INFLUENCE-DIRECTED DISTILLATION: SOLVING THE PASS@k BOTTLENECK IN SAMPLED-TOKEN ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sampled-token on-policy distillation (OPD) efficiently transfers capabilities from teacher to student using student-generated tokens, requiring teacher probabilities only for sampled tokens. Yet gains in pass@1 do not always translate into comparable gains in pass@k, leaving a pass@k bottleneck in distillation. We introduce *First-Order Local Entropy Influence*, which factorizes the first-order entropy change under a local logit update into the teacher–student log-probability gap and a student-side factor. Selection by its sign empirically steers aggregate training entropy. Motivated by this, we propose *Influence-Directed Adaptive On-Policy Distillation* (IDA-OPD): without requiring full-vocabulary Forward-KL objectives, it retains nonnegative-influence updates and shrinks negative-influence advantages adaptively with discrepancy, using only the teacher's sampled-token log-probability. Experiments on reasoning-oriented distillation show IDA-OPD consistently improves pass@k, narrows the pass@k gap to the teacher, matches the strongest teacher-informed methods with lower teacher-probability transfer and storage requirements, and broadly maintains vanilla OPD's pass@1, all without full-vocabulary teacher information.

1 INTRODUCTION

On-policy distillation (OPD) is increasingly central to large language model (LLM) post-training. It has been widely integrated into the post-training pipelines of recent large-scale systems, including GLM-5 (Zeng et al. 2026), KiMi (Team et al. 2025), and Qwen3 (Yang et al. 2025), yielding significant improvements. Its standard form, however, uses the teacher's full-vocabulary probability distribution, whose transfer and storage can be costly at scale; this motivates sampled-token OPD, which needs only the teacher's sampled-token probability and avoids transferring and storing full-vocabulary teacher logits (Fu et al., 2026; Jia et al., 2026).

However, sampled-token OPD can face a *pass@k bottleneck*: the student's pass@1 improves while its pass@k plateaus (Fu et al., 2026; Cui et al., 2025).

Prior work suggests that excessive entropy reduction in the student during OPD contributes to this bottleneck (Jin et al., 2026; Jia et al., 2026). Existing methods for mitigating this entropy reduction fall into two categories, which differ in their intervention signals and teacher-information requirements. Teacher-informed methods inject richer teacher signals to guide the student: divergence-based criteria target areas of high teacher-student disagreement (Xu et al., 2026; Wang et al., 2026), AOPD decouples exploration from imitation in these specific regions (Jia et al., 2026), and Entropy-Aware OPD introduces a forward-KL penalty for high-uncertainty tokens (Jin et al., 2026). These methods require transferring and storing top-K teacher probabilities at selected positions for Forward KL. In contrast, student-side methods retain the sampled-token teacher-information requirement while intervening on entropy. They achieve this by relaxing heavy-tailed credits (Ko et al., 2026), reinforcing negative samples (Zhu et al., 2026), adding an entropy bonus (Schulman et al., 2017), or adapting advantage shapes and entropy coefficients (Cheng et al., 2025; Zhang et al., 2025; Jiang et al., 2025; Yu et al., 2025). These approaches motivate examining the local entropy effect of sampled-token OPD to develop a selective intervention. This raises a question:

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

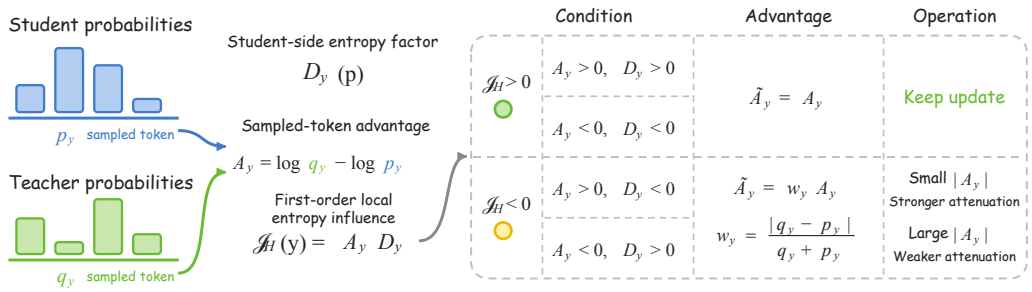


Figure 1: **Overview of IDA-OPD.** The sampled-token advantage A_y and the student-side factor D_y determine the first-order local entropy influence. Nonnegative-influence updates retain their advantage; negative-influence updates receive the retention weight w_y, weakening reinforcement or suppression without reversing direction. Only the sampled-token teacher probability is queried.

> *Can we overcome the pass@k bottleneck during OPD using only the teacher's sampled-token probability?*

Following prior work linking policy entropy to exploration and reasoning performance (Wang et al., 2025; Cui et al., 2025; Dai et al., 2026), we study how updates in sampled-token OPD affect the student's entropy change. Under a local logit-space update, we derive *First-Order Local Entropy Influence* $\mathcal{I}_H(y)$, which factorizes the first-order entropy change into the teacher–student log-probability difference, i.e., the advantage A_y, and a factor determined by the student's local probability structure. Under this approximation, scaling A_y scales the predicted entropy change, motivating selective intervention during training.

However, uniformly shrinking A_y at all negative-influence positions may weaken useful learning signals. This motivates a closer look at where the contraction originates to design a selective intervention. We measure the cumulative entropy loss across positions, and Figure 3 shows the contraction peaks sharply in the low-discrepancy region. This region also contains a large share of sampled tokens, associating low discrepancy with substantial cumulative entropy loss in our measurements.

Combining these two findings yields our method, *Influence-Directed Adaptive On-Policy Distillation* (IDA-OPD) (Figure 1). We retain nonnegative-influence updates and reweight the advantage A_y of each negative-influence update ($\mathcal{I}_H(y) < 0$) by a divergence-adaptive weight w_y: the attenuation is strongest in the low-discrepancy region, where we observe the largest cumulative entropy loss, and weakens as the discrepancy grows.

Extensive experiments on reasoning-oriented distillation demonstrate that IDA-OPD substantially mitigates the pass@k bottleneck. It curbs the entropy contraction and consistently improves pass@k over methods that operate under the same sampled-token budget, while remaining on par with the strongest teacher-informed methods with lower teacher-probability transfer and storage requirements—broadly maintaining pass@1 accuracy and using no full-vocabulary teacher information. Our analysis further shows that selection by the sign of $\mathcal{I}_H(y)$ can steer aggregate entropy in the evaluated training settings, and that our divergence-adaptive shrinkage targets the low-discrepancy region where entropy contraction concentrates.

