lengths $\{256, 512, 1024, 2048\}$ (Qwen3-1.7B-Base). **Left**: accuracy; **Right**: effective rollout length. At moderate-to-large budgets, KAT (hatched) improves accuracy over the fixed-prefix baseline (solid) while shortening rollouts, with the length gap widening as the budget grows.

centrated in informative prefixes, so adaptive termination can improve the accuracy–compute trade-off without preserving uniformly long rollouts.

We also observe that longer fixed prefixes do not monotonically improve training. In particular, the 4096-token prefix underperforms the 2048-token prefix despite exposing the student to more teacher supervision. This suggests that late rollout tokens may provide weak or even harmful signals: once the student has drifted into an unrecoverable reasoning state, the teacher can no longer offer effective corrective guidance. Thus, simply extending rollout length may dilute useful early-token supervision with low-quality suffix supervision.

KAT on Top of Different Prefix Lengths. To disentangle KAT from the choice of prefix budget, we apply it on top of fixed-prefix OPD across prefix lengths $\{256, 512, 1024, 2048\}$ (Fig. 4). At moderate-to-large budgets, KAT improves accuracy while shortening the effective rollout, as it removes uninformative suffix tokens and concentrates supervision on informative prefixes; the length gap also widens with larger budgets, which admit more post-trap suffix to truncate. At the smallest budget (256), however, KAT yields no accuracy gain and even a slight drop, since the short cap already discards most of the low-KL agreement region, and further truncation begins to remove useful early signal as well. This indicates that KAT is complementary to length-based truncation rather than a substitute: a fixed prefix bounds the worst-case length, while KAT adaptively removes the low-KL agreement region within that budget, improving the accuracy–compute trade-off whenever the budget is large enough to contain such a region.

Training Efficiency. Table 2 reports the end-to-end training cost of KAT-OPD compared with standard OPD and the Fixed-Prefix baseline. KAT-OPD consistently reduces both wall-clock time and compute across model scales. On Qwen3-1.7B-Base, it reduces training time from 12.30h to 5.20h (-57.7%) and compute from 1.98 to 1.30 EFLOPs (-34.3%) relative to OPD. On Qwen3-4B-Base, it trains $2.4\times$ faster (11.00h vs. 26.28h) and uses 59.4% fewer EFLOPs (2.30 vs. 5.67). Compared with Fixed Prefix, KAT-OPD remains 23–35%

Method	GPU Hours	EFLOPs
Qwen3-1.7B-Base		
OPD	12.30	1.98
- *Fixed Prefix*	7.96	1.54
KAT-OPD	**5.20**	**1.30**
Qwen3-4B-Base		
OPD	26.28	5.67
- *Fixed Prefix*	14.28	3.04
KAT-OPD	**11.00**	**2.30**

Table 2: End-to-end training cost in GPU hours and EFLOPs. KAT-OPD reduces training time by up to $2.4\times$ and compute by up to 59% over standard OPD, while also outperforming the fixed-prefix baseline.

faster and uses 16–24% fewer FLOPs, showing that adaptive termination is more efficient than a uniform length cap. Together with the accuracy gains in Table 1, these results show that KAT-OPD achieves a better accuracy–compute trade-off.

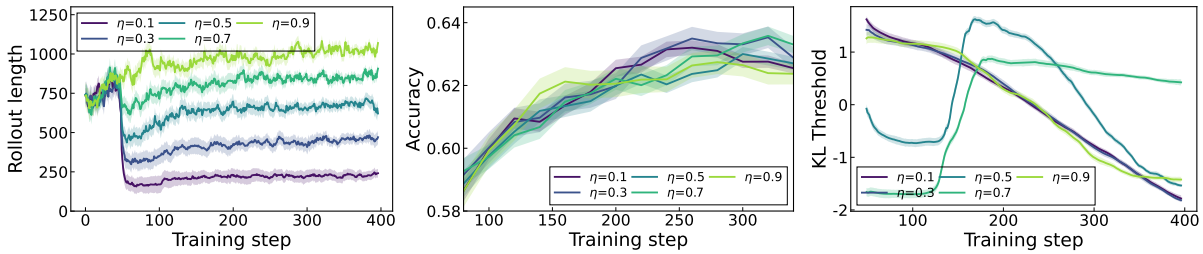
Stage-wise Supervision Ablation. The low-KL agreement trap partitions each rollout into pre-agreement, agreement, and post-agreement stages. To test this decomposition, we retrain OPD by adding different truncated stages back to the KAT training signal. Figure 6 shows that adding either the agreement or post-agreement stage reduces accuracy compared with KAT-OPD, and adding the post-agreement stage causes a larger drop. This indicates that supervision after entering the agreement trap is already weak, and the later post-trap suffix provides even lower-quality signals. These results support KAT's design choice of terminating rollouts at the trap entry and training only on the informative pre-agreement prefix.

