Your Teacher Can't Help You Here: Combating Supervision Fidelity Decay in On-Policy Distillation

Yanjiang Liu[1,2] **Jie Lou**[4,†] **Xinyan Guan**[1,2] **Yuqiu Ji**[4] **Hongyu Lin**[2]

Ben He[1,2] **Xianpei Han**[2,3] **Le Sun**[2,3,†] **Xing Yu**[4] **Yaojie Lu**[2,3]

[1]University of Chinese Academy of Sciences
[2]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
[3]Key Laboratory of System Software, Chinese Academy of Sciences
[4]Xiaohongshu

Abstract

On-policy distillation transfers reasoning capabilities by training a student model on its own generated trajectories using token-level feedback from a teacher. However, we identify a critical bottleneck, **Supervision Fidelity Decay (SFD)**: as student-generated prefixes lengthen, the teacher's next-token distribution becomes less confident and less discriminative. Consequently, the teacher-dependent corrective signal in reverse-KL distillation weakens, causing student drift to compound across long reasoning chains. To mitigate SFD, we introduce **Lookahead Group Reward (LGR)**. Building on the insight that next-step teacher confidence reflects the discriminative strength of future reverse-KL supervision, LGR evaluates the student's top-K candidate tokens by the teacher confidence they induce at the subsequent step and assigns a group-normalized reward. To maintain computational efficiency, we further design an entropy-triggered tree-attention mechanism. Across six math and code benchmarks, LGR improves mean@8 by **2.57** points over OPD for a 7B student, with gains increasing in longer-generation and reaching **+4.92** points on AIME-26 at 39k tokens.

1 Introduction

Large language models (LLMs) with reasoning capabilities have achieved remarkable performance on complex mathematical and coding tasks. Recent advance models [1–3] demonstrate that extended reasoning chains can unlock capabilities previously thought to require much larger models. On-policy distillation (OPD), where a student model generates its own reasoning trajectories and learns from the teacher's token-level feedback, has emerged as a primary paradigm for transferring these capabilities to efficient, deployable models [4–10].

However, current OPD methods [4–7, 11] implicitly treat the teacher as a *static oracle* whose supervision quality is unaffected by what the student generates. We challenge this assumption and reveal a critical failure mode: as the student generates increasingly long sequences, its outputs progressively deviate from the teacher's training distribution, causing the teacher's supervision quality to *degrade monotonically*, which we refer to as **Supervision Fidelity Decay (SFD)**.

Does teacher supervision quality remain reliable throughout long reasoning chains — and if not, how can we actively maintain it?

Email: liuyanjiang22@mails.ucas.ac.cn, {luyaojie,hongyu,xianpei,sunle}@iscas.ac.cn
Codes are available at https://github.com/zui-jiang/LGR.
† Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

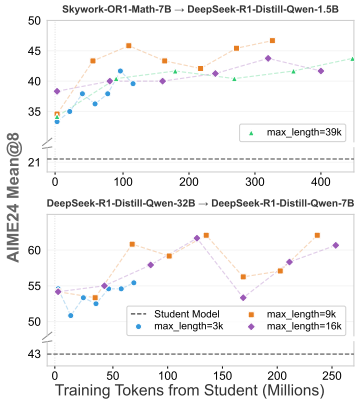


Figure 1: **Performance of different generation length in OPD.** AIME24 over training tokens for two model pairs. Performance improves from 3k to 9k, plateaus around 16k, and degrades at 39k.

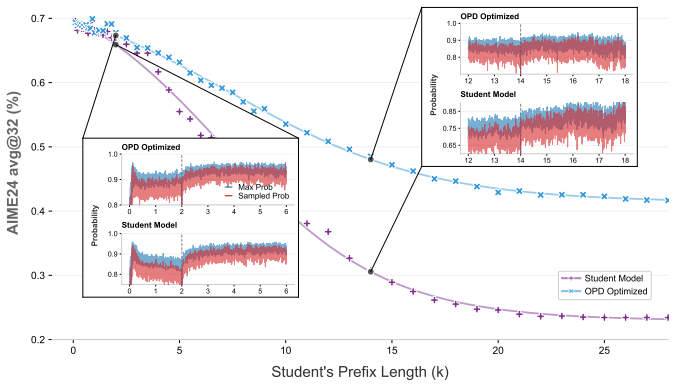


Figure 2: **Supervision Fidelity Decay.** *Main:* Teacher completion accuracy decays with student prefix length. *Insets:* Teacher confidence (max/sampled prob) at the student-to-teacher handoff for ∼2k (left) and ∼14k (right) prefixes. Confidence jumps at the handoff (dashed line), confirming token choices affect next-step teacher confidence.

As an initial observation, training OPD with varying maximum generation lengths (Figure 1) reveals that performance improves from 3k to 9k tokens and plateaus around 16k, followed by a significant decline at 39k. This trend indicates a potential failure mode during extremely long generations. To isolate the cause, we design a controlled prefix-completion experiment (Figure 2): as the student prefix grows longer, both the teacher's downstream task accuracy and peak next-token probability decay monotonically, while the teacher on its own prefixes maintains significantly higher confidence, which indicating that SFD stems from student drift. The subplots further show that teacher confidence jumps immediately when the teacher takes over, confirming that different token choices lead to different teacher confidence one step ahead.

Through theoretical analysis, we find that the declining max-prob directly collapses the reverse-KL gradient. As teacher confidence falls, its log-probability varies less across token choices, effectively reducing the learning signal to a student-only signal that reinforces existing modes without correction. Yet observations from the subplot suggest a solution; even at the same out-of-distribution context, teacher confidence at the subsequent position differs across token choices. By the same gradient analysis, higher next position confidence implies a more discriminative future signal: choosing the token that maximizes the next position's confidence directly preserves future supervision quality. We operationalize this as **Lookahead Group Reward (LGR)**, a group-normalized reward over the student's top-K candidates, where this group normalization removes the high-variance absolute confidence level and retains only the relative ranking across candidates. To maintain computational efficiency, we further design a tree-attention mechanism triggered by entropy.

Our key contributions are:

- **We identify SFD as a fundamental failure mode of OPD.** We show that teacher accuracy and peak confidence decay as student prefix length increases, a process that we prove collapses the reverse KL gradient into a signal that reinforces itself based solely on student outputs.
- **We propose a principled remedy that looks one step ahead.** Since higher teacher confidence at the next position provides more discriminative future supervision, LGR selects tokens using group normalized rewards and an efficient tree attention mechanism triggered by entropy.
- **We show LGR's gains grow with reasoning length.** LGR significantly outperforms OPD and alternative distillation methods, with pronounced gains on long reasoning tasks.

2 Supervision Fidelity Decays Along Student Trajectories

2.1 On-Policy Reverse-KL as Policy Gradient

Let π_T denote a teacher model and π_θ a student model parameterized by θ. Given a prompt $\mathbf{c}$, the student autoregressively generates a sequence $\mathbf{x} = (x_1, x_2, \ldots, x_L)$. On-policy reverse-KL

distillation minimizes [4, 12]:

$$\mathcal{L}_{\text{R-KL}}(\theta) = \mathbb{E}_{\mathbf{x}\sim\pi_\theta(\cdot|\mathbf{c})}\left[\sum_{t=1}^{L}\log\frac{\pi_\theta(x_t|\mathbf{x}_{<t},\mathbf{c})}{\pi_T(x_t|\mathbf{x}_{<t},\mathbf{c})}\right]. \tag{1}$$

Following standard practice in on-policy distillation [5], we detach the generated sequence $\mathbf{x}$ from the computation graph during loss computation. Under this stop-gradient assumption, the per-position KL gradient decomposes independently, yielding the policy gradient form:

$$\nabla_\theta\mathcal{L}_{\text{R-KL}} = \mathbb{E}_{\mathbf{x}\sim\pi_\theta}\left[\sum_{t=1}^{L}\nabla_\theta\log\pi_\theta(x_t|\mathbf{x}_{<t})\cdot A_t\right], \tag{2}$$

where the per-token advantage is:

$$A_t = 1 + \log\pi_\theta(x_t|\mathbf{x}_{<t}) - \log\pi_T(x_t|\mathbf{x}_{<t}). \tag{3}$$

This reveals that on-policy reverse-KL is equivalent to a policy gradient in the style of REINFORCE. Equivalently maximizing the per-token reward $r_t = \log\pi_T(x_t|\mathbf{x}_{<t}) - \log\pi_\theta(x_t|\mathbf{x}_{<t})$ [13, 14]. We refer to this as **local supervision**, in which the teacher provides a distributional target at each position.

2.2 The Supervision Capability Functional and Supervision Fidelity Decay

Supervision as a functional. The student policy π_θ generating a sequence of length L induces a *state visitation distribution* $\rho^L_{\pi_\theta}(c)$ over prefix contexts c. The teacher's supervision capability is then a **functional** of the student policy:

$$\mathcal{C}[\pi_\theta] = \mathbb{E}_{c\sim\rho^L_{\pi_\theta}}\left[f_T(c)\right], \quad \text{where} \quad f_T(c) = \max_{v\in\mathcal{V}}\pi_T(v|c) \tag{4}$$

is the teacher's local supervision quality at state c. This functional captures a key asymmetry. While the teacher remains unchanged [4–9], the student's state visitation distribution $\rho^L_{\pi_\theta}$ that shifts, dragging the integration domain into regions where f_T is low.

Position-dependent SFD curve. We define the position-dependent version:

$$\mathcal{C}^{(t)}[\pi_\theta] = \mathbb{E}_{c\sim\rho^t_{\pi_\theta}}\left[\max_v\pi_T(v|c)\right], \tag{5}$$

so that $\mathcal{C}^{(t)}[\pi_\theta]$ as a function of t traces the SFD curve. The aggregate $\mathcal{C}[\pi_\theta] = \mathbb{E}_t[\mathcal{C}^{(t)}[\pi_\theta]]$ summarizes total supervision quality. When L is short, $\rho^L_{\pi_\theta}$ remains within the teacher's training manifold and $f_T(c)$ stays high; when L is long, autoregressive drift causes $\rho^L_{\pi_\theta}$ to shift into the teacher's OOD region, where $f_T(c)$ collapses.

Supervision fidelity as a downstream metric. For a specific student-generated prefix $\mathbf{x}_{<t}$, the teacher's *supervision fidelity* at position t is:

$$\mathcal{F}(t) \coloneqq \max_{v\in\mathcal{V}}\pi_T(v|\mathbf{x}_{\le t},\mathbf{c}), \tag{6}$$

i.e., the teacher's peak next-token probability after observing the student's prefix up to t. Note that $\mathcal{C}^{(t)}[\pi_\theta] = \mathbb{E}[\mathcal{F}(t)]$ is the expectation of $\mathcal{F}(t)$ over student trajectories. At the task level, we can also measure supervision fidelity as $P(\text{teacher completes correctly from position } t \mid \mathbf{x}^\theta_{<t})$, which correlates strongly with $\mathcal{F}(t)$, thereby validating that max-p is an effective proxy for downstream supervision quality.

Empirical observation. Using DeepSeek-R1-Distill-Qwen-1.5B [2] as the student and 32B [2] as the teacher on AIME [15], we generate student prefixes of varying lengths and hand off to the teacher. Figure 2 shows: (1) teacher completion accuracy drops monotonically with prefix length; (2) $\bar{\mathcal{F}}$ decreases correspondingly, validating max-p as a proxy; (3) the teacher on its own prefixes maintains significantly higher fidelity, confirming SFD stems from student drift. The inset subplots further show that at the handoff point, teacher confidence *jumps immediately*, which demonstrates that token choices at position t do affect teacher confidence at $t{+}1$, even under severe SFD. The jump shrinks at longer prefixes ($\sim$14k vs. $\sim$2k) and after OPD optimization, consistent with accumulated drift. This directly motivates Section 3.2.

2.3 Theoretical Analysis: Signal Collapse and Compounding Drift

We formalize the impact of SFD on gradient quality. As the teacher's distribution becomes diffuse, its discriminative contribution vanishes; consequently, this single-position failure compounds across positions into a self-reinforcing drift.

Proposition 1 (Teacher signal vanishing under SFD)**.** *Define* $\Delta_T(t) \coloneqq \mathrm{Var}_{x_t \sim \pi_\theta}[\log \pi_T(x_t|\mathbf{x}_{<t})]$. *Then* $\Delta_T(t) \leq \log^2 |\mathcal{V}| - \mathrm{Ent}(\pi_T(\cdot|\mathbf{x}_{<t}))^2$*, so as the teacher's distribution becomes diffuse under SFD,* $\Delta_T(t) \to 0$ *and* $\mathrm{SNR}_T(t) = O(\Delta_T(t)) \to 0$*: the teacher's discriminative contribution to the gradient vanishes entirely, leaving a student-only signal that reinforces existing modes without correction.* (Proof: Appendix F.1.)