Our core contributions are summarized as follows:

- We investigate the pass@k bottleneck through entropy dynamics and observe substantial cumulative entropy loss in the low-discrepancy region.
- We derive *First-Order Local Entropy Influence* $\mathcal{I}_H(y) = A_y D_y$ under a local first-order analysis, and show that selection by its sign can steer aggregate training entropy.
- We propose IDA-OPD, which reweights negative-influence advantages with a divergence-adaptive shrinkage weight w_y, improving pass@k over same-budget methods and matching the strongest teacher-informed ones without full-vocabulary logits or auxiliary FKL objectives.

2 PRELIMINARIES

We formalize the training process of on-policy distillation as follows. Let $\mathcal{V}$ denote the vocabulary and let h be a decoding prefix visited by a student rollout, with student and teacher next-token distributions $p_h(\cdot) = p_\theta(\cdot \mid h)$ and $q_h(\cdot) = p_{\mathrm{T}}(\cdot \mid h)$. OPD matches these distributions on student-induced prefixes by optimizing the reverse KL $\mathbb{E}_{h\sim p_\theta}[D_{\mathrm{KL}}(p_h \| q_h)]$.

Computing this divergence exactly requires full-vocabulary teacher probabilities at every prefix, increasing teacher-probability transfer and storage requirements. Since reverse KL is an expectation under the student distribution, sampled-token OPD or K1 estimator is often applied by drawing a single token from distribution $y \sim p_\theta(\cdot \mid h)$, which queries only the teacher log-probability of y (Fu et al., 2026; Jia et al., 2026). The advantage of the sampled token is

$$A_y = \log p_{\mathrm{T}}(y \mid h) - \log p_\theta(y \mid h),$$

and the vanilla sampled-token OPD loss is

$$\ell_y^{\mathrm{OPD}} = -\,\mathrm{sg}(A_y) \log p_\theta(y \mid h),$$

where $\mathrm{sg}(\cdot)$ is the stop-gradient operator. Although locally unbiased, this efficient estimator replaces the full teacher distribution with a noisy one-token signal, which can cause premature entropy collapse (Jin et al., 2026; Ko et al., 2026). Our work therefore aims to improve student pass@k while retaining the sampled-token teacher-information requirement.

3 UNDERSTANDING THE PASS@k BOTTLENECK

In this section, we investigate the pass@k bottleneck in sampled-token OPD through the lens of entropy. We first show that the advantage A_y alone does not determine the entropy effect of an update, which also depends critically on the student's current distribution. This observation motivates a first-order analysis, through which we derive First-Order Local Entropy Influence to characterize the local, first-order entropy effect of a sampled-token update.

3.1 ADVANTAGE ALONE DOES NOT DETERMINE ENTROPY

In the sampled-token OPD loss, A_y governs the learning signal for the sampled token: its sign determines whether y is reinforced or suppressed, while its magnitude controls the update strength. Accordingly, prior work has used A_y as a natural signal for analyzing and regulating OPD (Ko et al., 2026; Jia et al., 2026).

Figure 2 provides the underlying intuition. Consider two updates with the same positive advantage $A_y > 0$. Reinforcing a token that already has high probability further concentrates the distribution and decreases entropy, whereas reinforcing a low-probability token can spread probability mass more evenly and increase entropy. Thus, the entropy effect is jointly determined by A_y and the student's current distribution, motivating a more systematic analysis of their interaction.

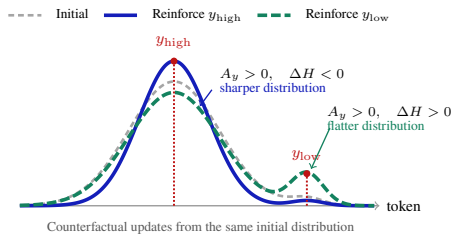


Figure 2: The same positive advantage sharpens a high-probability mode (lowering entropy) or promotes a low-probability alternative (raising entropy).

3.2 DIAGNOSTIC: FIRST-ORDER LOCAL ENTROPY INFLUENCE

Let $p = (p_i)_{i\in\mathcal{V}}$ be the student's current next-token distribution, where $p_i = p_\theta(i \mid h)$, with entropy $H(p) = -\sum_{i\in\mathcal{V}} p_i \log p_i$. Let p' denote the distribution after a small local sampled-token OPD update with step size η, and define the resulting local entropy change as

$$\Delta H = H(p') - H(p).$$

Our goal is to understand how the sampled-token advantage A_y and the student's current distribution jointly determine the sign and magnitude of ΔH. We characterize this interaction through a first-order expansion.

Theorem 1 (First-Order Local Entropy Influence)**.** *Under a local logit-space gradient step on the sampled-token OPD loss with learning rate η, the local entropy change satisfies*

$$\Delta H = \eta \cdot \mathcal{I}_H(y) + \mathcal{O}(\eta^2),$$

where we define the First-Order Local Entropy Influence *as $\mathcal{I}_H(y) = A_y D_y$. The entropy-direction factor is*

$$D_y = \sum_i p_i^2 \big(\log p_i + H(p)\big) - p_y \big(\log p_y + H(p)\big). \tag{1}$$

Proof sketch. A single sampled-token OPD update perturbs the logits along $\Delta z_k = \eta A_y(\mathbb{I}[k = y] - p_k)$, so the advantage A_y scales the entire update while its direction $(\mathbb{I}[k = y] - p_k)$ is fixed by the student. Propagating this through the softmax Jacobian $\partial p_i / \partial z_k = p_i(\mathbb{I}[i = k] - p_k)$ gives a first-order probability change in which A_y again factors out cleanly,

$$\Delta p_i = p_i' - p_i = \eta A_y \, p_i \big(\mathbb{I}[i = y] - p_i - p_y + S_2\big) + \mathcal{O}(\eta^2),$$

where $S_2 = \sum_k p_k^2$. Expanding the entropy and using $\sum_i \Delta p_i = 0$ gives $\Delta H = -\sum_i \log p_i \, \Delta p_i + \mathcal{O}(\eta^2)$. Substituting the probability change and collecting terms, the common factor ηA_y pulls out, leaving a purely student-side remainder D_y:

$$\Delta H = \eta \, A_y D_y + \mathcal{O}(\eta^2).$$

The full derivation is given in Appendix. □

Under the local update in Theorem 1, $A_y D_y$ governs ΔH to first order, with D_y determined by the student's current distribution. We call updates with $\mathcal{I}_H(y) < 0$ *negative-influence* and those with $\mathcal{I}_H(y) \geq 0$ *nonnegative-influence*. In full-model training, this local classification serves as a criterion for update selection.

4 Influence-Directed Adaptive On-Policy Distillation

Using the local proxy from Section 3, we select negative-influence updates for intervention. Uniformly penalizing them may weaken useful teacher signals, so we examine where cumulative entropy loss concentrates. Selective OPD commonly assumes the most consequential entropy shifts occur at high-divergence tokens where teacher and student sharply disagree (Xu et al., 2026; Wang et al., 2026; Jin et al., 2026). To test this, we bin sampled-token updates by the normalized discrepancy $\delta_y = \frac{q_h(y) - p_h(y)}{q_h(y) + p_h(y)} \in [-1, 1]$ and, within each bin, measure the cumulative entropy loss, the actual shift in policy entropy across genuine optimizer steps, not the first-order proxy, alongside the token count. This analysis uses intermediate checkpoints from the training runs described in Section 5.