Impact of Quantile η. Figure 5 studies how the quantile parameter η affects rollout length, accuracy, and the adaptive threshold. Under our threshold convention, smaller η gives a more permissive threshold, making low-KL windows easier to truncate and thus producing shorter effective rollouts (Fig. 5a). The accuracy curves in Fig. 5b show that smaller η often performs better, suggesting that more aggressive removal of low-quality suffixes can improve the effective training signal. This further supports our main observation that many late rollout tokens provide limited useful supervision. Finally, Fig. 5c shows that the normalized adaptive threshold generally decreases during training as the student becomes closer to the teacher. Around step 100, several settings exhibit a temporary plateau or increase, likely due to training instability enlarging the estimate. This larger threshold can be beneficial in unstable phases by filtering more noisy suffixes, before relaxing as training stabilizes. We additionally sweep the window size W and trigger length T in Appendix B; the default setting $W = 64$ and $T = 4$ achieves the highest accuracy in both sweeps, while the results provide practical guidance on the detection-sensitivity and confirmation-delay trade-offs.

6 Related Work

On-policy Distillation. On-policy distillation (OPD) trains students on self-generated states with dense teacher supervision, reducing off-policy train–test mismatch while improving sample efficiency over sparse reward RL (Agarwal et al., 2024; Zhang et al., 2026d). Recent work studies its mechanisms and limitations, including early alignment with the eventual update trajectory (Cai et al., 2026) and factors such as teacher–student compatibility, shared high-probability tokens, and response-length sweet spots (Li et al., 2026). Other methods refine the OPD objective or teacher signal with adaptive targets, entropy-aware KL, relaxed imitation, or generalized KL constraints (Jang et al., 2026; Jin et al., 2026; Ko et al., 2026; Yang et al., 2026). Efficiency-oriented work such as prefix-only OPD truncates rollouts based on the observation that useful supervision often concentrates in early reasoning prefixes (Zhang et al., 2026a). Complementarily, we identify persistent low-KL agreement on corrupted prefixes as a trajectory-dependent failure mode and use it as an online signal to terminate uninformative suffixes without changing the OPD loss.

Rollout Filtering in RL. Recent RL post-training methods filter generated rollouts because many are redundant, noisy, or harmful. Existing approaches operate at different granularities, including rollout-level sampling or overlong filtering (Yu et al., 2026; Xu et al., 2025), prompt/task-level filtering by learning utility or difficulty (Zheng et al., 2026; Bae et al., 2026), and response-level truncation or confidence-based trace filtering (Fan et al., 2025; Fu et al., 2025). They typically rely on rewards, difficulty, length, or confidence. Complementarily, we filter within OPD trajectories using the teacher–student KL itself, terminating suffixes where persistent low KL indicates weak corrective supervision rather than useful agreement.



(a) Effective rollout length vs. η. (b) Average accuracy vs. η. (c) Normalized adaptive threshold.

Figure 5: Effect of the quantile parameter η on KAT. Smaller η produces a more permissive threshold and therefore shorter effective rollouts (a), which generally yields higher average accuracy (b), supporting the view that many late rollout tokens contribute weak supervision. The normalized adaptive threshold (c) decreases as training proceeds, with a transient plateau near step 100 that lets KAT filter noisy suffixes during unstable phases.