Once the teacher signal vanishes at position t, the student selects tokens from its own sharpened distribution, pushing the next context further out-of-distribution, thereby further degrading the signal at $t{+}1$.

Proposition 2 (Self-reinforcing drift under reverse-KL)**.** *Let* $d_t \coloneqq D(\pi_T(\cdot|\mathbf{x}^\theta_{<t}), \pi_T(\cdot|\mathbf{x}^*_{<t}))$ *be the distributional drift at position t, and assume* $\Delta_T(t)$ *is non-increasing in* d_t *(greater drift degrades teacher discriminability, as implied by SFD). When* $\Delta_T(t) < \delta_{\mathrm{crit}}$*, the teacher-free advantage reinforces the student's existing modes, causing* $\mathbb{E}[d_{t+1}|d_t] \geq d_t$*: drift compounds across positions, creating a positive feedback loop that degrades supervision irreversibly. Forward-KL avoids this by construction (*$d_t = 0$ *always) but introduces exposure bias.* (Proof sketch: Appendix F.2.)

2.4 Training Dynamics: The Vicious Cycle and Supervision Boundary Contraction

The functional perspective reveals a deeper consequence. Define the *effective learning horizon* $t_{\mathrm{eff}}(k) = \sup\{t : \mathcal{C}^{(t)}[\pi_{\theta_k}] > C_{\min}\}$ as the furthest position where teacher supervision exceeds a minimum useful threshold at training step k.

Under the self-reinforcing drift established in Proposition 2, training induces a **vicious cycle**: positions beyond t_{eff} receive no useful supervision ($\ell \approx 0$) and therefore do not improve; the student's uncorrected behavior at these positions continues to push the teacher further out-of-distribution, which in turn may cause t_{eff} to *shrink* at the next training step. This creates a "learning desert" that expands over training, establishing a **reasoning length ceiling** t^* beyond which the student can never improve under standard on-policy distillation.

The $\mathcal{C}^{(t)}$ curve follows a sigmoidal decay: the early portion maintains high supervision quality, the late portion collapses, and the transition sharpens progressively over training. We verify this empirically: fitting a logistic sigmoid to both curves in Figure 2 yields $R^2 > 0.997$, and the fitted parameters show that OPD training contracts the supervision boundary (t^*: $7.43 \to 7.04$k) and steepens the transition slope (ε: $0.21 \to 0.26$). Notably, all four parameter shifts align with the direction predicted by the vicious cycle (see Appendix C for full fit results).

This compounding effect is fundamental to on-policy reverse-KL and cannot be resolved by simply adjusting learning rates or adding regularization, as these measures only slow the drift rate ε without changing the structural problem. Instead, it motivates directly optimizing the supervision functional $\mathcal{C}[\pi_\theta]$ itself, which provides a structurally different signal that steers the student toward states where the teacher can provide high-quality supervision.

3 Lookahead Confidence as a Supervision Signal

3.1 From Gradient Failure to One-Step Lookahead

The preceding analysis shows that SFD collapses the gradient: as $\pi_T(\cdot|\mathbf{x}_{<t})$ becomes diffuse, $A_t = 1 + \log \pi_\theta - \log \pi_T$ loses discriminability and reduces to a student-only signal (Proposition 1). The standard objective (Eq. 1) is thus **agnostic to state quality**: it still forces the student to match the teacher's now-uninformative distribution.

One-step-ahead retains discriminability. Even when $\pi_T(\cdot|\mathbf{x}_{<t})$ is near-uniform, different token choices $x_t^{(k)}$ steer the context into different next-states, and the teacher's distributions at $t{+}1$ can differ substantially across candidates. Instead of asking what the teacher wants at position t (unanswerable

under SFD), we ask which candidate causes the *least further drift*, a question that remains informative even when local supervision has collapsed.

Breaking the vicious cycle. This signal directly targets Proposition 2: by favoring tokens that maintain higher teacher confidence at $t{+}1$, the student slows the per-step drift rate, preventing t_{eff} from contracting. The concrete realization is a per-token one-step-ahead confidence reward (Section 3.2), which provides a greedy one-step approximation to $\nabla_\theta \mathcal{C}[\pi_\theta]$, sufficient to slow drift without requiring full multi-step lookahead. We now formalize when this approximation retains useful discriminability.

Proposition 3 (One-step-ahead discriminability survives local supervision failure)**.** *Define the one-step-ahead discriminability at position t as:*

$$D_{ahead}(t) \coloneqq \max_{k,k'} \left| \max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k)}) - \max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k')}) \right|, \tag{7}$$

*which measures the maximum difference in teacher confidence at $t{+}1$ across candidate token choices at t. Then $D_{\text{ahead}}(t)$ is **independent** of the teacher's local entropy $\mathcal{H}(\pi_T(\cdot|\mathbf{x}_{<t}))$: even when $\pi_T(\cdot|\mathbf{x}_{<t})$ is uniform (maximum SFD, $\Delta_T(t) = 0$), $D_{\text{ahead}}(t)$ can be arbitrarily large.* (Proof: Appendix F.3.)

This proposition is the theoretical foundation for our confidence reward: even when local supervision (Proposition 1) has completely failed, the one-step-ahead comparison can still extract useful guidance from the teacher.

3.2 Confidence Reward Design

We operationalize the one-step-ahead comparison as a reward signal. At each position t, after the student samples $x_t \sim \pi_\theta(\cdot|\mathbf{x}_{<t})$, we evaluate the teacher's supervision fidelity at position $t{+}1$:

$$r_{\text{raw}}(x_t) = \max_{v \in \mathcal{V}} \pi_T(v|\mathbf{x}_{<t}, x_t, \mathbf{c}). \tag{8}$$

This measures the teacher's peak next-token probability *after* observing the student's action x_t. The critical distinction is temporal: local supervision (R-KL) evaluates the teacher's state at t given context $\mathbf{x}_{<t}$, while the confidence reward evaluates the teacher's state at $t{+}1$ given context $(\mathbf{x}_{<t}, x_t)$. By comparing this quantity across K candidates, we identify which token choice at t **causes the least further degradation** in teacher supervision quality at $t{+}1$. This selection seeks not to recover an in-distribution state, but rather to minimize the next step of SFD progression.

Proposition 4 (Max-p as relative drift indicator)**.** *Let P_T^* denote the teacher's next-token distribution conditioned on an in-distribution prefix, and let $P_T^{(\mathbf{x})}$ denote the distribution conditioned on a student-generated prefix $\mathbf{x}$. If the teacher is β-smooth (small perturbations in context produce bounded distributional shifts), then:*

$$\max_v P_T^*(v) - \max_v P_T^{(\mathbf{x})}(v) \leq \|P_T^* - P_T^{(\mathbf{x})}\|_\infty \leq \beta \cdot d(\mathbf{x}, \mathcal{X}_T), \tag{9}$$

*where $d(\mathbf{x}, \mathcal{X}_T)$ measures the distance from the teacher's in-distribution manifold $\mathcal{X}_T$. That is, **higher max-p implies closer proximity to the teacher's competent region**.* (Proof: Appendix F.4.)

Remark 1 (Max-p vs. entropy)**.** *Maximizing $\max_v \pi_T(v|\cdot)$ is directionally equivalent to minimizing the teacher's prediction entropy $\mathcal{H}(\pi_T(\cdot|\cdot))$.*

Group normalization. Since r_{raw} varies in absolute magnitude across positions and tasks, we normalize within the top-K student candidates (ranked by π_θ), inspired by GRPO [2]:

$$r_{\text{conf}}(x_t) = \frac{r_{\text{raw}}(x_t) - \mu_K}{\sigma_K + \epsilon}, \tag{10}$$

where μ_K, σ_K are the group mean and std over $\{r_{\text{raw}}^{(k)}\}_{k=1}^K$. This removes absolute-scale variance while preserving relative ranking across candidates (Appendix F.5).

Combined loss. Adding the confidence reward to the per-token advantage (Eq. (3)) gives the LGR loss:

$$\mathcal{L}_t^{LGR} = A_t + \gamma \cdot r_{\text{conf}}(x_t). \tag{11}$$

Algorithm 1 LGR: Lookahead Group Reward

Require: Student π_θ, Teacher π_T, entropy threshold τ, top-K, reward weight γ

for each training step **do**
  Sample prompt $\mathbf{c}$ from dataset
  Generate $\mathbf{x} = (x_1, \ldots, x_L) \sim \pi_\theta(\cdot|\mathbf{c})$ *// Student rollout*
  Compute student logits and entropy $\mathcal{H}_t$ for all positions
  Identify high-entropy set $\mathcal{S} = \{t : \mathcal{H}_t > \tau\}$
  Extract top-K candidate tokens at each $t \in \mathcal{S}$
  Construct tree-attention input: main sequence + all candidates
  Construct multi-branch tree mask $\mathbf{M}$
  Run teacher forward pass with tree mask $\mathbf{M}$ *// Single pass for all positions*
  Extract $r_{\text{raw}}^{(k)}$ for all candidates; compute r_{conf} via Eq. (10)
  for each position $t = 1, \ldots, L$ **do**
    if $t \in \mathcal{S}$ **then**
      $\mathcal{L}_t \leftarrow \mathcal{L}_t^{LGR}$ via Eq. (12) *// Local + lookahead*
    else
      $\mathcal{L}_t \leftarrow \log \frac{\pi_\theta(x_t|\mathbf{x}_{<t})}{\pi_T(x_t|\mathbf{x}_{<t})}$ *// Standard R-KL*
    end if
  end for
  Update θ with $\nabla_\theta \sum_t \mathcal{L}_t$
end for

Since we already compute teacher confidence for the top-K candidates, we extend to a multi-sample estimator. Letting $A_t^{(k)} = 1 + \log \pi_\theta(x_t^{(k)}|\mathbf{x}_{<t}) - \log \pi_T(x_t^{(k)}|\mathbf{x}_{<t})$, the LGR (topk) loss weights each candidate by its student probability:

$$\mathcal{L}_t^{LGR(topk)} = \sum_{k=1}^{K} \pi_\theta(x_t^{(k)}|\mathbf{x}_{<t}) \cdot \left[A_t^{(k)} + \gamma \cdot r_{\text{conf}}(x_t^{(k)}) \right]. \tag{12}$$

3.3 Efficient Computation

Entropy trigger. Computing the confidence reward for all L positions with K candidates each would be prohibitively expensive. We observe that at positions where the *student* has low generation entropy, the top-1 token dominates the probability mass, and the remaining candidates carry negligible weight in the policy gradient. We trigger the confidence reward only at positions $\mathcal{S} = \{t : \mathcal{H}(\pi_\theta(\cdot|\mathbf{x}_{<t})) > \tau\}$; elsewhere, only the standard reverse-KL loss is applied. This typically selects $|\mathcal{S}| \approx 0.15L$–$0.25L$ positions early in training, stabilizing around $0.10L$ as the student distribution sharpens, reducing computational overhead by 4–7 times.

The triggering uses *student* entropy rather than teacher entropy, avoiding circularity: teacher entropy at t is exactly what degrades under SFD, so triggering on it would disable the reward precisely where it is most needed. This selection is near-lossless (Appendix D).

Tree attention. Naively evaluating K candidates at each $t \in \mathcal{S}$ requires $K \cdot |\mathcal{S}|$ teacher forward prefill for student's prefix. We instead construct an extended sequence (main sequence $\mathbf{x}$ followed by all candidates as branches) with a tree-structured mask $\mathbf{M}$ [16–18] : main branch tokens attend causally; each candidate $x_t^{(k)}$ attends to $\mathbf{x}_{\leq t-1}$ but not to other candidates (see Figure 12 for an example). In practice, GPU memory limits the total sequence length, so candidates are processed in N segments rather than all at once. Appendix E gives the full cost analysis.

4 Experiments

4.1 Experimental Setup

Full training and evaluation details are provided in Appendix A.

Models. We evaluate two teacher–student configurations sharing the same vocabulary, each with a single teacher: (1) **1.5B student:** DeepSeek-R1-Distill-Qwen-1.5B [2] as the student, trained separately for math (teacher: SkyWork-OR1-Math-7B [19]) and code (teacher: DeepCoder-14B [20]).

Table 1: Comparison of distillation methods on mathematical reasoning and code benchmarks (mean@8 / pass@8). Δ denotes the absolute difference between our best method and OPD (9k).

