When Are Teacher Tokens Reliable? Position-Weighted On-Policy Self-Distillation for Reasoning

Xiaogeng Liu[1]* **Xinyan Wang**[2] **Yingzi Ma**[2] **Yechao Zhang**[3] **Chaowei Xiao**[1]

[1]Johns Hopkins University [2]University of Wisconsin–Madison
[3]Nanyang Technological University

Abstract

On-policy self-distillation (OPSD) trains a student on its own rollouts using a privileged teacher, but its standard objective weights all generated tokens equally, implicitly treating the privileged teacher target as equally reliable at every student-visited prefix. Existing entropy-based on-policy distillation methods relax this uniformity by modulating token-level supervision with teacher entropy, but high teacher entropy in reasoning has an ambiguous reliability meaning: it can reflect either non-viable uncertainty or benign solution diversity. To separate the two, we introduce a branch-viability diagnostic. Specifically, we record next-token alternatives from the privileged-answer teacher prompt, force each alternative after the student prompt plus its on-policy spine prefix, and test whether the resulting student-template continuation recovers the correct answer. On Qwen3-4B, we find that an oriented within-sequence position score is the strongest tested predictor of teacher-token reliability, reaching an area-under-ROC-curve (AUROC) of 0.83 with a 95% cluster-bootstrap interval of $[0.66, 0.95]$; local uncertainty scores are at most 0.57. Motivated by this trajectory-level structure, we propose Position-Weighted On-Policy Self-Distillation (PW-OPSD), which applies an increasing position weight while keeping the same student rollout, privileged teacher pass, and clipped forward-KL target as OPSD. Across three evaluation seeds, the diagnostic-derived PW-OPSD improves AIME 2024 and AIME 2025 Avg@12 over OPSD by $+1.0$ and $+1.1$ points, and a generalization evaluation on two larger-scale models from different families, DeepSeek-R1-Distill-Llama-8B and Olmo-3-7B-Think, also demonstrates consistent aggregate Avg@12 improvements. These results show that teacher-token reliability in reasoning distillation is trajectory-structured and can be utilized without additional teacher computation. The code is available at `https://github.com/SaFo-Lab/PW-OPSD`.

1 Introduction

On-policy self-distillation (OPSD) [Zhao et al., 2026] is a practical recipe for distilling mathematical reasoning because it supervises the student on prefixes the student actually visits. The student samples a rollout from the ordinary problem prompt, and a privileged copy of the model, conditioned on reference information, provides token-level targets along that same rollout. This construction avoids the trajectory mismatch of teacher-generated demonstrations while still injecting privileged information at training time.

The standard OPSD objective nevertheless makes an implicit reliability assumption: every generated token receives the same forward-KL supervision [Zhao et al., 2026]. This uniform objective treats the

*Corresponding to `xliu316@jhu.edu, chaoweixiao@jhu.edu`.

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

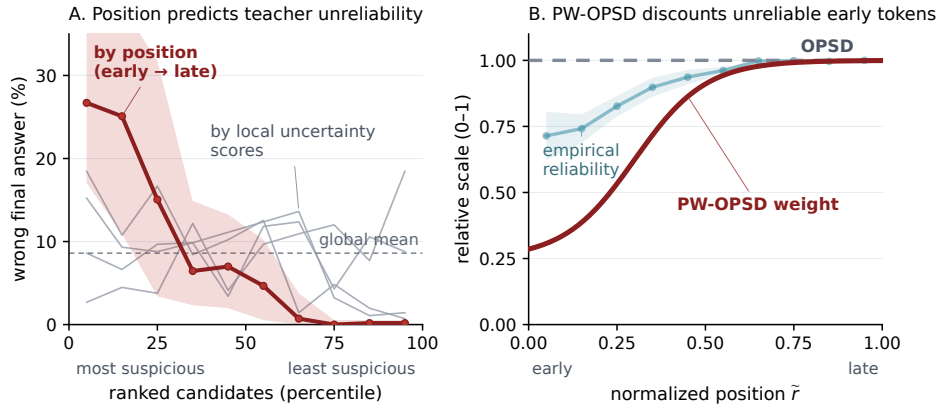


Figure 1: Branch viability reveals a positional reliability structure in Qwen3-4B reasoning traces. Alternatives are selected by the same model under the privileged-answer teacher prompt and rolled out under the ordinary student prompt; for high-ambiguity candidate positions, these forced student-context continuations fail more often at early positions and rarely fail late in the trace. **(A)** Sorting candidates by normalized position therefore separates teacher-unreliable branch points from viable diversity, whereas local uncertainty (entropy) scores yield near-flat failure curves. **(B)** The resulting empirical reliability curve motivates the PW-OPSD weight schedule, which discounts early high-ambiguity positions while preserving full-strength supervision later. See Section 3.2 for the details of the diagnostic experiments.