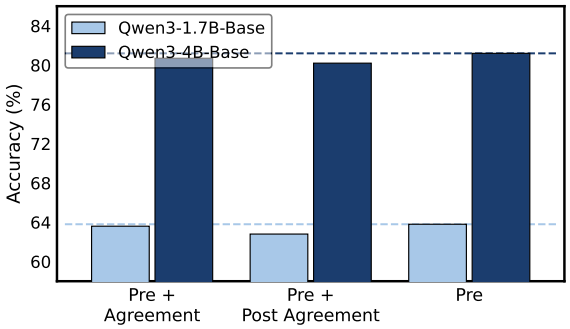


Figure 6: Stage-wise ablation on MATH500. Each rollout is split into pre-agreement, agreement, and post-agreement stages; dashed lines denote KAT-OPD, which trains only on pre-agreement tokens. Adding back any truncated stage reduces accuracy, with post-agreement causing a larger drop, indicating that supervision further deteriorates after the agreement trap.

7 Conclusion

We identified low-KL agreement traps in on-policy distillation, a failure mode where the teacher and student remain locally close on degraded rollout prefixes but provide weak corrective supervision. To address this, we proposed KAT, an online termination rule that reuses the reverse KL already computed in OPD to detect sustained low-KL agreement with a training-adaptive threshold. By truncating uninformative suffixes without changing the OPD objective or adding auxiliary models, KAT improves the performance while substantially reducing rollout length, showing that KAT can make OPD both more effective and efficient.

Limitations

This work focuses on mathematical reasoning tasks. Although the proposed termination rule is simple and model-agnostic, its behavior on broader domains such as code generation, open-ended instruction following, and multi-turn interaction remains to be further studied. In addition, KAT introduces several hyperparameters. While these parameters are intuitive and inexpensive to tune, more systematic guidance could further improve usability across training settings.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No.92370204), in part by the National Key R&D Program of China (Grant No.2023YFF0725001), in part by Guangdong Provincial Key Laboratory of Frontier Basic Science for All-domain Intelligence, in part by the guangdong Basic and Applied Basic Research Foundation (Grant No.2023B1515120057), in part by the Key-Area Special Project of Guangdong Provincial Ordinary Universities (2024ZDZX1007).

References

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

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2026. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 700–719.

Yuchen Cai, Ding Cao, Liang Lin, Chunxi Luo, Xin Xu, Kai Yang, Weijie Liu, Saiyong Yang, Tianxiang Zhao, Guangzhong Sun, and 1 others. 2026. Learning to foresee: Unveiling the unlocking efficiency of on-policy distillation. *arXiv preprint arXiv:2605.11739*.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*.

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

Tiantian Fan, Lingjun Liu, Yu Yue, Jiaze Chen, Chengyi Wang, Qiying Yu, Chi Zhang, Zhiqi Lin, Ruofei Zhu, Yufeng Yuan, and 1 others. 2025. Truncated proximal policy optimization. *arXiv preprint arXiv:2506.15050*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Zheng Gong and Ying Sun. 2024. Graph reasoning enhanced language models for text-to-sql. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2447–2451.

Zheng Gong, Ying Sun, Ping Li, Yi Zheng, and Zhefeng Wang. 2026. Punctuation-steered representation fine-tuning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–8.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. 2026. Stable on-policy distillation through adaptive target reformulation. *arXiv preprint arXiv:2601.07155*.

Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. 2025. A comprehensive survey on self-interpretable neural networks. *Proceedings of the IEEE*.

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

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

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

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and 1 others. 2026. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*.

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

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, and 1 others. 2025. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pages 79791–79821.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, and 1 others. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.

Haoran Xin, Ying Sun, Chao Wang, and Hui Xiong. 2025. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(5):1–33.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. 2025. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

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

Mingxu Zhang, Yuhan Li, Lujundong Li, Dazhong Shen, Hui Xiong, and Ying Sun. 2026b. Sae-fd: Sparse autoencoder feature distillation for continual learning of large language models. *Preprint*, arXiv:2605.25525.

Mingxu Zhang, Dazhong Shen, Qi Zhang, and Ying Sun. 2026c. REAP: Towards effective training-free chemical reasoning with explicit atomic priors. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 2037–2062, San Diego, California, United States. Association for Computational Linguistics.

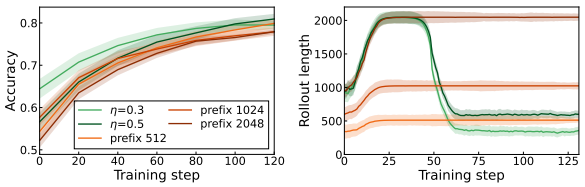
Mingxu Zhang, Huicheng Zhang, Jiaming Ji, Yaodong Yang, and Ying Sun. 2026d. Enhance the safety in reinforcement learning by adrc lagrangian methods. *Preprint*, arXiv:2601.18142.