	Math					Code	
Model	**AIME-24**	**AIME-25**	**AIME-26**	**HMMT-25**	**HMMT-26**	**LCB_v6**	**AVG.**
SkyWork-OR1-Math-7B / DeepCoder-14B → DeepSeek-R1-Distill-Qwen-1.5B							
SkyWork-OR1-Math-7B	67.50/83.33	52.08/66.67	62.08/80.00	32.22/55.17	36.74/54.55	–	–
DeepCoder-14B	–	–	–	–	–	76.87/86.34	–
Base Model	24.17/50.00	21.67/33.33	17.92/46.67	11.67/23.33	12.50/27.27	10.13/31.06	16.34/35.28
+ GRPO	38.33/66.67	30.83/43.33	28.75/50.00	16.74/20.69	20.46/27.27	24.56/40.31	26.61/41.38
+ OPD (9k)	45.83/80.00	33.75/46.67	32.08/56.67	20.42/36.67	22.73/27.27	34.58/50.44	31.57/49.62
+ OPD (topk)	45.83/70.00	31.67/43.33	33.75/66.67	20.50/27.59	21.59/33.33	26.04/37.89	29.90/46.47
+ JSD	38.75/66.67	34.17/50.00	32.50/50.00	18.75/23.33	21.21/33.33	29.02/46.92	29.07/45.04
+ REOPOLD	41.67/60.00	33.33/46.67	30.00/46.67	19.17/26.67	22.35/27.27	33.59/49.56	30.02/42.81
+ LGR	**46.67**/73.33	**35.42**/50.00	**34.17**/63.33	**22.08**/36.67	23.86/37.50	**36.89**/53.97	**33.18**/52.47
+ LGR (topk)	44.17/80.00	**35.42**/56.67	33.75/63.33	**22.08**/46.67	**24.62**/34.38	33.43/50.00	32.25/55.18
Δ	+0.84	+1.67	+2.09	+1.66	+1.89	+2.31	+1.61
DeepSeek-R1-Distill-Qwen-32B → DeepSeek-R1-Distill-Qwen-7B							
DeepSeek-R1-Distill-Qwen-32B	80.83/93.33	50.83/73.33	70.00/83.33	50.83/70.00	43.94/63.64	68.34/82.16	–
Base Model	52.08/80.00	43.33/63.33	42.92/63.33	24.17/43.33	28.03/39.39	41.96/61.23	38.75/58.44
+ GRPO	67.50/83.33	52.08/66.67	62.08/80.00	32.22/55.17	36.74/54.55	50.67/69.33	50.22/68.17
+ OPD (9k)	**62.08**/76.67	45.42/63.33	52.50/76.67	27.50/50.00	32.95/45.45	54.24/72.01	45.78/64.02
+ OPD (topk)	57.50/80.00	45.00/66.67	52.08/70.00	29.58/56.67	30.30/42.42	53.36/71.81	44.64/64.60
+ JSD	60.83/83.33	45.83/56.67	51.25/66.67	25.83/43.33	31.82/42.42	50.39/68.01	44.33/60.07
+ REOPOLD	57.08/83.33	47.50/70.00	**54.17**/80.00	30.00/56.67	32.95/46.88	54.41/72.47	46.02/68.23
+ LGR	61.67/83.33	**49.58**/73.33	53.75/80.00	**31.67**/60.00	**34.85**/51.52	**58.59**/74.45	**48.35**/70.44
+ LGR (topk)	61.67/80.00	47.08/66.67	53.75/76.67	27.92/53.33	30.68/36.36	56.61/73.34	46.29/64.40
Δ	-0.41	+4.16	+1.25	+4.17	+1.90	+4.35	+2.57

(2) **7B student:** DeepSeek-R1-Distill-Qwen-7B as the student, with DeepSeek-R1-Distill-Qwen-32B as the teacher.

Benchmarks. We evaluate our method on several rigorous datasets. For mathematical reasoning, we utilize AIME-24 [15], AIME–25 [21], and AIME-26 [22], as well as the February sessions of HMMT-25 and HMMT-26 [23]. For evaluation of code generation, we employ the LiveCodeBench v6 suite [24].

Baselines. We compare LGR against: (1) **GRPO** [2]: Group Relative Policy Optimization with outcome-level reward; (2) **OPD**: on-policy distillation with standard reverse-KL, plus a top-K multi-sample variant (OPD topk); (3) **JSD** [4]: Jensen–Shannon divergence distillation; and (4) **REOPOLD** [11]: Relaxed On-Policy Distillation, which stabilizes RKL training via mixture-based reward clipping (to handle heavy-tailed negative rewards) and entropy-guided token-level dynamic sampling (to filter near-zero reward tokens).

4.2 Main Results

Table 1 presents the main comparison across both teacher–student configurations.

LGR delivers superior aggregate performance across benchmarks. In the 1.5B student setting, LGR achieves an average of 33.18% mean@8, which is an increase of 1.61% absolute over OPD 9k. These gains are consistent across all six evaluation sets, with the largest improvements occurring on AIME-26 and LCB_v6. For the larger 7B student model, LGR widens the lead to an average improvement of 2.57%. While it performs competitively but slightly behind the baseline on AIME-244, LGR yields significant gains on the remaining five benchmarks, particularly on AIME-25 and LCB_v6 where improvements exceed 4.1%.

The advantage is more pronounced at larger scale. In the 7B student setting, LGR achieves 48.35% average mean@8, a +2.57% improvement over OPD 9k. The gains are particularly large on AIME-25 (+4.16%), HMMT-25 (+4.17%), and LCB_v6 (+4.35%), suggesting that LGR is especially effective when the student has sufficient capacity to leverage the improved supervision signal.

LGR vs. LGR (topk). The top-K extension provides complementary but not strictly additive gains. In the 1.5B setting, LGR (topk) achieves notably higher pass@8 on several benchmarks (e.g., 56.67% vs. 50.00% on AIME-25), suggesting that the multi-sample estimator helps with diversity. In the 7B

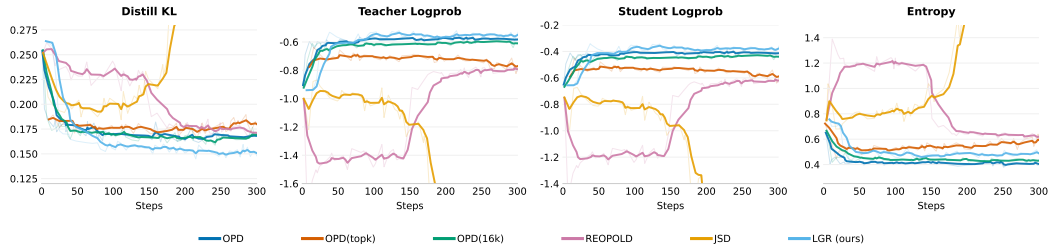


Figure 3: **Training dynamics across distillation methods.** Comparison of metrics over training steps. LGR maintains higher teacher log-probability and more stable entropy.

setting, LGR without top-K generally outperforms, indicating that the single-sample estimator with confidence reward is sufficient at larger scale.

Stabilization alone does not address SFD. JSD and REOPOLD achieve comparable or lower performance than OPD, confirming that the core issue is not the choice of divergence or optimization instability, but the degradation of teacher supervision quality on student-generated contexts, a problem that demands a structurally different solution.

4.3 LGR Addresses Supervision Fidelity Decay

Training dynamics. Figure 3 tracks distill KL, teacher log-probability, student log-probability and entropy during training. Compared to OPD variants, LGR maintains higher teacher log-probability throughout training, indicating that the student's generated contexts remain closer to the teacher's in-distribution manifold. The entropy curves show that LGR maintains more stable generation diversity rather than collapsing into the mode-sharpening behavior predicted by Proposition 2.

LGR's advantage grows with maximum generation length. Table 2 compares OPD and LGR at different maximum generation lengths (3k, 9k, 16k, 39k). At short lengths (3k), where SFD is minimal, LGR provides no advantage because the teacher's local supervision is already reliable. As the maximum length increases, LGR increasingly outperforms OPD. At 39k, LGR achieves the strongest gains across all three AIME benchmarks (e.g., 36.75 vs. 32.08 on AIME-25, 34.92 vs. 30.00 on AIME-26). This pattern is precisely what our SFD analysis predicts: the confidence reward becomes increasingly valuable as the teacher's local supervision degrades at longer positions.

Table 2: **OPD vs. LGR across max generation lengths.** Mean@8/Pass@8 on AIME benchmarks (1.5B student).

Max Len	Method	AIME-24	AIME-25	AIME-26
3k	OPD	42.92/76.67	33.33/50.00	32.92/60.00
	LGR	41.25/66.67	32.92/40.00	27.50/43.33
9k	OPD	45.83/**80.00**	33.75/46.67	32.08/56.67
	LGR	**46.67**/73.33	35.42/50.00	34.17/63.33
16k	OPD	44.17/73.33	33.75/50.00	34.17/63.33
	LGR	46.25/76.67	36.25/53.33	33.33/63.33
39k	OPD	43.33/66.67	32.08/53.33	30.00/53.33
	LGR	46.25/73.33	**36.75/60.00**	**34.92/66.67**

Confidence reward dynamics. Figure 4 tracks the LGR confidence reward over training. The average confidence reward increases over training (top panel), indicating that the student progressively learns to generate tokens that lead to higher teacher confidence at the next position. The applied ratio (bottom panel) stabilizes around 10%, showing that the entropy-triggered activation identifies a consistent fraction of high-entropy decision points. This stable activation ratio confirms that the entropy trigger effectively identifies positions where lookahead guidance is most needed, without requiring manual tuning over the course of training.

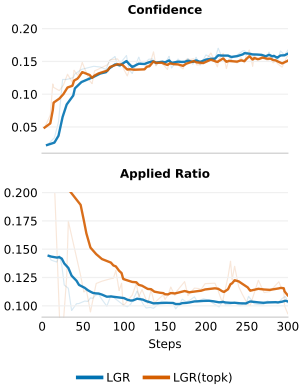


Figure 4: **LGR confidence reward dynamics.** *Top:* The average confidence reward increases over training. *Bottom:* The applied ratio stabilizes around 10%.

4.4 Ablation Studies

Confidence metric comparison. We compare four candidate confidence metrics for the group-normalized reward (Table 3a). Max-p consistently outperforms the alternatives, achieving 46.67

Table 3: **Ablation studies** on AIME benchmarks (Mean@8/Pass@8, 1.5B student). *Left:* Confidence metric comparison. *Right:* Sensitivity to confidence weight γ.

(a) Confidence metric

Metric	AIME-24	AIME-25	AIME-26
PPL	37.92/70.00	26.67/43.33	26.67/53.33
Entropy	42.08/73.33	29.58/46.67	25.83/53.33
Sampled-p	43.75/70.00	29.17/43.33	32.50/50.00
Max-p	**46.67/73.33**	**35.42/50.00**	**34.17/63.33**

(b) Confidence weight γ

γ	AIME-24	AIME-25	AIME-26
0.1	43.33/73.33	31.67/46.67	32.08/56.67
1.0	**46.67**/73.33	**35.42/50.00**	**34.17/63.33**
1.5	44.58/**76.67**	34.17/50.00	33.75/63.33
10.0	37.08/70.00	30.00/40.00	28.33/50.00

on AIME-24 compared to 43.75 for Sampled-p, 42.08 for entropy, and 37.92 for PPL. The strong advantage of Max-p aligns with Proposition 4: it directly measures the proximity to the teacher's in-distribution regime, while PPL and entropy are noisier proxies that can be inflated by irrelevant low-probability tokens.

Sensitivity to confidence weight γ. Table 3b illustrates how the choice of γ affects the distillation outcomes. When γ is as low as 0.1, the influence of the lookahead signal is negligible, and the performance remains close to the standard OPD baseline. The optimal results are achieved in the range of 1.0 to 1.5, where the confidence reward effectively guides the student toward states that maintain high supervision quality. In contrast, setting γ to 10.0 causes a significant decline in accuracy. In this high weight regime, the optimization becomes susceptible to reward hacking, as the student model tends to generate repetitive token sequences that artificially inflate teacher confidence but lack the logical substance required to solve the task correctly.

5 Related Work

On-Policy Distillation. Knowledge distillation [25–30] transfers capabilities from teacher to student. For LLMs, training on teacher-generated (off-policy) data creates a distribution mismatch at inference time [31–33]. MiniLLM [6] addresses this by minimizing reverse-KL on student-generated rollouts to avoid the mode-averaging problem of forward-KL; GKD [4] further demonstrates that on-policy generated data consistently outperforms off-policy training. These works establish OPD as the standard paradigm for reasoning model compression. Subsequent work has explored several directions. On *training stability*: REOPOLD [11] stabilizes reverse-KL training via mixture-based reward clipping and entropy-guided token-level dynamic sampling. On *objective design*: G-OPD [34] formalizes OPD as KL-constrained RL and proposes reward extrapolation to learn beyond the teacher ceiling; TSD-KD [35] applies KL loss selectively on high-entropy tokens where the student is genuinely uncertain, reducing noise from low-uncertainty positions.