privileged teacher target as equally useful at every student-visited prefix. Recent adaptive, relaxed, or gated distillation objectives challenge this uniformity assumption: EOPD augments reverse-KL on-policy distillation (OPD) with forward KL on high-teacher-entropy tokens [Jin et al., 2026], ToDi mixes forward and reverse KL per token using a teacher-student probability log-ratio [Jung et al., 2025], REOPOLD treats the teacher-student log-likelihood ratio as a token reward with reward clipping and entropy-based sampling [Ko et al., 2026], and GATES gates privileged self-distillation by tutor consensus [Stein et al., 2026].

All of these criteria measure how ambiguous the teacher's local distribution is, but token-local ambiguity is *not* the same as low teacher reliability. A high-entropy teacher distribution can reflect non-viable uncertainty: several locally plausible next-token alternatives receive probability mass, but some of them fail to recover the correct answer under continued student-context generation. However, it can also reflect benign diversity: the teacher assigns mass to multiple viable solution routes or surface realizations that still preserve correctness. Thus a token-local uncertainty score can identify ambiguity, but it does not distinguish non-viable uncertainty from viable solution diversity.

The practical cost is that any reweighting following local ambiguity alone will either amplify supervision at branch points where the teacher derails, or downweight supervision at benign branch points where the teacher stays correct. Thus, the right object to estimate for adaptive token weighting is not teacher entropy but teacher *reliability*: a per-context probability that the teacher's local target is useful supervision for the student. We formalize this *reliability* as a latent indicator I_t for whether matching the teacher target at distillation context c_t helps preserve or recover the correct solution. The reliability-weighted surrogate then weights the forward-KL term by $\rho^*(c_t) = \Pr(I_t = 1 \mid c_t)$.

To connect this latent quantity to observable behavior, we introduce a branch-viability diagnostic. Throughout, "teacher" and "student" refer to two prompt templates applied to the same model: the teacher template includes the privileged ground-truth answer; the student template is the ordinary problem prompt. Starting from a student-template rollout (the *spine*) that reaches the correct answer, we use the teacher template to propose high-ambiguity next-token alternatives, force each alternative after the student-template prompt plus the spine prefix, and continue under the student template. We use student-template continuation because a teacher-template continuation can re-use the privileged answer and recover from almost any forced branch, collapsing the labels. If the forced alternatives usually fail to recover the correct answer, the candidate is labeled *real-uncertain*: at that prefix, the teacher's local alternatives are low-reliability supervision for the student distribution. If the alternatives remain viable, the ambiguity is treated as benign diversity rather than as evidence of low teacher reliability.

On Qwen3-4B [Yang et al., 2025], this diagnostic shows a clear positional pattern, as demonstrated in Fig. 1. In the correct-spine subset, real-uncertain candidates concentrate early in the normalized trace, while diversity candidates are more common later. After within-problem residualization, an oriented position score separates real-uncertain candidates from diversity candidates with AUROC 0.83 and a 95% cluster-bootstrap interval of $[0.66, 0.95]$; the tested local uncertainty diagnostics reach residual AUROC at most 0.57. Thus the diagnostic supplies the direction of the reliability prior: early high-ambiguity branch points are where the privileged teacher's local alternatives most often become unreliable, while later positions are more reliable under forced continuation. The pattern also holds without the entropy pre-filter and on a chemistry benchmark, and forcing lower-ranked teacher tokens identifies it as a property of the student trajectory.