Figure 3 shows a secondary cluster of entropy loss at the high-divergence negative tail ($\delta_y \approx -1.0$), but the dominant peak lies near agreement ($\delta_y \approx 0$). The token-count histogram shows that this region also contains a large population of sampled tokens. This observation identifies where cumulative entropy loss concentrates, without establishing the learning value of individual updates.

This concentration motivates stronger attenuation near agreement and weaker attenuation at larger discrepancies, rather than a uniform penalty on all negative-influence updates ($\mathcal{I}_H(y) < 0$). We use δ_y to implement this divergence-adaptive shrinkage and evaluate it empirically below.

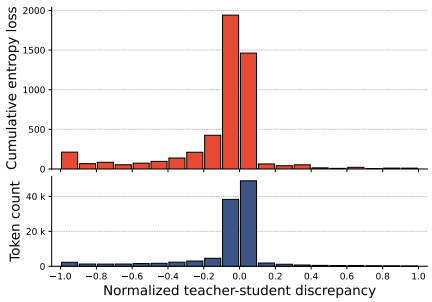


Figure 3: Cumulative entropy loss (top) and token counts (bottom) across normalized teacher–student discrepancy δ_y. The dominant contraction coincides with the large population near agreement.

Method	Math								Code			
	AIME24		AIME25		HMMT Feb		HMMT Nov		MBPP++		LCB	
	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16
Student	21.7	56.7	22.5	56.7	23.1	39.0	11.2	36.7	64.7	73.5	17.9	23.1
Teacher	53.3	80.0	53.5	73.3	26.7	53.5	35.4	63.3	70.5	74.6	28.0	36.1
+ OPD	54.6	78.7	51.8	65.7	**26.5**	51.7	31.2	58.3	69.5	73.3	28.4	34.8
+ REOPOLD	57.2	77.6	52.1	66.1	25.3	52.1	31.6	58.7	70.3	74.1	28.7	34.8
+ Entropy Bonus	55.4	78.2	51.4	66.3	25.6	53.5	31.0	58.9	69.6	74.4	28.4	34.9
+ Advantage Shaping	56.8	79.4	52.0	66.4	26.0	55.6	31.5	59.5	69.8	73.2	28.5	34.3
+ AOPD	**61.0**	81.1	52.4	69.7	26.1	57.4	30.3	59.4	**70.9**	72.8	**29.0**	34.2
+ EOPD	59.2	80.0	51.2	66.6	26.1	57.3	31.9	59.8	68.3	73.7	27.9	34.6
IDA-OPD	56.7	**83.3**	**53.3**	**70.0**	26.3	**60.0**	**32.3**	**60.1**	69.8	**74.8**	28.9	**36.3**
Δ vs OPD	$+2.1$	$+4.6$	$+1.5$	$+4.3$	-0.2	$+8.3$	$+1.1$	$+1.8$	$+0.3$	$+1.5$	$+0.5$	$+1.5$

Table 1: 4B distillation results on mathematics and code benchmarks, reporting pass@1 and pass@16. The student is Qwen3-4B-Non-Thinking; teachers are Qwen3-4B-Non-Thinking-RL-Math for Math and Qwen3-4B-Non-Thinking-RLCode for Code. Bold marks the best distilled-student result in each column. Δ reports absolute gains over OPD.

4.1 Solution: Divergence-Adaptive Shrinkage

We therefore propose *Influence-Directed Adaptive On-Policy Distillation* (IDA-OPD), which attenuates the advantage A_y at negative-influence positions ($\mathcal{I}_H(y) < 0$) according to the symmetric relative teacher–student disagreement at the sampled token,

$$w_y = \frac{|q_y - p_y|}{q_y + p_y} \in [0, 1). \tag{2}$$

Acting as a scale-free retention factor, w_y is small when teacher and student nearly agree and approaches 1 as their disagreement grows; it introduces no signal beyond A_y.

We use w_y to modulate the advantage at negative-influence positions:

$$\widetilde{A}_y = \begin{cases} A_y, & \mathcal{I}_H(y) \geq 0, \\ w_y\, A_y, & \mathcal{I}_H(y) < 0, \end{cases} \tag{3}$$

$$\ell_y^{\text{IDA-OPD}} = -\,\mathrm{sg}(\widetilde{A}_y) \log p_y. \tag{4}$$

By construction, this rule is sign-preserving, it never flips the direction of the teacher correction, and divergence-adaptive: it attenuates updates more strongly near agreement, where we observe substantial cumulative entropy loss, and less strongly as disagreement grows. Figure 1 illustrates how this rule retains nonnegative-influence updates and weakens reinforcement or suppression at negative-influence positions.

Proposition 1 (Quadratic attenuation near agreement)**.** *For every prefix h and sampled token y, $w_y\, A_y = \mathcal{O}(A_y^2)$ as $A_y \to 0$, and $w_y\, A_y \to A_y$ as $|A_y| \to \infty$.*

The proof, together with the closed-form identity relating w_y to A_y, is given in Appendix. Proposition 1 formalizes the two behaviors required from a divergence-adaptive shrinkage: near-quadratic attenuation near agreement and asymptotically lossless preservation as disagreement grows. The former targets the low-discrepancy region with the largest observed cumulative entropy loss. Consequently, IDA-OPD targets the pass@k bottleneck without invoking full-vocabulary Forward KL evaluations. An end-to-end training procedure is provided in Algorithm 1 (Appendix F). Teacher-information requirements and analytical loss-side costs are detailed in Appendix B.

5 Experiments

Models and Training Details. We evaluate IDA-OPD on mathematics and code at both the 4B and 8B scales, using Qwen3-4B-Non-Thinking and Qwen3-8B-Non-Thinking as students. For each student, we use the corresponding same-scale RL-Math teacher for mathematics and RLCode teacher