Two concurrent works are most closely related. Revisiting OPD [36] identify that OPD becomes unreliable on student generations and propose teacher top-K support matching. Rethinking OPD [37] study OPD phenomenology and find that reward quality degrades with trajectory depth—consistent with our SFD analysis. Our work differs in two respects: (1) we provide a formal analysis of *why* and *where* supervision degrades—characterizing SFD as a position-dependent functional that worsens monotonically along student trajectories and establishes a reasoning length ceiling; and (2) we propose a one-step lookahead reward remedy that directly optimizes teacher confidence at the future position, orthogonal to divergence modification and applicable on top of OPD objective.

6 Conclusion

We proposed Lookahead Group Reward, which augments standard reverse-KL distillation with a group-normalized confidence reward that directly optimizes the teacher's supervision capability functional. Unlike masking or truncation strategies that passively avoid weak-supervision regions, LGR actively steers the student toward trajectories where the teacher maintains high supervision fidelity. The entropy-triggered tree-attention mechanism makes this approach computationally practical. Achieving a 1000$\times$ speedup compared to a naïve implementation.

Limitations and future work

First, LGR requires white-box access to the teacher model's logits, which may not always be available. Second, the confidence reward relies on an implicit assumption that the teacher's *relative ranking* of candidate tokens remains informative even when its absolute predictions are unreliable. While our group normalization design and empirical results support this assumption, it may weaken for extremely out-of-distribution student trajectories or poorly calibrated teachers. Third, the current design uses a fixed number of candidates K and a static entropy threshold τ; dynamically adjusting these based on training progress could improve both efficiency and effectiveness. Finally, extending the confidence reward framework to multi-modal reasoning settings represents a promising direction.

Ackonwledge

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the Natural Science Foundation of China (No. 62536008, 62572456, 62476265), the Basic Research Program of ISCAS, Grant No. ISCAS-JCMS-202602.

References

[1] OpenAI, :, A. Jaech, et al. Openai o1 system card, 2026.

[2] Guo, D., D. Yang, H. Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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

[4] Agarwal, R., N. Vieillard, Y. Zhou, et al. On-policy distillation of language models: Learning from self-generated mistakes, 2024.

[5] Lu, K., T. M. Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. Https://thinkingmachines.ai/blog/on-policy-distillation.

[6] Gu, Y., L. Dong, F. Wei, et al. Minillm: On-policy distillation of large language models, 2026.

[7] Patiño, C. M., K. Rasul, Q. Gallouédec, et al. Unlocking on-policy distillation for any model family, 2025.

[8] Yang, A., A. Li, B. Yang, et al. Qwen3 technical report, 2025.

[9] Team, C., B. Xiao, B. Xia, et al. Mimo-v2-flash technical report, 2026.

[10] Yang, Z., Z. Liu, Y. Chen, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation, 2026.

[11] Ko, J., S. Abdali, Y. J. Kim, et al. Scaling reasoning efficiently via relaxed on-policy distillation, 2026.

[12] Xiao, T., Y. Yuan, M. Li, et al. On a connection between imitation learning and rlhf, 2025.

[13] Sutton, R. S., A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edn., 2018.

[14] Rafailov, R., A. Sharma, E. Mitchell, et al. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine, eds., *Advances in Neural Information Processing Systems*, vol. 36, pages 53728–53741. Curran Associates, Inc., 2023.

[15] Zhang, Y., T. Math-AI. American invitational mathematics examination (aime) 2024, 2024.

[16] Cai, T., Y. Li, Z. Geng, et al. Medusa: Simple llm inference acceleration framework with multiple decoding heads, 2024.

[17] Li, Y., F. Wei, C. Zhang, et al. Eagle: Speculative sampling requires rethinking feature uncertainty, 2025.

[18] —. Eagle-3: Scaling up inference acceleration of large language models via training-time test, 2025.

[19] He, J., J. Liu, C. Y. Liu, et al. Skywork open reasoner 1 technical report, 2025.

[20] Balog, M., A. L. Gaunt, M. Brockschmidt, et al. Deepcoder: Learning to write programs, 2017.

[21] Zhang, Y., T. Math-AI. American invitational mathematics examination (aime) 2025, 2025.

[22] —. American invitational mathematics examination (aime) 2026, 2026.

[23] Dekoninck, J., N. Jovanović, T. Gehrunger, et al. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026.

[24] Jain, N., K. Han, A. Gu, et al. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024.

[25] Hinton, G., O. Vinyals, J. Dean. Distilling the knowledge in a neural network, 2015.

[26] Ko, J., T. Chen, S. Kim, et al. Distillm-2: A contrastive approach boosts the distillation of llms, 2025.

[27] Kim, Y., A. M. Rush. Sequence-level knowledge distillation. *CoRR*, abs/1606.07947, 2016.

[28] Ye, T., L. Dong, Z. Chi, et al. Black-box on-policy distillation of large language models, 2026.

[29] Hübotter, J., F. Lübeck, L. Behric, et al. Reinforcement learning via self-distillation, 2026.

[30] Kim, T., J. Oh, N. Kim, et al. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation, 2021.

[31] Bengio, S., O. Vinyals, N. Jaitly, et al. Scheduled sampling for sequence prediction with recurrent neural networks, 2015.

[32] He, T., J. Zhang, Z. Zhou, et al. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation?, 2021.

[33] Kim, Y., D. Shin, M. Kang, et al. Distillation of large language models via concrete score matching, 2026.

[34] Yang, W., W. Liu, R. Xie, et al. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026.

[35] Kim, M., S. J. Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation, 2026.

[36] Fu, Y., H. Huang, K. Jiang, et al. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

[37] Li, Y., Y. Zuo, B. He, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026.

[38] Zhu, Z., C. Xie, X. Lv, et al. slime: An llm post-training framework for rl scaling. `https://github.com/THUDM/slime`, 2025. GitHub repository. Corresponding author: Xin Lv.

[39] Zheng, L., L. Yin, Z. Xie, et al. Sglang: Efficient execution of structured language model programs, 2024.

[40] Lin, A., J. Wohlwend, H. Chen, et al. Autoregressive knowledge distillation through imitation learning. In B. Webber, T. Cohn, Y. He, Y. Liu, eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6121–6133. Association for Computational Linguistics, Online, 2020.

[41] Bhatia, R., C. Davis. A better bound on the variance. *The american mathematical monthly*, 107(4):353–357, 2000.

[42] Cover, T., J. Thomas. *Elements of Information Theory*. Wiley, 2012.

Technical Appendices and Supplementary Material

Table of Contents

A. Training and Evaluation Details 13
B. Additional Experimental Results 14
B.1. Effect of Renormalization in Top-K Training 15
B.2. Effect of Rollout Temperature 16
B.3. Per-Token Top-K Logit Visualization 17
B.4. Future-KL with GAE Weighting 18
B.5. Comparison of KL Divergence Objectives 19
C. Sigmoid Fit of the SFD Curve 20
D. Entropy-Triggered Activation is Near-Lossless 21
E. Tree Attention: Cost Analysis and Practical Segmentation 21
F. Proofs of Theoretical Results 22
F.1. Proof of Proposition 1 (Teacher Signal Vanishing under SFD) 22
F.2. Proof Sketch of Proposition 2 (Self-Reinforcing Drift under Reverse-KL) 23
F.3. Proof of Proposition 3 (One-Step-Ahead Discriminability) 24
F.4. Proof of Proposition 4 (Max-p as Relative Drift Indicator) 24
F.5. Group Normalization: Design Rationale and Formal Properties 25

A Training and Evaluation Details

Training framework. All models are trained on 4 nodes of 8×H20 GPUs (32 GPUs total) with the SLIME framework [38], using SGLang [39] as the inference backend. We made two adaptations to SGLang: (1) we extended it to support tree-structured attention masks, enabling the single-pass multi-candidate teacher evaluation described in Section 3.3; (2) we patched SGLang to return temperature-scaled logits for the prefill portion, which was required for experiments exploring non-unit rollout temperatures (see below).

Training data. For math training we use the Polaris dataset; for code training we use the DeepScaler code dataset. This applies uniformly across all student model sizes.

Rollout temperature. All experiments use rollout temperature $= 1.0$ for both student and teacher. We also evaluated two non-unit configurations: (1) $t_{\text{student}} = 1, t_{\text{teacher}} = 0.2$ (sharpened teacher distribution); (2) $t_{\text{student}} = 0.8, t_{\text{teacher}} = 0.8$ (symmetric low temperature). Both configurations degraded performance. We therefore fix temperature $= 1.0$ across all methods and settings.

Baseline-specific configurations.

- **GRPO:** Rather than training GRPO from scratch, we report results from publicly available RL-trained checkpoints to ensure a fair and reproducible comparison. For the **1.5B student** setting, we use DeepScaler-1.5B, which was obtained by applying GRPO to DeepSeek-R1-Distill-Qwen-1.5B on a large-scale math dataset. For the **7B student** setting, we use SkyWork-OR1-Math-7B, which was obtained by applying GRPO to DeepSeek-R1-Distill-Qwen-7B. Both checkpoints share the same base model as the corresponding distillation experiments, making the comparison directly controlled for initialization.
- **OPD (topk) and LGR (topk):** The per-token top-K candidate set is the union of the student's top-10 and teacher's top-10 tokens, renormalized to a valid probability distribution. Renormalization is essential: without it the joint candidate set has inconsistent probability mass and training diverges. The union construction also guarantees that teacher-preferred tokens are always included even when the student assigns them low probability.
- **REOPOLD:** We follow the original training configuration but set staleness $= 1$ (fully on-policy). The original paper uses staleness $= 4$; we found this setting to be unstable in our experiments, likely because the larger policy lag interacts poorly with the entropy-triggered reward clipping mechanism.

Table 4: Training hyperparameters for the two student configurations.

Hyperparameter	**1.5B Student**	**7B Student**
Optimizer	AdamW	AdamW
Adam β_1, β_2	0.9, 0.98	0.9, 0.98
Weight decay	0.1	0.1
Learning rate	1×10^{-5}	3×10^{-6}
Batch size	128	128
Warmup steps	0	0
Training steps	300	300
Rollout temperature	1.0	1.0
Rollout samples per prompt	1	1
LGR-specific		
Top-K candidates (K)	8	8
Entropy threshold (τ)	0.2	0.2
Confidence reward weight (γ)	1.0	1.0
Tree attention segments (N)	8	8

- **JSD:** We use mixture coefficient $\beta = 0.2$ (i.e., $\pi_{\text{mix}} = 0.2\,\pi_T + 0.8\,\pi_\theta$).

Evaluation. All models are evaluated with a maximum generation length of 39k tokens for math benchmarks and 36k tokens for code benchmarks (code prompts are longer, leaving less budget for generation), at temperature $= 1$. We report mean@8 and pass@8 across 8 responses per problem.

On the choice of student models. Our main experiments use DeepSeek-R1-Distill models as students (1.5B and 7B), which are initialized from a base model via supervised fine-tuning on reasoning traces. We also experimented with applying OPD to Qwen3 reasoning models (specifically, distilling from Qwen3-32B to Qwen3-1.7B), but found that training is highly unstable: performance initially improves but then degrades progressively rather than converging—a pattern visible in Figures 7 and 8, where even the best-configured runs show initial gains followed by gradual degradation. We attribute this to the nature of the Qwen3 reasoning models, which are the result of extensive integrated multi-domain RL fusion applied to a broad mixture of tasks—yielding a well-balanced student that has already converged on the teacher's distribution across diverse contexts. When such a model is used as a student in further OPD, the training signal is dominated by the small residual distribution mismatch rather than systematic supervision failures, making the optimization landscape highly sensitive and difficult to stabilize. Note that our results differ significantly from those reported by Thinking Machine Lab on similar model families: their experiments use a base model or an SFT-from-base model as the student, rather than a fully trained reasoning model, which explains the more stable training dynamics they observe.

In contrast, DeepSeek-R1-Distill models are derived from a base model through reasoning-focused supervised learning, without the breadth of integrated multi-domain RL fusion. This leaves meaningful room for OPD to provide corrective supervision and makes the SFD phenomenon clearly observable—as the student has not already adapted to the teacher's distribution on student-generated prefixes. Using this model family therefore provides a cleaner testbed for diagnosing and addressing SFD, and the instability observed on Qwen3 further underscores the importance of student model selection in OPD experiments.