Haizhong Zheng, Yang Zhou, Brian Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2026. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *Advances in Neural Information Processing Systems*, 38:124321–124346.

Appendix

A Analysis on Qwen3-4B-Base Student

We report the analysis on the larger Qwen3-4B-Base student. Figure 7 reports the training dynamics of KAT against fixed-prefix OPD variants, and Fig. 8 reports the effect of the quantile parameter η on rollout length, accuracy, and the normalized adaptive threshold.

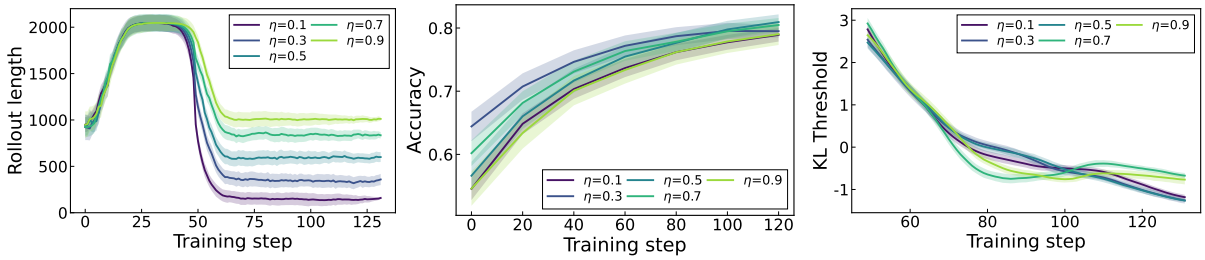


(a) Accuracy over optimizer steps. (b) Effective rollout length over optimizer steps.

Figure 7: Training dynamics of KAT versus fixed-prefix OPD with varying lengths on the Qwen3-4B-Base student. As in the smaller-student setting, KAT reaches the converged accuracy region earlier (a) while maintaining substantially shorter effective rollout lengths (b).

Training dynamics. Figure 7 mirrors the trend observed for the smaller student. KAT converges to its accuracy plateau noticeably earlier than the fixed-prefix variants while operating with a substantially shorter effective rollout length throughout training. The relative ordering of fixed-prefix variants is also consistent: blindly increasing the prefix length does not monotonically improve accuracy, again indicating that uniformly long rollouts contain late tokens whose teacher signal has become weak or even harmful. This reinforces that the gains of KAT are not a quirk of the smaller student but arise from selectively removing low-quality suffix supervision.

Quantile sensitivity. Figure 8 shows that the qualitative effect of η is preserved at 4B scale. More permissive η shortens rollouts (Fig. 8a) and tends to produce equal or higher average accuracy (Fig. 8b), confirming that aggressive removal of low-quality suffix tokens remains beneficial as the student grows. The normalized adaptive threshold in Fig. 8c again drifts downward as the student becomes closer to the teacher, with a similar transient bump in the early training phase that lets KAT filter noisier suffixes while training is still unstable. Together with the 1.7B results, these plots indicate that the behavior of KAT transfers consistently across student scales.



(a) Effective rollout length vs. η.

(b) Average accuracy vs. η.

(c) Normalized adaptive threshold.

Figure 8: Effect of the quantile parameter η on KAT for the Qwen3-4B-Base student. Smaller η again yields shorter effective rollouts (a) and generally higher average accuracy (b), and the normalized adaptive threshold (c) decreases as training proceeds.

B Sensitivity to Window Size and Trigger Length

We further study the two hyperparameters that control trap detection: the sliding-window size W used to compute the local KL statistic and the trigger length T required to confirm consecutive low-KL windows. We sweep each parameter while holding all other settings fixed at their defaults.

Window size W	Avg. rollout length	Acc. (%)
16	360.5	51.8
32	386.6	52.4
64	414.2	**53.4**
128	458.5	52.2

Trigger length T	Avg. rollout length	Acc. (%)
2	413.9	51.8
3	407.4	52.0
4	414.2	**53.4**
5	414.1	52.2

Table 3: Sensitivity to the sliding-window size W and consecutive trigger length T. Each sweep varies one parameter while holding all other settings fixed.