Inspired by this finding, we propose Position-Weighted On-Policy Self-Distillation (PW-OPSD), which implements this reliability trend as a plug-in approximation to $\rho^*(c_t)$. It keeps the same student rollout, privileged teacher pass, and per-vocabulary clipped forward-KL surrogate as OPSD, but aggregates token losses with an increasing sigmoid function of normalized within-sequence position. The floor keeps early tokens partially supervised, while the increasing schedule assigns stronger weight to later positions where the diagnostic indicates higher teacher reliability. The method therefore changes only the outer reliability-weighted aggregation, adding no extra teacher pass or auxiliary verifier.

We evaluate PW-OPSD on Qwen3-4B math reasoning benchmarks with the same maximum generation length of $38{,}912$ tokens as the OPSD reference evaluation [Zhao et al., 2026]. The diagnostic-derived schedule matches OPSD on MATH-500 Avg@12 (95.34 vs. 95.33) and improves Avg@12 by $+1.0$ point on AIME 2024 and $+1.1$ points on AIME 2025, with the same teacher cost. The position-schedule sweep further shows how the strength of the early-token discount affects the hardest benchmark: the Aggressive schedule improves HMMT 2025 Avg@12 by $+1.48$ points over OPSD and Maj@12 by $+1.11$ points. Without retuning, the diagnostic-derived schedule improves the four-benchmark average on DeepSeek-R1-Distill-Llama-8B and Olmo-3-7B-Think. These results support position as a structural reliability prior and show that reasoning distillation benefits from modeling teacher reliability along the trajectory.

2 Related work

On-policy distillation. Knowledge distillation trains a student to match soft teacher probabilities rather than only hard labels [Hinton et al., 2015]. Sequence-level distillation extends this idea from token labels to generated completions [Kim and Rush, 2016]. Modern LLM post-training popularized instruction- and feedback-based tuning [Ouyang et al., 2022]; in distillation, recent LLM methods often move supervision on policy, so the student is trained on prefixes it actually visits [Agarwal et al., 2024, Gu et al., 2024, Ko et al., 2024, Zhao et al., 2026, Sang et al., 2026, Xu et al., 2026, Hou et al., 2026]. This follows the interactive imitation-learning intuition of querying expert feedback on learner-visited states [Ross et al., 2011]; empirical work on LLM imitation also reports degradation under off-policy imitation in multi-step generation settings [Gudibande et al., 2023]. GKD formalizes generalized on-policy knowledge distillation [Agarwal et al., 2024], MiniLLM optimizes sequence-level reverse-KL distillation on student/teacher-mixed on-policy samples through a policy-gradient formulation [Gu et al., 2024], and DistiLLM studies skewed KL objectives for white-box OPD on student-generated outputs [Ko et al., 2024]. See Song and Zheng [2026] for a broader survey of OPD. OPSD is the main baseline in this paper: it uses a privileged teacher prompt containing reference information to provide token-level targets along student rollouts [Zhao et al., 2026]. PW-OPSD keeps the privileged on-policy construction of OPSD and makes the outer reliability weight trajectory-structured rather than uniform.

Divergence choice and adaptive token weighting. Several distillation methods change the divergence or token-level controller between teacher and student. MiniLLM replaces standard forward-KL KD with sequence-level reverse KL and optimizes the resulting on-policy objective via policy-gradient returns [Gu et al., 2024]; DistiLLM uses skewed KL objectives to stabilize white-box OPD on student-generated outputs [Ko et al., 2024]. Broader f-divergence work studies how divergence choice changes sequence-level KD behavior [Wen et al., 2023]. AKL revisits the usual mode-seeking/mode-covering framing for LLM distillation, arguing that forward KL (FKL) and reverse KL (RKL) share the same asymptotic objective but that FKL emphasizes head probabilities while RKL