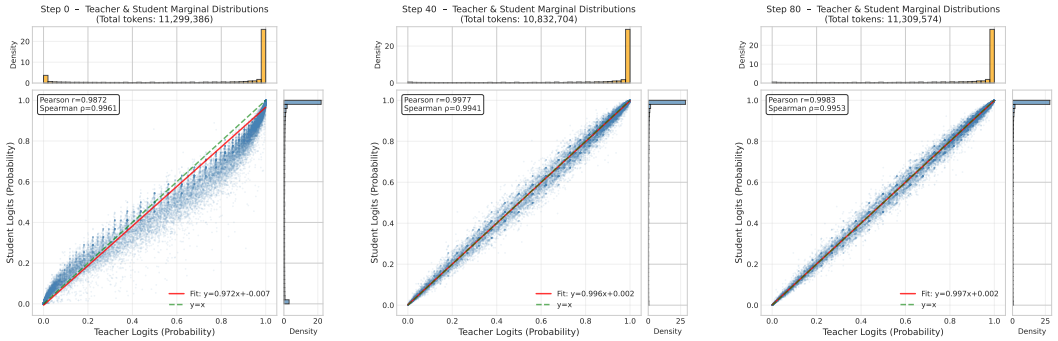
B Additional Experimental Results

B.1 Homogeneous vs. Heterogeneous Teacher–Student Pairs

In on-policy distillation, the teacher and student can either share the same base model (*homogeneous*) or originate from different model families (*heterogeneous*) [40, 7]. In the homogeneous setting, the teacher is obtained by further training the student base model via RLVR, so the two models share identical tokenization and pretraining priors—the student's on-policy distribution is close to the teacher's from the outset. In the heterogeneous setting (used throughout this paper), the student and

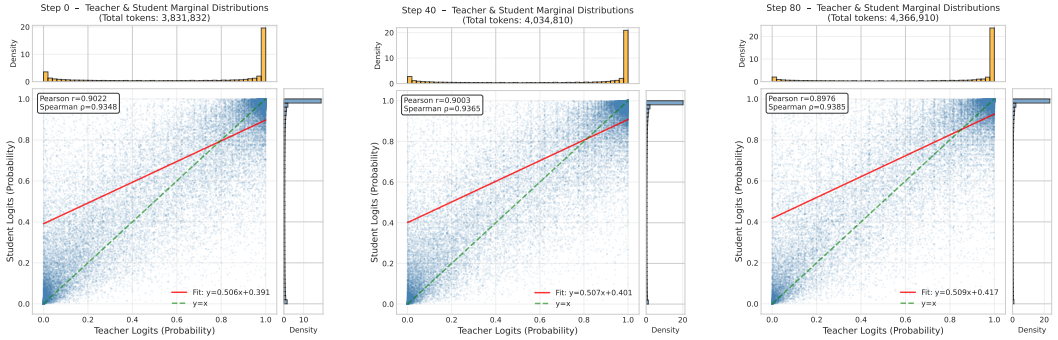
teacher come from different base models, introducing a structural distribution gap that is present even before any distillation training begins.

Figures 5 and 6 compare the joint distribution of per-token student log-probability and teacher log-probability at training steps 0, 40, and 80 for the two settings. In the homogeneous case, the scatter plots show a tight correlation between student and teacher log-probabilities throughout training: the student's on-policy tokens remain well within the teacher's in-distribution region, and this alignment persists even as training progresses. In the heterogeneous case, the scatter is substantially wider at all steps, with the student frequently visiting token positions where the teacher assigns low probability—precisely the regime in which supervision fidelity degrades. This structural distribution gap is why the SFD phenomenon is clearly observable in the heterogeneous setting, and why our experiments adopt this configuration as the primary testbed.



(a) Step 0 (b) Step 40 (c) Step 80

Figure 5: Homogeneous teacher–student pair: joint distribution of per-token student log-probability vs. teacher log-probability at training steps 0, 40, and 80. The teacher is obtained by further RLVR training from the same student base model, so the two models share pretraining priors and the student tokens remain well within the teacher's in-distribution region throughout training.



(a) Step 0 (b) Step 40 (c) Step 80

Figure 6: Heterogeneous teacher–student pair: joint distribution of per-token student log-probability vs. teacher log-probability at training steps 0, 40, and 80. The teacher and student originate from different base model families. The scatter is substantially wider than the homogeneous case, indicating that the student frequently generates tokens in low-probability regions of the teacher's distribution—the primary driver of supervision fidelity decay.

B.2 Effect of Renormalization in Top-K Training

In OPD (topk) and LGR (topk), each token's distribution is restricted to the union of the student's top-10 and teacher's top-10 candidates, and the resulting truncated distribution is renormalized to sum to one before computing the KL loss. Here we ablate the necessity of this renormalization step

by training the same models without it—i.e., computing the KL directly over the raw unnormalized top-K union probabilities.

Figure 7 compares OPD (topk) with and without renormalization on the Qwen3-1.7B student configuration across AIME benchmarks. We make three observations.

Unnormalized training shows higher teacher and student log-probabilities. Counterintuitively, models trained without renormalization exhibit *higher* teacher and student log-probabilities throughout training. We attribute this to an artifact of the truncation: without renormalization, the top-K probabilities do not sum to one, which effectively inflates the absolute probability of each candidate token. During rollout sampling this inflation makes it *easier* to draw low-probability tokens, distorting the on-policy distribution and masking the true quality of the supervision signal.

Performance improves with larger top-K but never matches renormalized training. Among the unnormalized variants, increasing the total top-K size (top-5 $\rightarrow$ top-10) improves performance, suggesting that a wider candidate set provides more signal. However, even at the same total top-K count, unnormalized training consistently underperforms its renormalized counterpart. This is because without renormalization, all probability mass assigned to tokens *outside* the top-K union is completely invisible to the KL loss—the model can shift mass to out-of-vocabulary positions without incurring any penalty.

Renormalization closes an optimization loophole. Without renormalization, the model can learn a degenerate strategy: push probability mass outside the top-K support (escaping KL penalty entirely), while simultaneously lowering the absolute probabilities of tokens within the top-K set, making the measured KL appear small. With renormalization, any mass that leaks outside the top-K union is implicitly redistributed back by the normalization step—the normalized probabilities within the top-K set rise whenever out-of-support mass increases, removing any incentive for this exploit and forcing the student to genuinely match the teacher on the retained candidates.

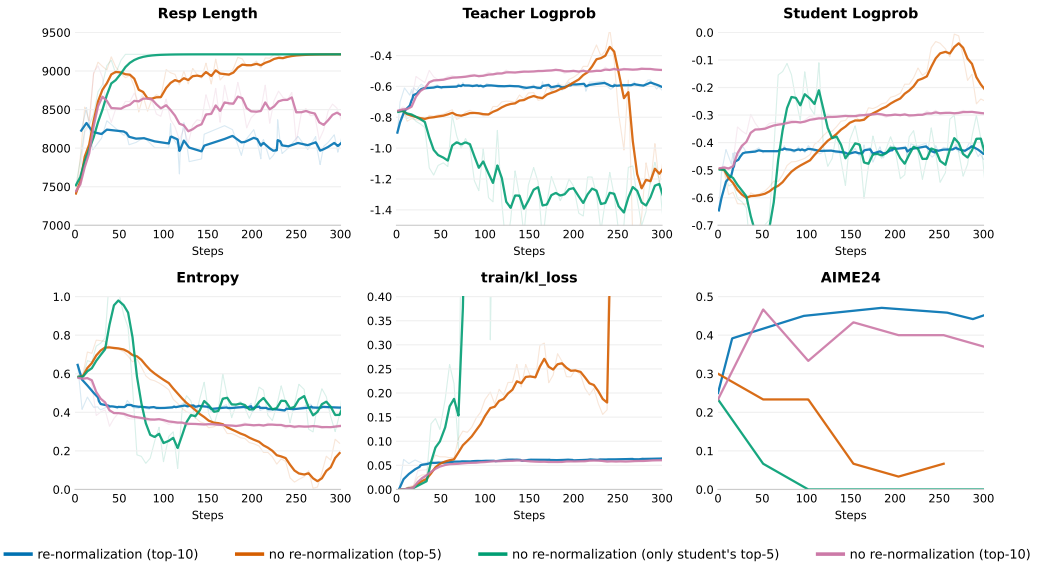


Figure 7: Effect of renormalization in top-K distillation training (Qwen3-1.7B student). Training curves and AIME-24 performance for OPD (topk) with renormalization vs. three unnormalized variants (top-5, top-10, student-only top-5). Without renormalization, training is unstable and final performance degrades substantially.

B.3 Effect of Rollout Temperature

All main experiments fix the rollout temperature to 1.0 for both the student and the teacher. We additionally evaluate two non-unit configurations: (1) $t_{\text{student}} = 1, t_{\text{teacher}} = 0.2$ (sharpened teacher distribution); (2) $t_{\text{student}} = 0.8, t_{\text{teacher}} = 0.8$ (symmetric low temperature).

$t_{\text{student}} = 1, t_{\text{teacher}} = 0.2$**.** A lower teacher temperature concentrates teacher probability mass sharply on its top tokens. Intuitively this sharpens the supervision signal, but it also increases the mismatch between the temperature-conditioned teacher logits and the student's on-policy distribution, causing training instability.

$t_{\text{student}} = 0.8, t_{\text{teacher}} = 0.8$**.** Lowering both temperatures symmetrically reduces generation diversity and dilutes the discriminative signal between high- and low-confidence token choices, weakening the confidence reward.

Figure 8 shows training curves and final evaluation scores under both settings compared to the default temperature-1.0 baseline. Both non-unit configurations degrade performance, confirming that the symmetric unit temperature is the best operating point and that the SGLang prefill temperature-scaling patch (Appendix A) is not needed in the final system.

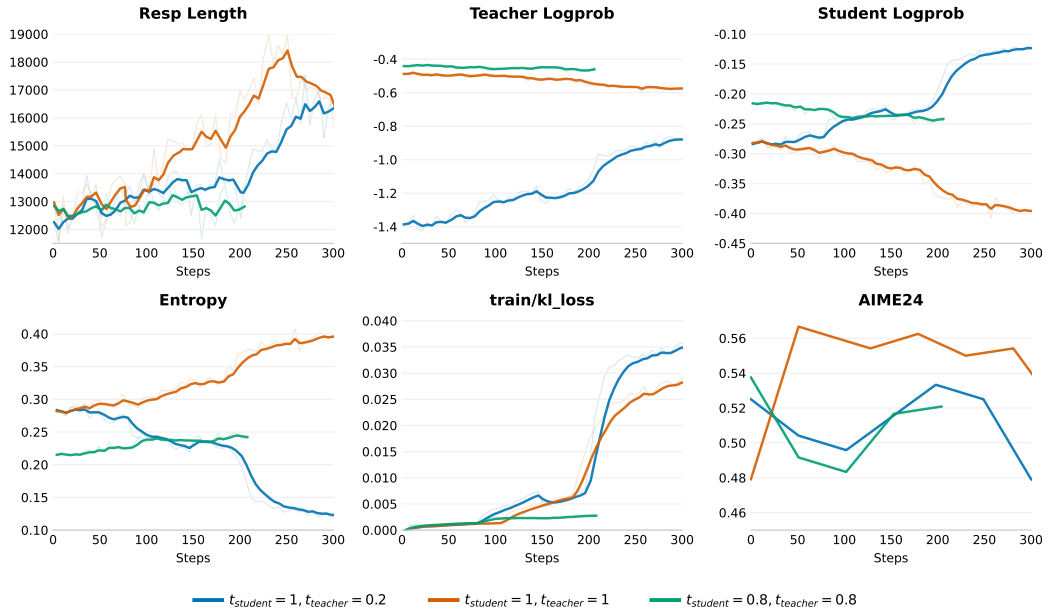


Figure 8: Effect of rollout temperature on training dynamics and final AIME-24 performance (Qwen3-1.7B student). Conditions: $t_{\text{student}} = 1, t_{\text{teacher}} = 1$ (default), $t_{\text{student}} = 1, t_{\text{teacher}} = 0.2$ (sharpened teacher), and $t_{\text{student}} = 0.8, t_{\text{teacher}} = 0.8$ (symmetric low temperature). Both non-unit configurations degrade performance, confirming that symmetric unit temperature is the optimal operating point.

B.4 Per-Token Top-K Logit Visualization

To understand *why* per-token RKL remains stubbornly high at certain positions even after an OPD gradient step, we visualize the full top-K logit distribution of the student at individual tokens, comparing the distributions before and after each optimization step.

High-RKL tokens have dispersed student top-K logits. Figure 9 shows representative tokens from a student rollout late in training. For tokens where the RKL is large and remains large after the update step, the student's top-K probability mass is spread relatively uniformly across many candidates—the student is genuinely uncertain, and no single token dominates. The teacher, by contrast, concentrates mass sharply on one or two tokens. The gradient step reduces the KL slightly but cannot collapse the student distribution in a single step given the flat landscape.