Table 3 shows that small windows can react prematurely to local KL fluctuations, whereas large windows delay detection; the intermediate default $W = 64$ achieves the highest accuracy. Average rollout length is comparatively stable across trigger lengths because KAT backtracks to the first low-KL window, so increasing T mainly delays confirmation rather than changing the retained prefix; the default $T = 4$ again performs best. Together with the quantile analysis in Fig. 5, these results show that W, T, and η respectively control local sensitivity, confirmation evidence, and threshold permissiveness.

C Teacher Recoverability Around Trap Boundaries

Low KL at an isolated position need not indicate degeneration: a transient dip can arise when the teacher and student benignly agree on a valid continuation. Our claim instead concerns persistent low KL across consecutive local regions, which may indicate that the rollout has entered a degraded prefix from which the teacher provides little corrective signal. We test this claim directly by measuring whether the teacher can recover a correct solution around each trap boundary detected by KAT.

Let τ denote a detected trap boundary and $\Delta = 64$ tokens. For the student-generated prefixes ending at $\tau - \Delta$, τ, and $\tau + \Delta$, we sample continuations from the teacher and report the fraction that reach the correct final answer. Because recoverability may decline naturally with rollout depth, we construct a position-matched control by permuting detected boundary positions across rollouts while preserving their marginal position distribution. We then evaluate teacher continuations at the corresponding permuted boundaries, denoted $\tilde{\tau}$.

Boundary	$-\Delta$	0	$+\Delta$
Detected trap (τ)	43.8%	38.4%	27.8%
Permuted control ($\tilde{\tau}$)	36.9%	32.5%	31.3%

Table 4: Teacher recoverability before, at, and after detected trap boundaries and position-matched permuted boundaries. Recoverability is the fraction of teacher continuations that reach the correct final answer.

As shown in Table 4, teacher recoverability decreases from 43.8% to 27.8% across a detected boundary, a 15.9-point drop up to rounding. Around the permuted boundary, it decreases from 36.9% to 31.3%, a 5.6-point drop. The resulting difference-in-differences is 10.3 percentage

points. Thus, the decline in recoverability is substantially sharper around detected trap boundaries than around position-matched controls. This result does not imply that every low-KL event is degenerate; rather, it directly associates the persistent low-KL pattern detected by KAT with degraded prefixes and reduced teacher recoverability.

D Algorithm and Online Teacher Scoring

Algorithm. The full KAT-OPD procedure described in Section 4 is summarized as pseudocode in Algorithm 1. The only additional online state is the rolling sum used to compute the sliding-window statistic z_t and the consecutive-window counter c. Both can be updated incrementally during generation, so the bookkeeping cost is $O(1)$ per token.

Online teacher scoring. Online termination requires the teacher to score student-generated prefixes during rollout, rather than after the full sequence has been generated. We implement this in a streaming manner: as the student generates new tokens, the teacher receives the same growing prefix and produces the corresponding next-token distribution used to compute the reverse KL. To keep this efficient, the teacher also maintains and reuses its KV cache, so each new scoring step only incurs incremental computation instead of recomputing the entire prefix. Thus, KAT reuses the teacher signal required by OPD while enabling termination before uninformative suffix tokens are generated.

E Implementation Details

Models and data. All experiments use models from the Qwen3 family. The teacher is Qwen3-8B,[1] and we consider two students of different scales: Qwen3-1.7B-Base[2] and Qwen3-4B-Base.[3] Training prompts are drawn from the DAPO-Math-17K dataset.[4] For evaluation, we measure mathematical reasoning on four standard benchmarks: AMC,[5] MATH500,[6] MinervaMath,[7] and AIME24.[8] All model checkpoints and datasets are obtained from the Hugging Face Hub.

Algorithm 1 KAT-OPD

Require: warmup K, window W, trigger length T, buffer size B, percentile η, exemption L_0, max length $L_{\max}$.