emphasizes tail probabilities in early epochs, then proposing an adaptive KL mixture [Wu et al., 2025]. Recent adaptive or privileged distillation methods also depart from uniform token-level supervision, but for different reasons and with different control signals. EOPD applies forward KL on high-teacher-entropy tokens while retaining reverse KL elsewhere in OPD [Jin et al., 2026]; REOPOLD [Ko et al., 2026] interprets OPD as policy optimization with a teacher-student log-likelihood-ratio token reward and relaxes strict imitation through reward clipping, entropy-based dynamic sampling, and exploration-to-refinement training. Outside this student-rollout OPD divergence-switching line, ToDi adaptively combines forward and reverse KL per token using a teacher-student probability log-ratio [Jung et al., 2025]; GATES instead gates privileged self-distillation using consensus among tutor traces [Stein et al., 2026]. These methods show that distillation supervision can be modulated per token through entropy, teacher-student ratios, clipped rewards, or consensus signals. However, these local control signals do not directly distinguish non-viable uncertainty from benign solution diversity. To distinguish the two, we propose a branch-viability diagnostic in which the same model, under the privileged-answer teacher prompt, selects forced alternatives and then rolls them out under the ordinary student prompt to test whether the correct answer is still recovered. We compare branch viability against standard uncertainty diagnostics, including predictive entropy, MC-dropout mutual information, and Dirichlet or evidential uncertainty [Kendall and Gal, 2017, Gal and Ghahramani, 2016, Malinin and Gales, 2018, Sensoy et al., 2018]. These baselines quantify local ambiguity, but they do not test whether high-probability teacher alternatives preserve correctness under continued generation.

3 Method

PW-OPSD is a reliability-weighted version of OPSD that preserves the student rollout, privileged teacher pass, and per-vocabulary clipped forward-KL surrogate. It changes the outer token aggregation through a position-dependent reliability weight and a per-sequence mean, motivated by a branch-viability diagnostic showing that early high-ambiguity branch points are often teacher-unreliable, while local uncertainty scores are weak predictors of this reliability event.

3.1 Background: On-policy self-distillation

Let x_{stu} be the ordinary student prompt and x_{tch} be the privileged teacher prompt that includes the reference solution. On-policy self-distillation (OPSD) [Zhao et al., 2026] first samples a completion from the current student,

$$y_{1:T} \sim \pi_\theta(\cdot \mid x_{\mathrm{stu}}),$$

and then queries a teacher copy of the same model on the privileged context and the student prefix. At each valid generated position t, the teacher target is

$$\mathbf{p}_t^{\mathrm{opsd}} = p_{\mathrm{tch}}(\cdot \mid x_{\mathrm{tch}}, y_{<t}), \qquad p_t = \pi_\theta(\cdot \mid x_{\mathrm{stu}}, y_{<t}).$$

Both distributions are temperature-scaled before the softmax, as in every forward-KL loss in this paper. With valid-token set $\mathcal{M}$, the implemented OPSD objective is

$$\mathcal{L}_{\mathrm{OPSD}} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \sum_{j \in \mathcal{V}} \min\left(\mathbf{p}_t^{\mathrm{opsd}}(j) \log \frac{\mathbf{p}_t^{\mathrm{opsd}}(j)}{p_t(j)}, \tau_{\mathrm{clip}} \right). \tag{1}$$

The clipping in Eq. (1) is element-wise over vocabulary terms: for each $j \in \mathcal{V}$, the quantity $\mathbf{p}_t^{\mathrm{opsd}}(j) \log(\mathbf{p}_t^{\mathrm{opsd}}(j)/p_t(j))$ is clipped, the clipped terms are summed over the vocabulary, and the result is averaged over valid tokens. This objective is the uniform-weight baseline whose outer token aggregation PW-OPSD will modify.

3.2 Motivation: Token position predicts teacher reliability

Branch-viability protocol. The branch-viability diagnostic asks whether a locally ambiguous teacher preference is reliable supervision for the student distribution. Throughout the diagnostic, "teacher" and "student" refer to two prompt templates applied to the same Qwen3-4B model: the *teacher* template includes the privileged ground-truth answer, while the *student* template is the ordinary problem prompt without that information.

Table 1: Branch-viability classification on Qwen3-4B. Entries are within-problem residualized AUROC for separating 8 *real-uncertain* candidates (whose forced student-context continuations fail to recover the correct answer) from 271 *diversity* candidates (whose continuations remain viable) across 61 problems, restricted to the correct-spine subset and to high-truncated-entropy candidate positions. For position, the scored feature is $1 - \widetilde{r}$, so larger values correspond to earlier tokens. Cluster-bootstrap 95% CIs are computed over problems.