Low-RKL tokens have logits concentrated at top-1. Tokens that achieve low RKL exhibit the opposite pattern: the student distribution is already sharply peaked at the same top-1 token as the teacher. These tokens contribute near-zero loss and near-zero gradient.

Persistent high-RKL late in training. As training progresses, the proportion of low-RKL (peaked) tokens grows, but a residual population of high-RKL (dispersed) tokens persists and proves resistant

to further optimization. Crucially, in the standard per-token average loss, the large and growing majority of low-RKL tokens effectively act as a low-magnitude denominator that dilutes the gradient contribution of the remaining high-RKL tokens. Each high-RKL token's gradient is upweighted in absolute terms but its relative influence in the batch average decreases as low-RKL tokens accumulate—creating a natural but undesirable gradient imbalance.



Figure 9: Per-token top-K student logit distributions before and after an OPD update step for four representative tokens from a student rollout late in training (DeepSeek-R1-Distill-Qwen-1.5B student). Each panel shows the token context, student top-K logits and probabilities, teacher top-K logits, and the per-token KL divergence before and after the gradient step. High-RKL tokens exhibit dispersed student probability mass across many candidates, while the teacher concentrates sharply on one or two tokens; the gradient step reduces the KL only marginally, consistent with the flat optimization landscape described in Section B.4.

B.5 Future-KL with GAE Weighting

Motivated by the credit assignment literature in RL, we explore an alternative to the local per-token RKL loss: rather than penalizing the student only for the divergence at position t, we compute a *future-KL* signal by aggregating the RKL over all future tokens $t' > t$ and weighting them with a generalized advantage estimate (GAE, λ-return) decay:

$$\mathcal{L}^{(t)}_{\text{future-KL}} = \sum_{t'=t}^{T} (\rho\lambda)^{t'-t}\, \mathrm{RKL}(\pi_\theta(\cdot \mid x_{<t'}) \,\|\, \pi_T(\cdot \mid x_{<t'}))\,, \tag{13}$$

where ρ is a discount factor and λ is the GAE trace-decay parameter. **Note that this λ is the GAE eligibility-trace coefficient and is distinct from the confidence reward weight γ in LGR; the two hyperparameters play entirely different roles.** Intuitively, this encourages the student to make choices at position t that lead to lower KL *throughout* the trajectory, not just locally—which could in principle counteract the self-reinforcing drift described in Proposition 2.

Figure 10 compares future-KL distillation against standard OPD under a sweep of (ρ, λ) pairs. Despite the appealing intuition, future-KL consistently fails to improve over standard per-token RKL. We hypothesize that the difficulty lies in the credit assignment itself: in long reasoning chains,

the future-KL signal at early positions is dominated by the noise accumulated over hundreds of subsequent tokens, making the gradient at position t effectively uninformative about the local decision quality. This suggests that the right way to address long-horizon supervision degradation is through the one-step-ahead lookahead used in LGR, rather than through discounted future aggregation.

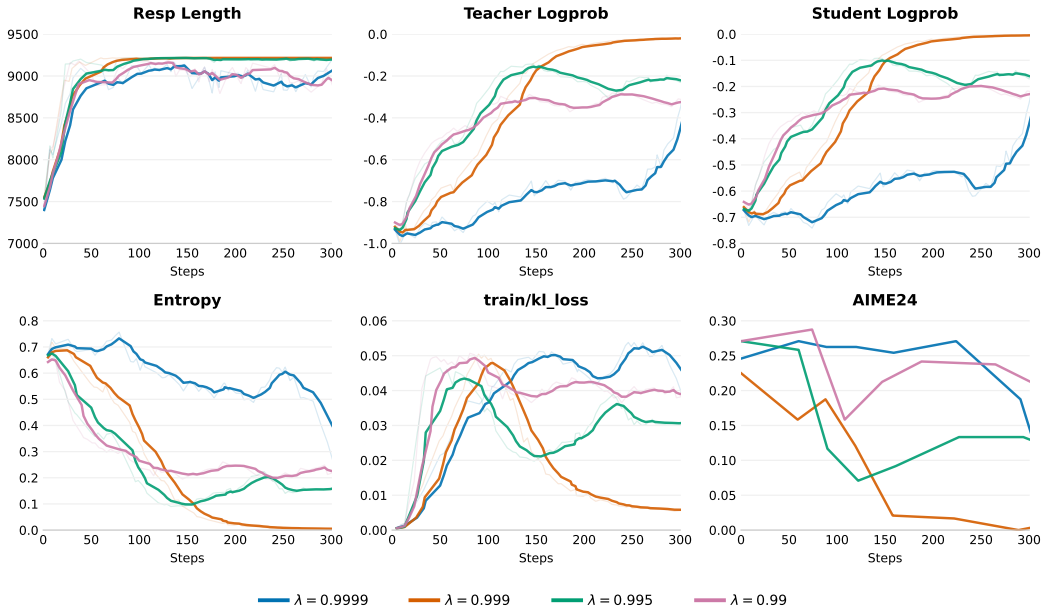


Figure 10: Future-KL distillation with GAE weighting across discount factors $\gamma \in \{0.9999, 0.999, 0.995, 0.99\}$, compared to standard per-token OPD (DeepSeek-R1-Distill-Qwen-1.5B student). Despite varying the discount factor across a wide range, future-KL consistently fails to improve over the per-token RKL baseline on AIME-24, with more aggressive discounting ($\gamma = 0.999, 0.995$) causing entropy collapse and severe performance degradation.

B.6 Comparison of KL Divergence Objectives

We compare three on-policy distillation objectives that differ in the direction of the KL divergence: reverse-KL (OPD/RKLD), forward-KL (FKLD), and Jensen–Shannon divergence (JSD). Figure 11 tracks training dynamics and AIME-24 performance across all three.

Forward-KL collapses. FKLD training is highly unstable: teacher log-probability degrades sharply after $\sim$50 steps, entropy explodes, and AIME-24 performance drops toward zero and does not recover. This confirms the exposure bias problem of forward-KL in the on-policy setting—the student is forced to cover the full teacher distribution using its own generated prefixes, leading to mode-covering behavior that pushes the student out of distribution.

JSD provides moderate stability but underperforms OPD. JSD achieves intermediate stability: entropy grows moderately and teacher log-probability declines more gradually than FKLD. However, final AIME-24 performance remains below OPD(RKLD), consistent with the main table results. The mixture objective partially inherits the forward-KL instability without the full benefits of reverse-KL's mode-seeking behavior.

OPD (RKLD) is the strongest baseline. Reverse-KL maintains the lowest distill KL, highest teacher log-probability, and stable entropy throughout training, confirming it as the appropriate base objective—and motivating LGR as an enhancement of OPD rather than a replacement of the divergence choice.

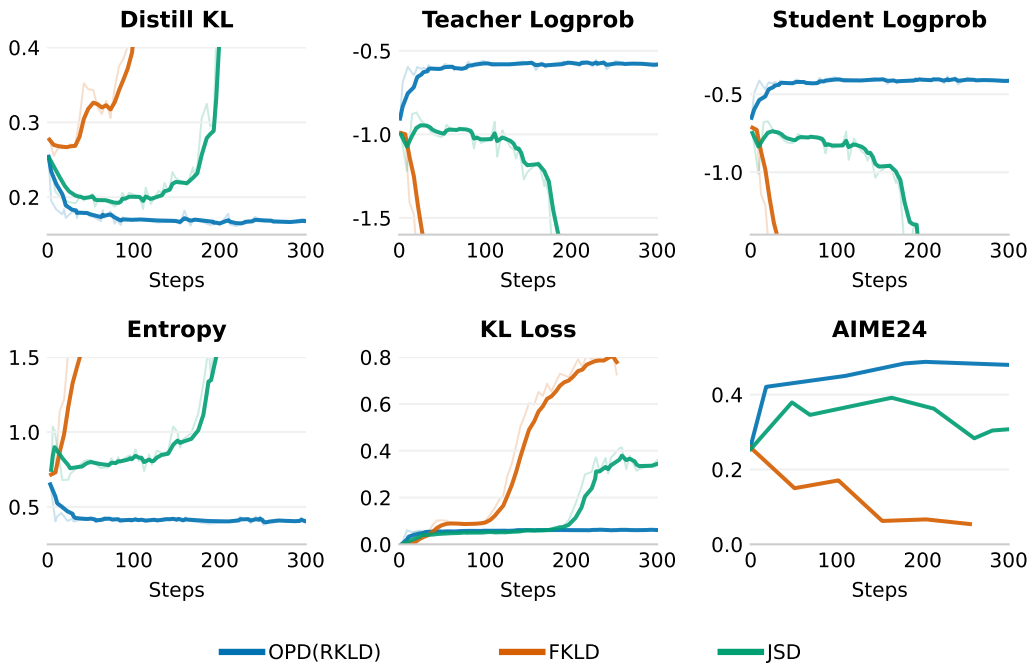


Figure 11: Training dynamics and AIME-24 performance for three KL divergence objectives (DeepSeek-R1-Distill-Qwen-1.5B student): OPD (reverse-KL), forward-KL (FKLD), and Jensen–Shannon divergence (JSD). FKLD collapses due to exposure bias; JSD is moderately stable but underperforms OPD; reverse-KL achieves the best training stability and final performance.

C Sigmoid Fit of the SFD Curve

Section 2.4 predicts that the SFD curve $\mathcal{C}^{(t)}$ follows a sigmoidal decay, and that OPD training causes the transition point to shift leftward (supervision boundary contraction). We verify this empirically by fitting the teacher completion accuracy curves from Figure 2 to a parametric sigmoid:

$$\mathcal{C}^{(t)} = \frac{A}{1 + e^{\,\varepsilon\,(t - t^*)}} + b, \tag{14}$$

where A is the decay amplitude, ε is the transition steepness, t^* is the midpoint of the transition (supervision boundary), and b is the residual supervision floor.

Fitted parameters. Table 5 reports the fitted parameters for the base student (before OPD) and the OPD-trained student (after OPD), along with goodness-of-fit R^2.

Table 5: Sigmoid fit parameters for the SFD curve before and after OPD training.

	A	ε	t^* (k tokens)	b	Upper asymptote	R^2
Before OPD ($+$)	0.3412	0.2145	7.43	0.4133	0.75	0.9980
After OPD ($\times$)	0.5452	0.2608	7.04	0.2294	0.77	0.9977
Δ	$+0.20$	$+0.046$	-0.39	-0.18	—	—

Interpretation. All four parameter shifts are consistent with the vicious cycle prediction from Proposition 2:

- t^* **shifts left** ($7.43 \rightarrow \mathbf{7.04}$k, $\Delta = -0.39$k): the supervision boundary contracts after OPD training, confirming optimized trajectories push the teacher into OOD regions earlier.
- ε **increases** ($0.21 \rightarrow \mathbf{0.26}$, $\Delta = +0.046$): the transition steepens, meaning supervision quality collapses more abruptly once the boundary is crossed.

- **Floor b drops** ($0.41 \to \mathbf{0.23}$, $\Delta = -0.18$): residual supervision quality at long prefixes degrades substantially after OPD training.
- **Amplitude A increases** ($0.34 \to \mathbf{0.55}$, $\Delta = +0.20$): total supervision loss is larger post-training, indicating deeper drift into the teacher's incompetent region.

These two snapshots (before/after training) do not constitute a full training trajectory, so we do not claim precise estimates of the logistic growth parameters. However, all four parameter shifts are in the direction predicted by the vicious cycle analysis, providing empirical support for the sigmoidal SFD structure described in Section 2.4.

D Entropy-Triggered Activation is Near-Lossless

Observation 1 (Entropy-triggered activation is near-lossless)**.** *At low-entropy positions ($\mathcal{H}(\pi_\theta(\cdot|\mathbf{x}_{<t})) \leq \tau$), the student's top-1 candidate dominates: $\pi_\theta(x_t^{(1)}) \gg \pi_\theta(x_t^{(k)})$ for $k \geq 2$. The combined loss in Eq.* (12) *is:*

$$\mathcal{L}_t^{LGR} = \sum_{k=1}^{K} \pi_\theta(x_t^{(k)}|\mathbf{x}_{<t}) \cdot \left[A_t^{(k)} + \gamma \cdot r_{\text{conf}}(x_t^{(k)})\right].$$

When $\pi_\theta(x_t^{(1)}) \approx 1$ and $\pi_\theta(x_t^{(k)}) \approx 0$ for $k \geq 2$, this reduces to $\mathcal{L}_t^{LGR} \approx A_t^{(1)} + \gamma \cdot r_{\text{conf}}(x_t^{(1)})$. Since $r_{\text{conf}}(x_t^{(1)})$ is a single-sample reward with near-zero gradient contribution when $\pi_\theta(x_t^{(1)}) \approx 1$ (the student has already committed to this token with probability 1), the confidence reward term adds negligible signal. Formally, the gradient of the confidence reward term with respect to θ is:

$$\nabla_\theta \left[\pi_\theta(x_t^{(1)}) \cdot r_{\text{conf}}(x_t^{(1)})\right] = r_{\text{conf}}(x_t^{(1)}) \cdot \nabla_\theta \pi_\theta(x_t^{(1)}) \approx r_{\text{conf}}(x_t^{(1)}) \cdot 0 = 0,$$

since $\pi_\theta(x_t^{(1)}) \approx 1$ implies $\nabla_\theta \pi_\theta(x_t^{(1)}) \approx 0$ (the probability is saturated). Disabling the confidence reward at low-entropy positions therefore incurs near-zero information loss while saving K teacher forward-pass evaluations per position.