Initialize FIFO buffer $\mathcal{B} \leftarrow \emptyset$.
for optimizer step $k = 1, \ldots, N$ **do**
  if $k > K$ and $|\mathcal{B}|$ sufficient **then**
    $s_k \leftarrow Q_{1-\eta}(\mathcal{B})$
  end if
  for prompt x in batch **do**
    $h_1 \leftarrow x$; $c \leftarrow 0$; $m \leftarrow +\infty$
    for $t = 1, \ldots, L_{\max}$ **do**
      $\hat{y}_t \sim \pi_\theta(\cdot \mid h_t)$
      Compute (or estimate) $d_t = D_{\mathrm{KL}}(p_t \| q_t)$
      if $t \geq L_0 + W$ **then**
        $z_t \leftarrow \frac{1}{W} \sum_{i=t-W+1}^{t} d_i$
        $m \leftarrow \min(m, z_t)$
        if $k > K$ and $z_t < s_k$ **then**
          $c \leftarrow c + 1$
        else
          $c \leftarrow 0$
        end if
        if $c \geq T$ **then**
          $\tau \leftarrow t$; **break**
        end if
      end if
    end for
    if $m < +\infty$ **then**
      push m into $\mathcal{B}$
    end if
  end for
  Compute standard OPD update on $\{(t, \hat{y}_t) : t \leq \tau\}$.
end for

Distillation objective. We run pure on-policy distillation ($\lambda = 1.0$) with reverse KL and softmax temperature $\tau = 1.0$. To control the cost of the divergence over a $\sim$150K-token vocabulary, both distributions are restricted to a truncated top-K support ($K = 64$) and re-normalized, with an explicit tail bucket carrying the residual probability mass; we find this approximation to match the full-vocabulary loss to within $\approx 0.5\%$ relative error while reducing the loss-side FLOPs by more than two orders of magnitude.

[1] https://huggingface.co/Qwen/Qwen3-8B
[2] https://huggingface.co/Qwen/Qwen3-1.7B-Base
[3] https://huggingface.co/Qwen/Qwen3-4B-Base
[4] https://huggingface.co/datasets/BytedTsinghua-SIA/DAPO-Math-17k
[5] https://huggingface.co/datasets/AI-MO/aimo-validation-amc
[6] https://huggingface.co/datasets/HuggingFaceH4/MATH-500
[7] https://huggingface.co/datasets/math-ai/minervamath
[8] https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

Student rollouts. During training, the student generates completions via a colocated vLLM engine (TP $= 1$, `gpu_memory_utilization` $= 0.3$ for the 1.7B student and 0.2 for the 4B student) under nucleus sampling with $T = 1.0$, top-$p = 0.95$, top-k disabled, and one completion per prompt. Validation rollouts use an independent budget of 4096 completion tokens. Student weights are synchronized into the vLLM engine after every optimizer step.

KL-based rollout truncation. Our proposed truncation operates on a sliding window of $L = 64$ response-token reverse KL values, opened after the first $t = 16$ tokens ($t = 32$ for the 4B student). A FIFO buffer of $M = 4096$ per-rollout minimum-window means is maintained on each worker; truncation is activated only after the buffer holds at least 512 samples and after a warm-up of $K = 50$ optimizer steps. The truncation threshold is set to the $(1 - \eta)$-quantile of the buffer with $\eta = 0.3$, and a rollout is cut once $N = 4$ consecutive windows fall below this threshold. As an ablation we also evaluate a per-rollout uniformly-random truncation baseline with a minimum keep length of 1 token.

Optimization. All experiments use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0, gradient clipping at 1.0, and a cosine learning-rate schedule with 3% linear warm-up. The peak learning rate is 1×10^{-6} for on-policy distillation. We train the 1.7B student for 3 epochs and the 4B student for 2 epochs. Per-device train batch size is 1 with 64 gradient-accumulation steps, yielding an effective batch of 128 sequences with 2 GPUs (1.7B) and 256 sequences with 4 GPUs (4B). Dropout is disabled, and all reported numbers are averaged over three runs with different random seeds for reproducibility.

Systems and hardware. All experiments run on a single node with $4\times$NVIDIA A800-SXM4 80 GB GPUs (NVLink). Training is launched through HuggingFace `Accelerate` in BF16 mixed precision with FlashAttention-2. The 1.7B student uses plain DDP across 2 GPUs; the 4B student uses DeepSpeed ZeRO-2 across 4 GPUs (no parameter or optimizer offload). Gradient checkpointing is enabled on the student; the teacher is kept frozen and runs a single forward pass per micro-batch.