Method	Math								Code			
	AIME24		AIME25		HMMT Feb		HMMT Nov		MBPP++		LCB	
	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16	pass@1	pass@16
Student	33.3	56.7	30.0	43.3	6.7	26.7	23.3	33.3	53.0	69.5	25.46	34.04
Teacher	60.0	83.3	53.5	73.3	30.0	50.0	46.7	63.3	58.6	70.7	40.6	53.9
+ OPD	61.7	79.1	47.9	70.7	37.5	49.5	38.6	60.8	57.1	69.3	40.6	50.7
+ REOPOLD	64.0	77.4	48.5	72.6	36.5	49.6	38.9	61.3	58.0	70.5	41.6	51.5
+ Entropy Bonus	62.0	78.3	47.2	72.0	36.5	50.8	38.4	61.7	57.1	71.2	41.4	51.9
+ Advantage Shaping	63.2	79.6	47.8	72.4	37.0	52.6	39.0	62.3	57.9	69.3	41.1	51.1
+ AOPD	**65.9**	81.0	48.6	76.0	37.2	54.4	37.6	62.7	58.2	69.5	41.7	51.4
+ EOPD	65.6	80.8	48.0	72.6	37.5	54.1	39.8	62.8	56.9	69.9	40.1	51.6
IDA-OPD	63.3	**83.3**	**50.0**	**76.7**	**37.6**	**56.7**	**40.1**	**63.3**	58.5	71.0	41.6	52.2
Δ vs OPD	$+1.6$	$+4.2$	$+2.1$	$+6.0$	$+0.1$	$+7.2$	$+1.5$	$+2.5$	$+1.4$	$+1.7$	$+1.0$	$+1.5$

Table 2: 8B distillation results on mathematics and code benchmarks, reporting pass@1 and pass@16. The student is Qwen3-8B-Non-Thinking; teachers are Qwen3-8B-Non-Thinking-RL-Math for Math and Qwen3-8B-Non-Thinking-RLCode for Code. Bold marks the best distilled-student result in each column. Δ reports absolute gains over OPD.

for code; all teachers are GRPO-trained following Yang et al. (2026). For mathematics, we follow Li et al. (2026b) and train on DeepMath103K filtered to difficulty level 6 (the hardest problems). For code, we train on the filtered code subset from Yang et al. (2026).

Evaluation. For mathematics, we evaluate on AIME 2024, AIME 2025, HMMT 2025 Feb, and HMMT 2025 Nov. Following Zhu et al. (2026), we adopt the unbiased pass@k estimator introduced by Chen et al. (2021a) (see Appendix D for the formula and variance rationale). Specifically, for each problem x, we generate $n = 128$ responses using independent random seeds, with temperature 1.0 and top-$p = 1.0$. We report pass@1 and pass@16 in Tables 1 and 2, with the complete pass@k curves presented in Section 5.2.

Baselines. We compare against standard OPD (Lu & Thinking Machines Lab, 2025), three recent improvements (EOPD (Jin et al., 2026), AOPD (Jia et al., 2026), and REOPOLD (Ko et al., 2026)), and two entropy-based exploration strategies on top of OPD: an entropy bonus (Schulman et al., 2017) and advantage shaping (Cheng et al., 2025). EOPD, AOPD and REOPOLD use the official implementation.

5.1 Main Results

Tables 1 and 2 report pass@1 and pass@16 for the 4B and 8B settings. On mathematics, student-side methods can improve pass@1 while pass@16 still trails the teacher. For example, 4B OPD reaches 65.7% on AIME 2025 against the teacher's 73.3%, with a similar gap in the 8B setting.

IDA-OPD attains the highest pass@16 across all four mathematics benchmarks at both scales. On HMMT Feb, gains over OPD reach $+8.3$ and $+7.2$ in the 4B and 8B settings, respectively. It also matches or surpasses the teacher on several benchmarks (e.g., 83.3% vs. 80.0% on 4B AIME 2024), while broadly improving pass@1 over OPD.

IDA-OPD also exceeds the entropy bonus (Schulman et al., 2017) and advantage shaping (Cheng et al., 2025) on mathematics pass@16, favoring selective attenuation over global entropy inflation. It matches or exceeds AOPD and EOPD using student logits and only the teacher's sampled-token probability, without their expensive top-K teacher distributions.

The gains extend to code: IDA-OPD improves pass@16 over OPD by $+1.5$ on both MBPP++ and LiveCodeBench v6 while improving pass@1 (Table 1). OPD's MBPP++ pass@16 remains below the student's (73.3 vs. 73.5).

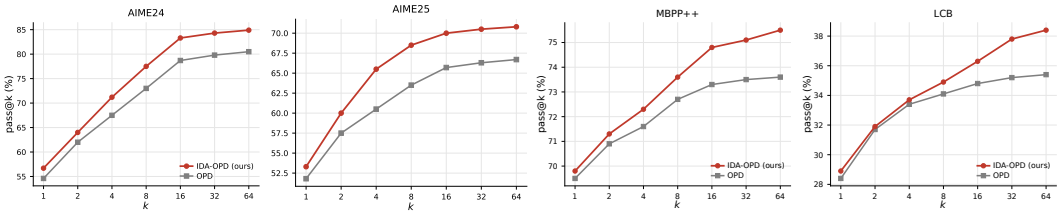


Figure 4: Pass@k in the 4B setting on AIME 2024, AIME 2025, MBPP++, and LiveCodeBench v6 (left to right), for $k \in \{1, 2, 4, 8, 16, 32, 64\}$. Red circles denote IDA-OPD and gray squares denote OPD.

5.2 Pass@k Performance

Following prior pass@k evaluations (Cheng et al., 2025; Jin et al., 2026; Ko et al., 2026), we measure the probability of obtaining at least one correct solution within k sampled responses. Figure 4 reports pass@k on AIME 2024, AIME 2025, MBPP++, and LiveCodeBench v6 for $k \in \{1, 2, 4, 8, 16, 32, 64\}$ in the 4B setting. On the AIME benchmarks, the two methods are close at $k = 1$ ($\sim +2$ points), but the gap widens to $+4$ to $+5$ points by $k = 8$–16, and the advantage is sustained through $k = 64$.

The code benchmarks show a similar benefit at larger sampling budgets. On MBPP++, IDA-OPD improves over OPD by 0.3 percentage points at $k = 1$, 1.5 at $k = 16$, and 1.9 at $k = 64$ (75.5% vs. 73.6%). On LiveCodeBench v6, the corresponding gains are 0.5, 1.5, and 3.0 points (38.4% vs. 35.4% at $k = 64$). Thus, the improvement extends beyond single-sample accuracy to the probability of finding a correct solution with a larger sampling budget, across both mathematics and code.

5.3 Ablation Study

IDA-OPD is composed of two coupled design choices: (i) using the sign of $\mathcal{I}_H(y)$ to select which positions to intervene on, and (ii) replacing the advantage at those positions with the divergence-adaptive shrinkage $w_y A_y$. To isolate the contribution of each component, we design three ablations, all trained under the same setting as IDA-OPD on the Qwen3-4B-Non-Thinking-RL-Math $\rightarrow$ Qwen3-4B block:

- **Uniform shrinkage (w/o $\mathcal{I}_H$ gate).** Apply the shrinkage at every position, $\widetilde{A}_y = w_y A_y$, ignoring the sign of $\mathcal{I}_H(y)$.
- **Hard mask (w/o shrinkage).** Keep the gate but zero out negative-influence updates: $\widetilde{A}_y = A_y$ if $\mathcal{I}_H(y) \geq 0$, else $\widetilde{A}_y = 0$.
- $\mathrm{sign}(A_y)$ **gate.** Replace the gate with the advantage sign ($w_y A_y$ when $A_y < 0$), testing whether $\mathcal{I}_H(y)$ adds information beyond the sign of A_y.