E Tree Attention: Cost Analysis and Practical Segmentation

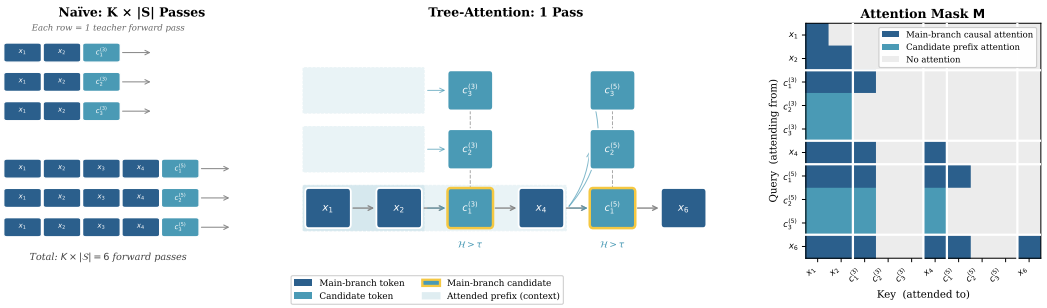


Figure 12: **Tree attention for efficient confidence-reward computation.** *Left*: the naïve approach requires $K \times |\mathcal{S}|$ separate teacher forward passes—one per candidate per high-entropy position. *Middle*: tree-attention construction. At each high-entropy position $t \in \mathcal{S}$, the top-K candidate tokens are appended as branches off the main sequence. Main-branch tokens attend causally; each candidate attends only to its own prefix $\mathbf{x}_{<t}$ (shaded region). *Right*: the resulting sparse attention mask $\mathbf{M}$, reducing $K \times |\mathcal{S}|$ passes to a single teacher forward pass.

Baseline cost (naïve prefill). Without tree attention, evaluating candidate $x_t^{(k)}$ requires prefilling the context $x_1, \ldots, x_{t-1}, x_t^{(k)}$ from scratch—the main sequence KV cache cannot be reused because positions $t \in \mathcal{S}$ have different prefix lengths. The prefill cost for a sequence of length t is $O(t^2 d)$, giving total cost:

$$C_{\text{naïve}} = K \sum_{t \in \mathcal{S}} t^2 d \;\approx\; K \cdot |\mathcal{S}| \cdot \frac{L^2}{3} \cdot d \;=\; \frac{0.2K}{3} L^3 d,$$

where we assume positions in $\mathcal{S}$ are roughly uniform over $[0, L]$, giving $\sum_{t\in\mathcal{S}} t^2 \approx |\mathcal{S}| \cdot L^2/3$. This is $O(L^3)$, cubically expensive.

Tree attention construction. Given $\mathbf{x} = (x_1, \ldots, x_L)$ and $\mathcal{S}$, we append all $K|\mathcal{S}|$ candidate tokens after the main sequence and apply a tree-structured mask $\mathbf{M}$:

- **Main tokens** $(1, \ldots, L)$: standard causal mask.
- **Candidate** $x_t^{(k)}$: attends to $\mathbf{x}_{\leq t-1}$ and itself only.

The main sequence is prefilled once (cost $O(L^2 d)$); each candidate is then a single decode step attending to $t-1$ cached keys (cost $O(t\,d)$ per candidate). Total:

$$C_{\text{tree}} = L^2 d + K \sum_{t\in\mathcal{S}} t\,d \;\approx\; L^2 d + K \cdot |\mathcal{S}| \cdot \frac{L}{2} \cdot d \;=\; L^2 d\,(1 + 0.1K).$$

This is $O(L^2)$—one order of magnitude cheaper than the naïve baseline.

Speedup.

$$\text{Speedup} = \frac{C_{\text{naïve}}}{C_{\text{tree}}} = \frac{0.2KL/3}{1 + 0.1K}.$$

The speedup grows *linearly* with L because naïve scales as $O(L^3)$ while tree attention scales as $O(L^2)$. For $K = 8$, $L = 16$k: Speedup $\approx \mathbf{4741}\times$.

Practical segmentation. In practice, the total sequence length $L + K|\mathcal{S}|/N$ must fit within GPU memory $L_{\max}^{\text{GPU}}$, requiring at least $N^* = \lceil K|\mathcal{S}|/(L_{\max}^{\text{GPU}} - L)\rceil$ segments. With N segments, the main sequence is re-prefilled N times:

$$C_{\text{tree},N} = N \cdot L^2 d + K \sum_{t\in\mathcal{S}} t\,d \approx L^2 d\,(N + 0.1K), \qquad \text{Speedup}_N = \frac{0.2KL/3}{N + 0.1K}.$$

In our experiments ($L = 16$k, $K = 8$, $|\mathcal{S}| \approx 3.2$k), we use $N = 8$ segments to stay within GPU memory, giving:

$$\text{Speedup}_{N=8} = \frac{0.2 \cdot 8 \cdot 16000/3}{8 + 0.1 \cdot 8} \approx \frac{8533}{8.8} \approx 970\times.$$

Even with segmentation, the speedup remains large because the $O(L^3)$ vs. $O(L^2)$ gap dominates.

F Proofs of Theoretical Results

F.1 Proof of Proposition 1

Proposition (Teacher signal vanishing under SFD, restated)**.** *Decompose the per-token advantage as $A_t = (1 + \log \pi_\theta(x_t|\mathbf{x}_{<t})) - \log \pi_T(x_t|\mathbf{x}_{<t})$. Define the teacher's discriminative signal as $\Delta_T(t) \coloneqq \mathrm{Var}_{x_t\sim\pi_\theta}[\log \pi_T(x_t|\mathbf{x}_{<t})]$. Then:*

1. *When $\pi_T(\cdot|\mathbf{x}_{<t}) = \mathrm{Uniform}(\mathcal{V})$, we have $\Delta_T(t) = 0$ and A_t depends only on the student.*
2. *$\Delta_T(t) \leq \log^2 |\mathcal{V}| - \mathrm{Ent}(\pi_T(\cdot|\mathbf{x}_{<t}))^2$.*
3. *$\mathrm{SNR}_T(t) = O(\Delta_T(t))$ and decreases monotonically under SFD.*

Proof. **Part (1).** When $\pi_T(\cdot|\mathbf{x}_{<t}) = \mathrm{Uniform}(\mathcal{V})$, we have $\pi_T(x_t|\mathbf{x}_{<t}) = 1/|\mathcal{V}|$ for every $x_t \in \mathcal{V}$, so $\log \pi_T(x_t|\mathbf{x}_{<t}) = -\log|\mathcal{V}|$ is a constant independent of x_t. Therefore $\Delta_T(t) = \mathrm{Var}_{x_t}[-\log|\mathcal{V}|] = 0$, and:

$$A_t = 1 + \log \pi_\theta(x_t|\mathbf{x}_{<t}) - (-\log|\mathcal{V}|) = 1 + \log \pi_\theta(x_t|\mathbf{x}_{<t}) + \log|\mathcal{V}|.$$

This depends only on π_θ and the constant $\log|\mathcal{V}|$, providing no teacher correction signal.

Part (2). Let $P = \pi_T(\cdot|\mathbf{x}_{<t})$ be a distribution over $\mathcal{V}$. Since $P(v) \in (0, 1]$ for all v, we have $\log P(v) \in [-\log|\mathcal{V}|, 0]$. Define the random variable $Z = \log P(x_t)$ where $x_t \sim \pi_\theta(\cdot|\mathbf{x}_{<t})$. We bound $\mathrm{Var}[Z]$ by bounding the second moment.

By the Bhatia–Davis inequality [41], for a bounded random variable $Z \in [a, b]$:

$$\mathrm{Var}[Z] \leq (b - \mathbb{E}[Z])(\mathbb{E}[Z] - a).$$

Here $a = -\log|\mathcal{V}|$, $b = 0$. The expected value satisfies:

$$\mathbb{E}_{x_t \sim \pi_\theta}[\log P(x_t)] = \sum_v \pi_\theta(v) \log P(v).$$

In the worst case (maximizing variance), this equals the cross-entropy $H(\pi_\theta, P)$. Applying the AM-GM inequality to the Bhatia–Davis bound:

$$\mathrm{Var}[Z] \leq \frac{(b-a)^2}{4} = \frac{\log^2|\mathcal{V}|}{4}.$$

A tighter bound exploiting the entropy of P proceeds as follows. Note that when P is uniform, $\log P(v) = -\log|\mathcal{V}|$ for all v, so $\mathrm{Var}[Z] = 0$ regardless of π_θ. As P becomes more peaked (lower entropy), the spread of $\log P(v)$ over the vocabulary increases, allowing $\Delta_T(t)$ to grow. Using the variance-entropy relationship for log-probabilities (a standard result in information theory [42]):

$$\mathrm{Var}_{x_t \sim \pi_\theta}[\log P(x_t)] \leq \log^2|\mathcal{V}| - \mathrm{Ent}(P)^2,$$

where equality holds when P is supported on a single token (maximum peakedness). As $\mathrm{Ent}(P) \to \log|\mathcal{V}|$ (P becomes uniform), $\Delta_T(t) \leq \log^2|\mathcal{V}| - \log^2|\mathcal{V}| = 0$, confirming $\Delta_T(t) \to 0$.

Part (3). The gradient direction attributable to the teacher is determined by the variation of $\log \pi_T(x_t)$ across token choices. Specifically, the teacher's contribution to the policy gradient is:

$$g_T = \mathbb{E}_{x_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(x_t) \cdot \log \pi_T(x_t)]\,.$$

The signal-to-noise ratio of this term scales as the standard deviation of $\log \pi_T(x_t)$ divided by the standard deviation of $\nabla_\theta \log \pi_\theta(x_t)$, i.e., $\mathrm{SNR}_T(t) \propto \sqrt{\Delta_T(t)}$. Under SFD, as the student's prefix drifts further from the teacher's training distribution, $\mathrm{Ent}(\pi_T(\cdot|\mathbf{x}_{<t}))$ increases monotonically (the teacher's distribution becomes more diffuse), so $\Delta_T(t)$ decreases monotonically by Part (2), and $\mathrm{SNR}_T(t) = O(\Delta_T(t)) \to 0$. □

F.2 Proof Sketch of Proposition 2

Proposition (Self-reinforcing drift under reverse-KL, restated)**.** *Define distributional drift $d_t := D(\pi_T(\cdot|\mathbf{x}^\theta_{<t}), \pi_T(\cdot|\mathbf{x}^*_{<t}))$, and assume $\Delta_T(t)$ is non-increasing in d_t (greater drift degrades teacher discriminability). Under reverse-KL with stop-gradient:*

1. *When $\Delta_T(t) = 0$, the advantage $A_t = 1 + \log \pi_\theta(x_t) + \log|\mathcal{V}|$ reinforces the student's existing mode without teacher correction.*
2. *When $\Delta_T(t) < \delta_{\mathrm{crit}}$, $\mathbb{E}[d_{t+1}|d_t] \geq d_t$, creating a positive feedback loop.*
3. *Forward-KL avoids SFD by construction but introduces exposure bias.*

Proof sketch. **Part (1).** From Part (1) of Proposition 1, when $\Delta_T(t) = 0$:

$$A_t = 1 + \log \pi_\theta(x_t|\mathbf{x}_{<t}) + \log|\mathcal{V}|.$$

Under gradient descent on $\mathcal{L}_{\mathrm{rkl}}$, the update to $\pi_\theta(x_t)$ is proportional to $-A_t$. Since A_t is an increasing function of $\log \pi_\theta(x_t)$, the gradient is negative (decreasing $\pi_\theta(x_t)$) when $\pi_\theta(x_t) > e^{-(1+\log|\mathcal{V}|)} = (e \cdot |\mathcal{V}|)^{-1}$. For any token x_t assigned probability above this threshold (which holds for the student's top tokens whenever the distribution is non-uniform), the gradient reinforces the student's high-probability tokens. Specifically, high-probability tokens have $A_t \gg 0$, receiving strong gradient, while low-probability tokens have $A_t < 0$ and are further suppressed—sharpening the distribution toward the student's existing mode without any teacher guidance.

Part (2). By assumption, $\Delta_T(t) \leq f(d_t)$ with f decreasing. When $\Delta_T(t) < \delta_{\mathrm{crit}}$, the teacher contributes negligible correction signal (Proposition 1, Part 3). Therefore, the token x_t selected at step t is drawn from the student's sharpened distribution $\pi_\theta(\cdot|\mathbf{x}_{<t})$ rather than being guided toward the teacher's preferred continuation. Let $x^*_t \in \arg\max_v \pi_T(v|\mathbf{x}^*_{<t})$ be the teacher's most likely

token at the corresponding teacher-generated position. Since the student selects from its own mode, we have $x_t \neq x_t^*$ with probability bounded away from zero. Appending the divergent token x_t to the context shifts the student's prefix further from the teacher's distribution:

$$d_{t+1} = D(\pi_T(\cdot|\mathbf{x}^\theta_{\leq t}), \pi_T(\cdot|\mathbf{x}^*_{\leq t})) \geq d_t + \delta$$

for some $\delta > 0$ depending on the token divergence. This establishes the positive feedback loop: d_t increases whenever $\Delta_T(t)$ is below the critical threshold, and increasing d_t further reduces $\Delta_T(t)$, sustaining the loop.

Part (3). Under forward-KL (off-policy), the training sequences are teacher-generated: $\mathbf{x} \sim \pi_T(\cdot|\mathbf{c})$. Therefore $\mathbf{x}^\theta_{<t} = \mathbf{x}^*_{<t}$ for all t, and $d_t = D(\pi_T(\cdot|\mathbf{x}^*_{<t}), \pi_T(\cdot|\mathbf{x}^*_{<t})) = 0$ throughout. SFD does not arise by construction. The cost is that at inference time the student generates $\mathbf{x} \sim \pi_\theta$, but during training it always conditioned on teacher-generated prefixes $\mathbf{x} \sim \pi_T$—a train-test mismatch known as exposure bias. □

F.3 Proof of Proposition 3

Proposition (One-step-ahead discriminability, restated)**.** *Define* $D_{\text{ahead}}(t) := \max_{k,k'} \left|\max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k)}) - \max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k')})\right|$. *Then* $D_{\text{ahead}}(t)$ *is independent of* $\text{Ent}(\pi_T(\cdot|\mathbf{x}_{<t}))$*: even when* $\pi_T(\cdot|\mathbf{x}_{<t})$ *is uniform (*$\Delta_T(t) = 0$*),* $D_{\text{ahead}}(t)$ *can be arbitrarily large.*

Proof. We prove by explicit construction. Let $\mathcal{V} = \{a, b, c\}$ and suppose $\pi_T(\cdot|\mathbf{x}_{<t}) = (1/3, 1/3, 1/3)$, i.e., perfectly uniform at position t with $\text{Ent}(\pi_T(\cdot|\mathbf{x}_{<t})) = \log 3$ (maximum entropy, $\Delta_T(t) = 0$). Assign the following teacher distributions at position $t{+}1$:

$$\begin{aligned}
\pi_T(\cdot|\mathbf{x}_{<t}, a) &= (1 - 2\varepsilon,\ \varepsilon,\ \varepsilon), \quad \max_v = 1 - 2\varepsilon, \\
\pi_T(\cdot|\mathbf{x}_{<t}, b) &= (1/3,\ 1/3,\ 1/3), \quad \max_v = 1/3, \\
\pi_T(\cdot|\mathbf{x}_{<t}, c) &= (1/3,\ 1/3,\ 1/3), \quad \max_v = 1/3,
\end{aligned}$$

for any $\varepsilon \in (0, 1/3)$. Then:

$$D_{\text{ahead}}(t) = (1 - 2\varepsilon) - \frac{1}{3} = \frac{2}{3} - 2\varepsilon.$$

As $\varepsilon \to 0$, $D_{\text{ahead}}(t) \to 2/3$, which is arbitrarily large relative to the uniform bound. This construction is valid for any vocabulary size $|\mathcal{V}| \geq 2$ and any level of local entropy $\text{Ent}(\pi_T(\cdot|\mathbf{x}_{<t})) = \log|\mathcal{V}|$ (maximum entropy at position t). Thus $D_{\text{ahead}}(t)$ is not bounded by $\text{Ent}(\pi_T(\cdot|\mathbf{x}_{<t}))$, confirming independence. The intuition is that $\pi_T(\cdot|\mathbf{x}_{<t})$ is a marginal distribution obtained by integrating over the next token; its entropy characterizes uncertainty *about the current position*, while $D_{\text{ahead}}(t)$ captures *differences across branching futures*—orthogonal quantities. □

F.4 Proof of Proposition 4

Proposition (Max-p as relative drift indicator, restated)**.** *Let* P_T^* *be the teacher's distribution on an in-distribution prefix, and* $P_T^{(\mathbf{x})}$ *on a student-generated prefix. If the teacher is* β*-smooth, then:*

$$\max_v P_T^*(v) - \max_v P_T^{(\mathbf{x})}(v) \leq \|P_T^* - P_T^{(\mathbf{x})}\|_\infty \leq \beta \cdot d(\mathbf{x}, \mathcal{X}_T).$$

Proof. **First inequality.** For any two distributions f, g over a finite set $\mathcal{V}$:

$$\max_v f(v) - \max_v g(v) \leq \max_v [f(v) - g(v)] \leq \max_v |f(v) - g(v)| = \|f - g\|_\infty.$$

The first step uses $\max_v f(v) = f(v^*) \leq f(v^*) - g(v^*) + \max_v g(v)$ for the maximizer $v^* = \arg\max_v f(v)$, giving $\max_v f(v) - \max_v g(v) \leq f(v^*) - g(v^*)$. The second step replaces the signed difference with the absolute value, and the last step is the definition of ℓ^∞ norm.

Second inequality. By β-smoothness of the teacher model, small perturbations in input context produce bounded output distribution shifts: there exists $\beta > 0$ such that for any two prefixes $\mathbf{x}$ and $\mathbf{x}^*$:

$$\|\pi_T(\cdot|\mathbf{x}) - \pi_T(\cdot|\mathbf{x}^*)\|_\infty \leq \beta \cdot d(\mathbf{x}, \mathbf{x}^*).$$

Setting $\mathbf{x}^* \in \mathcal{X}_T$ (in-distribution prefix minimizing distance from $\mathbf{x}$) gives the result with $d(\mathbf{x}, \mathcal{X}_T) = \min_{\mathbf{x}^* \in \mathcal{X}_T} d(\mathbf{x}, \mathbf{x}^*)$.

Implication for relative comparison. Critically, this proposition is used in a *relative* sense: we compare max-p across K candidates $\{x_t^{(k)}\}$ at the same position. For candidates k and k':

$$\max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k)}) - \max_v \pi_T(v|\mathbf{x}_{<t}, x_t^{(k')}) \approx \beta \cdot [d(\mathbf{x}_{<t} \cdot x_t^{(k)}, \mathcal{X}_T) - d(\mathbf{x}_{<t} \cdot x_t^{(k')}, \mathcal{X}_T)].$$

Thus a higher max-p at $t{+}1$ for candidate k implies candidate k has caused less drift from the teacher's in-distribution manifold—not a return to in-distribution, but less *additional* drift. This is the "relative drift indicator" interpretation used in Section 3.2. □

F.5 Group Normalization: Design Rationale and Formal Properties

Why group normalization is necessary. The raw confidence $r_{\text{raw}}(x_t) = \max_v \pi_T(v|\mathbf{x}_{<t}, x_t)$ varies substantially across positions and tasks for reasons unrelated to the relative quality of token choices: (i) token frequency effects (common tokens systematically attract higher teacher probability), (ii) context difficulty (some prefixes are inherently harder to continue regardless of token choice), and (iii) vocabulary size variation across model configurations. Using raw max-p directly as a reward would introduce a high-variance baseline that dominates the gradient signal. Group normalization subtracts the group mean μ_K computed over the student's top-K candidates at the *same position and context*, canceling all position- and task-level confounds. Only the *relative ranking* across candidates at the same position survives, which is exactly the signal we need: which token choice causes the least additional drift.

Connection to GRPO. This normalization is inspired by Group Relative Policy Optimization (GRPO), which normalizes rewards within a group of sampled responses. Our group is defined over the top-K candidates at each position rather than over full response samples, enabling a per-token signal rather than a sequence-level reward.

Proposition (Properties of group normalization)**.** *The group-normalized confidence reward* $r_{\text{conf}}(x_t^{(k)}) = (r_{\text{raw}}^{(k)} - \mu_K)/(\sigma_K + \epsilon)$ *satisfies:*

1. ***Zero-mean:*** $\sum_{k=1}^K \pi_\theta(x_t^{(k)}) \cdot r_{\text{conf}}(x_t^{(k)}) \approx 0$ *when top-K probabilities are approximately equal.*
2. ***Graceful degradation:*** *When* $\sigma_K \to 0$, $r_{\text{conf}}(x_t^{(k)}) \to 0$ *for all k.*
3. ***Scale invariance:*** *The ranking by* r_{conf} *is invariant to affine transformations of* r_{raw}.

Proof. **Part (1).** By the definition of μ_K:

$$\sum_{k=1}^K (r_{\text{raw}}^{(k)} - \mu_K) = \sum_{k=1}^K r_{\text{raw}}^{(k)} - K\mu_K = 0.$$

When $\pi_\theta(x_t^{(k)}) \approx 1/K$ for all k (approximately uniform top-K):

$$\sum_{k=1}^K \pi_\theta(x_t^{(k)}) \cdot r_{\text{conf}}(x_t^{(k)}) \approx \frac{1}{K(\sigma_K + \epsilon)} \sum_{k=1}^K (r_{\text{raw}}^{(k)} - \mu_K) = 0.$$

For non-uniform π_θ, the weighted sum is not exactly zero but remains small: it equals $\text{Cov}_{k \sim \pi_\theta}(1, r_{\text{conf}}^{(k)}) = 0$ by the zero-mean property of $r_{\text{raw}}^{(k)} - \mu_K$ under uniform weighting.

Part (2). When all candidates lead to the same teacher confidence, $r_{\text{raw}}^{(k)} = c$ for all k, so $\mu_K = c$ and $\sigma_K = 0$. The numerator $r_{\text{raw}}^{(k)} - \mu_K = 0$ for all k, giving $r_{\text{conf}}(x_t^{(k)}) = 0/\epsilon = 0$. More generally, as $\sigma_K \to 0$, the numerators approach zero while the denominator is bounded below by

$\epsilon > 0$, so $r_{\text{conf}} \to 0$. This is the desired graceful degradation: at positions where the teacher cannot discriminate between candidates (all lead to equally uncertain teacher states), the confidence reward automatically suppresses itself without requiring external gating.

Part (3). Let $\tilde{r}_{\text{raw}}^{(k)} = \alpha r_{\text{raw}}^{(k)} + \beta$ for constants $\alpha > 0$, $\beta \in \mathbb{R}$. Then:

$$\tilde{\mu}_K = \alpha\mu_K + \beta, \quad \tilde{\sigma}_K = |\alpha|\sigma_K = \alpha\sigma_K \quad (\text{since } \alpha > 0).$$

Therefore:

$$\tilde{r}_{\text{conf}}^{(k)} = \frac{(\alpha r_{\text{raw}}^{(k)} + \beta) - (\alpha\mu_K + \beta)}{\alpha\sigma_K + \epsilon} = \frac{\alpha(r_{\text{raw}}^{(k)} - \mu_K)}{\alpha\sigma_K + \epsilon}.$$

For large σ_K (where ϵ is negligible), $\tilde{r}_{\text{conf}}^{(k)} \approx r_{\text{conf}}^{(k)}$, and the ranking is preserved. More precisely, for any k, k': $\tilde{r}_{\text{conf}}^{(k)} > \tilde{r}_{\text{conf}}^{(k')}$ iff $r_{\text{raw}}^{(k)} - \mu_K > r_{\text{raw}}^{(k')} - \mu_K$ (since $\alpha > 0$), iff $r_{\text{raw}}^{(k)} > r_{\text{raw}}^{(k')}$, iff $r_{\text{conf}}^{(k)} > r_{\text{conf}}^{(k')}$. This makes the reward robust to systematic shifts in absolute confidence level across positions and tasks. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction section strictly follow the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide proofs in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We summarize all the information for experimental reproduction in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code is provided in the anonymized link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We reported Avg@K and Pass@k for evaluations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide them in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow every aspect of the NeurIPS Code of Ethics in this research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact in Limitations.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite every paper of the existing assets we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLM for writing, editing and formatting.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.