Among the tested shrinkage mappings, the linear map achieves the best pass@1 and pass@16 on AIME24 and AIME25; the full comparison is in Appendix E.

5.4 Effect of Influence-Based Selection on Training Entropy

We now examine how $\mathcal{I}_H(y)$-based selection affects the entropy trajectory during training. Figure 5a tracks the student's policy entropy during training under vanilla OPD and two hard-masking variants: one that drops negative-influence updates ($\mathcal{I}_H(y) < 0$), and another that drops nonnegative-influence updates ($\mathcal{I}_H(y) \geq 0$).

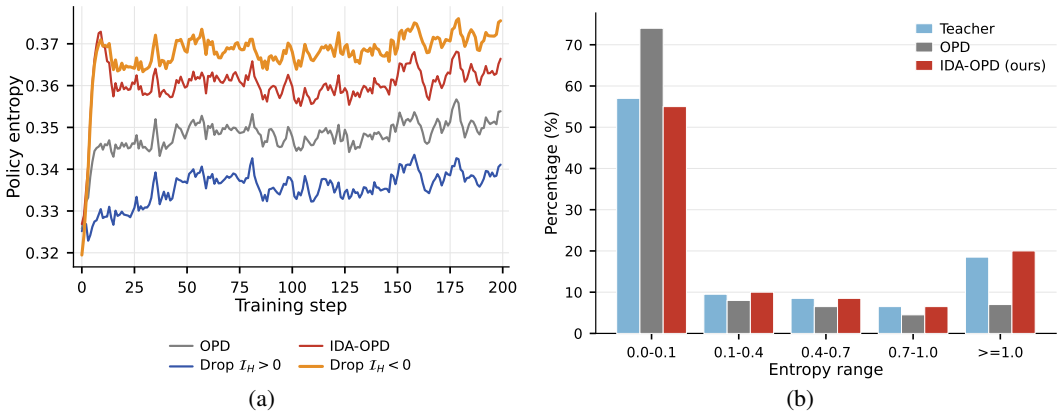
The trajectories bracket the baseline in the predicted directions, providing empirical support for the proxy's utility in guiding update selection under AdamW training. Unlike Theorem 1's idealized logit-space analysis, these aggregate observations do not establish token-level predictive accuracy under shared-parameter optimization.

Masking nonnegative-influence updates lowers entropy below OPD, whereas masking negative-influence updates produces the highest overall entropy. Thus, sign-based selection can steer the

Variant	AIME24		AIME25	
	pass@1	pass@16	pass@1	pass@16
IDA-OPD (full)	**56.7**	**83.3**	**53.3**	**70.0**
w/o $\mathcal{I}_H$ gate	53.8	81.5	51.2	68.4
w/o shrinkage (hard mask)	52.1	82.0	50.4	68.9
sign(A_y) gate	54.2	76.8	51.9	65.2

Table 3: Ablation study on the Qwen3-4B-Non-Thinking-RL-Math → Qwen3-4B setting. Each row disables one design choice of IDA-OPD while keeping the rest of the training pipeline unchanged.

Table 3 shows every ablation degrades relative to full IDA-OPD. Removing the $\mathcal{I}_H(y)$ gate and shrinking indiscriminately also attenuates nonnegative-influence updates, dropping pass@1 (56.7 → 53.8 on AIME24) with only modest pass@16 loss, so entropy-directed selection is necessary. Hard masking retains relatively strong pass@16 (pass@16 82.0/68.9) but inflicts the largest pass@1 damage (56.7 → 52.1), suggesting that removing all negative-influence updates can weaken useful learning signals. Using $\mathrm{sign}(A_y)$ as the gate omits the student-side factor in the local entropy effect (Section 3.1); empirically, it gives the lowest pass@16 (pass@16 76.8/65.2) while trailing on pass@1. The gate and the shrinkage are thus complementary and both required.



(a) (b)

Figure 5: Entropy analysis. **(a)** Policy entropy under OPD, IDA-OPD, and masking nonnegative- or negative-influence updates. IDA-OPD remains above OPD and between the two masking variants. **(b)** Token-level entropy distributions on AIME 2024 for the teacher, OPD, and IDA-OPD.

aggregate entropy trajectory in this training setting. The latter variant also has the largest pass@1 drop in Table 3, showing that higher entropy alone does not ensure better accuracy and motivating attenuation rather than complete removal of negative-influence updates.

5.5 Token-Level Entropy Analysis

To examine how IDA-OPD reshapes per-token uncertainty, we run a token-level analysis on AIME 2024 dataset. Each model rolls out its own response and, following Section 3.1, we compute the entropy of its predicted distribution at every token; Figure 5b aggregates these into a histogram comparing the teacher against OPD- and IDA-OPD-trained students.

In the mid-entropy range ($\sim$ 0.1–1.0) OPD and IDA-OPD both track the teacher; the differences concentrate at the ends. In the lowest-entropy bin (0.0–0.1) OPD piles up far more mass than the teacher ($\sim$ 74% vs. $\sim$ 57%), the signature of over-confident, near-deterministic predictions from entropy contraction, whereas IDA-OPD stays close ($\sim$55%) and correspondingly retains more mass in the high-entropy region ($\geq$1.0) where OPD is depleted. This aggregate pattern is consistent with reduced entropy contraction under IDA-OPD; the resulting histogram is closer to the teacher's.

6 Related Work

On-Policy Distillation for Language Models. Offline knowledge distillation supervises the student on fixed teacher outputs (Hinton et al., 2015; Kim & Rush, 2016), but never corrects it on the

prefixes it visits at inference. On-policy distillation (OPD) removes this train–inference mismatch by training on the student's *own* sampled prefixes (Agarwal et al., 2024; Yu et al., 2026). OPD variants differ in per-token teacher cost: *full-distribution* OPD matches the teacher's entire next-token distribution but needs full-vocabulary logits (Gu et al., 2024; Ko et al., 2024; Xu et al., 2024), whereas *sampled-token* OPD queries only the sampled token's probability, the single-sample estimator of the reverse KL (Lu & Thinking Machines Lab, 2025; Fu et al., 2026; Jia et al., 2026), reducing teacher-probability transfer and storage requirements for long rollouts. This sampled-token formulation has been extended to cross-tokenizer, agent, and unified-framework settings (Sun et al., 2026a; Li et al., 2026a; Sun et al., 2026b; Zhong et al., 2026). Yet by relying on a single sampled-token signal, sampled-token OPD can quietly contract the student's entropy, motivating a local analysis of the sampled-token objective.

Token Selection and Reweighting in OPD. The closest work reconsiders *which* updates OPD should trust. Disagreement- and importance-based methods select or reweight tokens by teacher–student divergence: TIP by token importance (Xu et al., 2026), Token Teachability by whether a disagreement is learnable (Wang et al., 2026), and Filter-Then-Reweight by optimization granularity (Li et al., 2026b). AOPD decouples exploration from imitation (Jia et al., 2026), while RE-OPOLD and negative-sample reinforcement reshape sampled-token credits (Ko et al., 2026; Zhu et al., 2026). Most related, Entropy-Aware OPD (EOPD) restores lost entropy with a forward-KL term at high-uncertainty tokens (Jin et al., 2026). All of these locate salient positions by teacher–student disagreement or uncertainty, and the strongest, EOPD (Jin et al., 2026) and AOPD (Jia et al., 2026), reintroduce full or top-K teacher distributions to act on them. IDA-OPD instead uses negative local entropy influence as an update-selection criterion and attenuates them most strongly near agreement, where cumulative entropy loss concentrates, using only sampled-token teacher probabilities. More broadly, RL studies show that policy entropy and a few high-entropy "forking" tokens are linked to exploration and reasoning performance (Cui et al., 2025; Wang et al., 2025); we analyze the local entropy effect of OPD and use it to guide update selection during training.

7 Conclusion

To address the pass@k bottleneck, IDA-OPD uses First-Order Local Entropy Influence ($\mathcal{I}_H(y)$) as a local proxy, retaining nonnegative-influence updates and shrinking negative-influence advantages by w_y. Sign-based selection can steer aggregate training entropy in our experiments. It improves pass@k over sampled-token baselines and matches the strongest teacher-informed methods, with lower teacher-probability transfer and storage requirements while broadly maintaining OPD's pass@1.

AI use statement

OpenAI Codex was used to migrate the LaTeX manuscript to the ICLR 2027 template, assist with document formatting, citation formatting, compilation, and layout checks, assist with Introduction editing and restoration, refine author-approved passages and captions for length, prepare plots and prepare the method overview diagram and its accompanying description. No other generative AI assistance was used in the research or writing of this paper.

References

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021a.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021b.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. doi: 10.48550/arXiv.2505.22617. URL https://arxiv.org/abs/2505.22617.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, et al. Cde: Curiosity-driven exploration for efficient reinforcement learning in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 59123–59144, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

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

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025. doi: 10.48550/arXiv.2509.25133. URL https://arxiv.org/abs/2509.25133.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo Angel, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

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

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026a.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026b.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab blog, October 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. SimCT: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026a.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Gengsheng Li, Zhepei Hong, Chang Wu, Pengfei Liu, Junfeng Fang, and Xiang Wang. EasyOPD: An easy-to-use on-policy distillation framework for large language models. *arXiv preprint arXiv:2607.11012*, 2026b.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.01939.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher–student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xin Yu, Liuchen Liao, Yiwen Zhang, Yingchen Yu, Lingzhou Xue, and Qinzhen Guo. Preference-based self-distillation: Beyond kl matching via reward regularization. *arXiv preprint arXiv:2605.05040*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Xiaoyun Zhang, Xiaojian Yuan, Di Huang, Wang You, Chen Hu, Jingqing Ruan, Ai Jian, Kejiang Chen, and Xing Hu. Revisiting entropy regularization: Adaptive coefficient unlocks its potential for LLM reinforcement learning. *arXiv preprint arXiv:2510.10959*, 2025. doi: 10.48550/arXiv.2510.10959. URL https://arxiv.org/abs/2510.10959.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. SOD: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*, 2026.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *Advances in Neural Information Processing Systems*, 38:126546–126573, 2026.

A Mathematical Proofs

A.1 Proof of Theorem 1

Assumptions and scope. Theorem 1 characterizes the *local, first-order* entropy effect of a single sampled-token update under the following idealization: (i) a plain gradient-descent step in logit space with fixed step size η, so the update takes the form $\Delta z_k = \eta A_y(\mathbb{I}[k = y] - p_k)$ (no adaptive optimizer, momentum, or weight decay); (ii) the step perturbs the current prefix's logits in isolation, ignoring parameter sharing across prefixes; (iii) a single token at a single position for a single step, rather than a minibatch or multi-step trajectory; and (iv) terms of $\mathcal{O}(\eta^2)$ are neglected, so the identity is exact only as $\eta \to 0$. Under real Adam-based, minibatch, multi-step training these do not hold exactly; we therefore treat $\mathcal{I}_H(y)$ as a leading-order *proxy* for the local entropy direction and use its sign as a selection criterion; Section 5.4 evaluates its effect on aggregate training entropy.

Proof. To compute the local entropy change, we first expand the probability change $\Delta p_i = p_i' - p_i$ for each token under the local logit update $\Delta z_k = \eta A_y(\mathbb{I}[k = y] - p_k)$.

The softmax Jacobian is given by:

$$\frac{\partial p_i}{\partial z_k} = p_i(\mathbb{I}[i = k] - p_k). \tag{5}$$

By the chain rule,

$$\begin{aligned}
\Delta p_i &= \sum_{k\in\mathcal{V}} \frac{\partial p_i}{\partial z_k}\Delta z_k + \mathcal{O}(\eta^2) \\
&= \eta A_y p_i \sum_{k\in\mathcal{V}} (\mathbb{I}[i = k] - p_k)(\mathbb{I}[k = y] - p_k) + \mathcal{O}(\eta^2) \\
&= \eta A_y p_i \left(\mathbb{I}[i = y] - p_i - p_y + \sum_{k\in\mathcal{V}} p_k^2\right) + \mathcal{O}(\eta^2)
\end{aligned} \tag{6}$$

Let $S_2 = \sum_{k\in\mathcal{V}} p_k^2$. A first-order expansion of the entropy gives

$$\Delta H = -\sum_{i\in\mathcal{V}} (\log p_i + 1)\Delta p_i + \mathcal{O}(\eta^2). \tag{7}$$

Since the probabilities sum to one, $\sum_i \Delta p_i = 0$, and the constant term vanishes:

$$\Delta H = -\sum_{i\in\mathcal{V}} \log p_i \Delta p_i + \mathcal{O}(\eta^2). \tag{8}$$

Substituting our derived probability change (Eq. 6) into this entropy derivative gives:

$$\begin{aligned}
\Delta H = -\eta A_y \Bigg[& p_y \log p_y - \sum_i p_i^2 \log p_i \\
& - p_y \sum_i p_i \log p_i \\
& + S_2 \sum_i p_i \log p_i \Bigg] + \mathcal{O}(\eta^2).
\end{aligned} \tag{9}$$

Using the definition of entropy $H(p) = -\sum_i p_i \log p_i$, we can simplify the expression to:

$$\begin{aligned}
\Delta H = \eta A_y \Bigg[& \sum_i p_i^2 \big(\log p_i + H(p)\big) \\
& - p_y \big(\log p_y + H(p)\big) \Bigg] + \mathcal{O}(\eta^2).
\end{aligned} \tag{10}$$

Finally, identifying the bracketed term as the entropy-direction factor D_y, the entropy change can be written as:

$$\Delta H = \eta A_y D_y + \mathcal{O}(\eta^2). \tag{11}$$

By substituting the definition of the *First-Order Local Entropy Influence*, $\mathcal{I}_H(y) = A_y D_y$, we obtain the final form:

$$\Delta H = \eta \cdot \mathcal{I}_H(y) + \mathcal{O}(\eta^2), \tag{12}$$

which concludes the proof. □

A.2 Proof of Proposition 1

We first derive a closed-form expression for the symmetric relative disagreement $w_y = |q_y - p_y|/(q_y + p_y)$ in terms of the advantage A_y, and then read off the two asymptotic behaviors claimed in Proposition 1.

Closed-form identity. By definition, $A_y = \log q_y - \log p_y$, hence $q_y/p_y = e^{A_y}$. Substituting into w_y and dividing numerator and denominator by p_y gives

$$w_y = \frac{|q_y - p_y|}{q_y + p_y} = \frac{|e^{A_y} - 1|}{e^{A_y} + 1} = \frac{e^{|A_y|} - 1}{e^{|A_y|} + 1} = \tanh\left(\frac{|A_y|}{2}\right).$$

Thus w_y depends on p_y and q_y only through their advantage A_y, and takes values in $[0, 1)$.

Asymptotic order. Using $\tanh(x) = x - x^3/3 + \mathcal{O}(x^5)$ as $x \to 0$,

$$w_y = \frac{|A_y|}{2} + \mathcal{O}\big(|A_y|^3\big), \qquad A_y \to 0.$$

Multiplying by A_y (which contributes the sign),

$$w_y\, A_y = \frac{A_y\, |A_y|}{2} + \mathcal{O}\big(A_y^3\big) = \mathcal{O}\big(A_y^2\big), \qquad A_y \to 0,$$

which establishes the first claim.

For the second claim, as $|A_y| \to \infty$ we have $\tanh(|A_y|/2) \to 1$, so

$$w_y\, A_y = \tanh\left(\frac{|A_y|}{2}\right) A_y \longrightarrow A_y,$$

i.e., the shrinkage becomes asymptotically lossless for large teacher–student disagreements. Combining the two limits yields Proposition 1.

B Computational Efficiency

Like OPD, IDA-OPD requires only one teacher probability per sampled position and no additional teacher distribution. This reduces teacher-probability transfer and storage requirements relative to supplying richer teacher distributions at the same positions. These requirements are distinct from end-to-end computational efficiency, which requires separate measurements of runtime, throughput, and peak memory; we do not report such measurements here.

Per-rollout resource requirements. Table 4 separates teacher-probability transfer from additional loss-side computation for rollout length L and vocabulary size V. Sampled-token methods transfer $O(L)$ teacher scalars. The selected-position top-K component of EOPD and AOPD scales as $O(L_{\text{sel}}K)$; its magnitude depends on the selected-position count L_{sel} and truncation size K. All methods still require teacher forward evaluation. IDA-OPD computes D_y using the available student logits, adding $O(LV)$ student-side reductions without an additional teacher query. The transfer/storage comparison therefore does not by itself establish lower overall computational cost.

Method	Teacher info needed	Teacher→student transfer / rollout	Extra loss-side compute	Teacher dist.?
OPD	sampled-token log-prob	$O(L)$	—	no
REOPOLD	sampled-token (reweighted)	$O(L)$	$O(L)$	no
EOPD	top-K dist. at sel. pos.	$O(L_{\text{sel}}K)$	$O(L_{\text{sel}}K)$	yes
AOPD	top-K dist. at sel. pos.	$O(L_{\text{sel}}K)$	$O(L_{\text{sel}}K)$	yes
IDA-OPD (ours)	sampled-token log-prob	$O(L)$	$O(LV)$ (student only)	no

Table 4: Analytical per-rollout resource requirements (not measured runtime) by rollout length L and vocabulary size V; L_{sel} is the number of positions at which teacher information is injected, and K is the teacher top-K truncation size ($1 \ll K \ll V$). All methods share the same $O(LV)$ teacher forward pass, which is omitted. The last column marks whether a method needs a multi-token teacher distribution beyond the single sampled-token scalar.

C Policy Entropy and Pass@k

Policy entropy is closely related to exploration in RL for reasoning: it tends to collapse over training (Cui et al., 2025), a small set of high-entropy "forking" tokens carries most of the useful gradient (Wang et al., 2025), and entropy regularization helps only when its coefficient is adapted rather than fixed (Zhang et al., 2025; Jiang et al., 2025; Schulman et al., 2017). System-level recipes such as DAPO counter this collapse with a decoupled "clip-higher" rule and dynamic sampling, but they act through coarse update-level clipping rather than using the local entropy-influence criterion considered here (Yu et al., 2025). We evaluate downstream reasoning performance with pass@k, the probability that at least one of k sampled responses is correct (Chen et al., 2021b; Cheng et al., 2025). This differs from selecting a final answer from those responses, as in self-consistency (Wang et al., 2023). Yue et al. (2025) show that reinforcement learning can lift pass@1 yet fall below the *base* model's pass@k at larger k, while distillation can improve performance across sampling budgets. This mismatch motivates our focus on improving pass@k in sampled-token OPD. We use entropy to investigate this bottleneck, deriving a local first-order criterion for update selection and evaluating its aggregate effect during training. The resulting pass@k gains are interpreted directly as improved success under repeated sampling.

D Implementation Details

Evaluation estimator. Following Zhu et al. (2026), we use the unbiased pass@k estimator of Chen et al. (2021a). For each problem x, we generate $n = 128$ responses using independent random seeds, with temperature 1.0 and top-$p = 1.0$. Let c_x denote the number of correct responses among the n samples. We estimate pass@k as

$$\text{pass@}k = \mathbb{E}_{x\sim\mathcal{D}}\left[1 - \frac{\binom{n-c_x}{k}}{\binom{n}{k}}\right], \qquad n \geq k. \tag{13}$$

Compared with directly evaluating pass@k using only k samples per problem, generating $n > k$ samples and applying the unbiased estimator mitigates the variance caused by stochastic decoding.

On-policy training. OPD, EOPD, and IDA-OPD are all implemented on top of the verl (Sheng et al., 2024) framework, using vLLM for student rollout and FSDP for training. As shown in Table 5, we use a training batch size of $B = 96$ and a mini-batch size of $B_{\text{mini}} = 24$, which yields four gradient update steps per training iteration. All three methods are on-policy: we generate a single rollout per problem during training and supervise the student on its own sampled prefixes. The three methods share an identical training configuration and differ only in the teacher signal each consumes: OPD and IDA-OPD use the sampled-token teacher log-probability, whereas EOPD queries a truncated top-K teacher distribution ($K = 64$) at selected positions.

Entropy Bonus. Entropy Bonus (Schulman et al., 2017) adds the policy entropy directly to the optimization objective, discouraging premature policy convergence and encouraging stochastic action selection during training.

Hyperparameter	OPD	EOPD	IDA-OPD
Learning rate	1e-6	1e-6	1e-6
LR scheduler type	cosine	cosine	cosine
Optimizer	AdamW	AdamW	AdamW
Training batch size	96	96	96
Mini batch size	24	24	24
Samples per prompt	1	1	1
Max response length	16384	16384	16384
Top-K	–	64	–
Training temperature	1.0	1.0	1.0

Table 5: Hyperparameters used for OPD, EOPD, and IDA-OPD. Top-K applies only to the teacher-distribution method (EOPD); the sampled-token methods (OPD, IDA-OPD) transfer a single scalar per position and do not use it.

Advantage Shaping. Advantage Shaping (Cheng et al., 2025) augments the advantage function by adding a gradient-detached entropy term shaped by $\psi(\cdot)$:

$$A_t \leftarrow A_t + \psi(H_t). \tag{14}$$

This mechanism encourages the model to reinforce uncertain decisions when $A_t > 0$ by providing additional reward, while reducing the penalty for uncertain decisions when $A_t < 0$, thereby promoting more exploratory reasoning behavior. Here, the shaping function $\psi(\cdot)$ is defined as

$$\psi(H_t) = \min\left(\alpha H_t^{\text{detach}},\ \frac{|A_t|}{\kappa}\right), \tag{15}$$

where $\alpha > 0$ and $\kappa > 1$ are hyperparameters. In the experiment described in Section 5.5, we follow the setup of Cheng et al. (2025) and set $\alpha = 0.1$ and $\kappa = 2$.

E Effect of the Divergence-Adaptive Shrinkage

We now examine the shape of the shrinkage, i.e., how negative-influence updates are attenuated as a function of disagreement. The shrinkage sets the multiplier c_y from the symmetric relative disagreement $w_y = \frac{|q_y - p_y|}{q_y + p_y} \in [0, 1)$. Holding the $\mathcal{I}_H(y)$ gate fixed, we compare four shapes: a non-adaptive **Constant**, concave **Sqrt** ($c_y = w_y^{1/2}$), our linear **Ours** ($c_y = w_y$), and convex **Square** ($c_y = w_y^2$).

Shrinkage $c_y = f(w_y)$	**AIME24**		**AIME25**	
	pass@1	pass@16	pass@1	pass@16
Constant ($c_y = \alpha$)	54.6	79.2	51.5	66.7
Sqrt ($c_y = w_y^{1/2}$)	55.9	80.4	52.4	67.5
Ours ($c_y = w_y$)	**56.7**	**83.3**	**53.3**	**70.0**
Square ($c_y = w_y^2$)	54.1	81.6	51.0	68.8

Table 6: Effect of the disagreement-to-shrinkage mapping $f(w_y)$ on the Qwen3-4B-Non-Thinking-RL-Math → Qwen3-4B setting. All variants share the same $\mathcal{I}_H(y)$ gate.

Table 6 shows our linear map tops every column (56.7/83.3, 53.3/70.0). The Constant map is worst, leaving the low-divergence entropy drain uncurbed while weakening high-divergence corrections; concave Sqrt under-attenuates low-divergence updates (pass@16 80.4/67.5); convex Square over-suppresses moderate-divergence corrections that still carry teacher signal, posting the lowest pass@1. Scaling attenuation linearly with w_y thus best balances suppressing low-discrepancy drain against preserving high-discrepancy corrections.

F IDA-OPD TRAINING ALGORITHM

Algorithm 1 summarizes the training procedure in Section 4. Each iteration generates one response per prompt from the current student and then updates the student over rollout minibatches. The teacher is frozen and returns only the sampled-token log-probability at each visited prefix. The student quantities are recomputed at each minibatch update; no gradient is propagated through the transformed advantage.

Algorithm 1: Influence-Directed Adaptive On-Policy Distillation (IDA-OPD)

Input: Prompt dataset $\mathcal{D}$; student p_θ; frozen teacher p_{T}; training iterations N; batch size B; minibatch size B_{mini}; AdamW optimizer and learning-rate schedule.
Output: Trained student parameters θ.

for $s = 1, \ldots, N$ **do**
  Sample B prompts from $\mathcal{D}$
  Generate one response per prompt using the current student p_θ
  For each response token y at prefix h, cache $\ell^{\mathrm{T}}_{h,y} \leftarrow \log p_{\mathrm{T}}(y \mid h)$
  foreach *rollout minibatch* $\mathcal{B}$ *of size* B_{mini} **do**
    Initialize the collection of token losses $\mathcal{E} \leftarrow \varnothing$
    foreach *valid response-token position* (h, y) *in* $\mathcal{B}$ **do**
      Compute student log-probabilities $\log p_i \leftarrow \log p_\theta(i \mid h)$ for $i \in \mathcal{V}$
      $p_i \leftarrow \exp(\log p_i)$ for $i \in \mathcal{V}$
      $A_y \leftarrow \ell^{\mathrm{T}}_{h,y} - \log p_y$
      $H \leftarrow -\sum_{i \in \mathcal{V}} p_i \log p_i$
      $D_y \leftarrow \sum_{i \in \mathcal{V}} p_i^2 (\log p_i + H) - p_y(\log p_y + H)$
      $\mathcal{I}_H(y) \leftarrow A_y D_y$
      if $\mathcal{I}_H(y) < 0$ **then**
        $w_y \leftarrow \tanh(|A_y|/2)$
        $\widetilde{A}_y \leftarrow w_y A_y$
      else
        $\widetilde{A}_y \leftarrow A_y$
      $\ell_{h,y} \leftarrow -\,\mathrm{sg}(\widetilde{A}_y) \log p_y$
      Append $\ell_{h,y}$ to $\mathcal{E}$
    $\mathcal{L} \leftarrow \mathrm{Aggregate}_{\mathrm{OPD}}(\mathcal{E})$
    Update θ with AdamW using $\nabla_\theta \mathcal{L}$ and the learning-rate schedule
return θ

Here, valid positions exclude prompt and padding tokens, and $\mathrm{Aggregate}_{\mathrm{OPD}}$ denotes the same masked loss reduction used by the OPD baseline. $\mathrm{sg}(\cdot)$ detaches the entire transformed advantage, so gradients flow only through the sampled-token student log-probability in each loss term. The retention factor $\tanh(|A_y|/2) = |q_y - p_y|/(q_y + p_y)$ follows from Appendix A.2; it uses the same sampled-token signal without requiring additional teacher probabilities. Nonnegative-influence updates are retained unchanged, while negative-influence updates are attenuated. All vocabulary-wide quantities in the algorithm come from the student. The batch sizes, optimizer, and other training settings are listed in Table 5.