Score (residualized within problem)	Residual AUROC	95% CI
$H(\mathbf{p}_t^{\mathrm{opsd}})$ (mean predictive entropy)	0.51	$[0.50, 0.84]$
MI (MC-dropout epistemic)	0.51	$[0.50, 0.62]$
$\log \hat{\kappa}$ (Dirichlet precision)	0.54	$[0.50, 0.69]$
h_{trunc} (top-16 valid entropy)	0.57	$[0.51, 0.72]$
$1 - \widetilde{r}$ (early-position score)	**0.83**	$[\mathbf{0.66}, \mathbf{0.95}]$

For each problem, the model under the student template first generates an on-policy rollout; this fixed token sequence is the *spine*. The model is then re-evaluated along the spine under the teacher template, which adds the privileged answer. Candidate positions are pre-selected from high top-16 valid-token truncated entropy on this teacher pass; Appendix A gives the filtering details. The diagnostic is deliberately targeted: it studies positions where the teacher already signals local ambiguity, rather than arbitrary positions on the rollout. The reliability-weighted view in Section 3.3 identifies the latent reliability posterior as the formal object that this diagnostic probes behaviorally.

At each candidate position, the teacher proposes preferred next-token alternatives. We then test whether each alternative leads back to the correct answer *from the student's perspective*: we form a prefix consisting of the student-template prompt, the spine truncated to the candidate position, and the forced alternative token, and complete the sequence under the student template (no privileged information). A candidate is labeled *real-uncertain* when these student-context continuations fail to recover the correct final answer; in this case the teacher's local alternatives are unreliable targets for that student prefix. A candidate is labeled *diversity* when the alternatives remain viable routes to the same answer. We continue under the student template rather than the teacher template because student rollouts and deployment use the ordinary prompt without privileged information, and a teacher-template continuation can recover from almost any forced branch by re-using the ground-truth answer in its prompt, which would mask the failures that the diagnostic measures. Appendix A gives the thresholded implementation details for the repeated forced-continuation rollouts.

Empirical finding. Table 1 reports 8 real-uncertain and 271 diversity candidates from 61 problems, restricted to the correct-spine subset; the underlying failure-rate-by-position curves are visualized in Fig. 1. Each feature is residualized within problem before scoring, so the AUROC reflects within-problem token-level separation rather than problem difficulty. Confidence intervals are cluster bootstraps over problems.

Position is the strongest predictor among the diagnostics we test. We define normalized position as $\widetilde{r} = (k + 0.5)/L$, where k is the zero-based spine index and L is the spine length. Since the real-uncertain candidates concentrate early, Table 1 reports the oriented early-position score $1 - \widetilde{r}$ for the real-uncertain label. This score reaches AUROC 0.83 with 95% CI $[0.66, 0.95]$, whereas predictive entropy, MC-dropout mutual information, Dirichlet precision, and top-16 truncated entropy remain weak predictors. Equivalently, raw position $\widetilde{r}$ is negatively associated with teacher unreliability, which is why PW-OPSD uses an increasing reliability weight.

The local diagnostics in Table 1 include standard teacher-side uncertainty scores: predictive entropy [Kendall and Gal, 2017], MC-dropout mutual information [Gal and Ghahramani, 2016], and Dirichlet precision and evidential-style categorical uncertainty [Malinin and Gales, 2018, Sensoy et al., 2018], plus our top-16 truncated entropy. They are included as diagnostic baselines for reliability prediction; implementation details for the MC-dropout scores are in Appendix B.

Robustness and mechanism. Position leads each local score in its a priori orientation by 0.29–0.40 paired AUROC on the same candidates, with every interval excluding zero (Appendix M.2). Without the entropy pre-filter, the local scores rise to 0.58–0.64, yet position remains the strongest predictor (0.78); on a chemistry benchmark the trend replicates (0.62, interval excluding 0.5). Forcing the teacher's fourth- to sixth-ranked tokens instead of its top-3 alternatives yields the same early loss of viability, so early failures are a property of the position in the student trajectory, not of the teacher's

particular preferences. This agrees with the compounding of early deviations in autoregressive generation and identifies the failure mode as structural to on-policy distillation: teacher-side signals cannot locate it, whereas a position weight addresses it directly. The label is defined on correct spines (Appendix M.4); training uses every rollout, so the downstream results measure the weight on the full rollout distribution.

Implication for adaptive distillation signals. The weak scores in Table 1 call into question a common assumption behind recent adaptive, relaxed, or privileged distillation methods: that local uncertainty or teacher-student discrepancy can serve as a reliable control signal for distillation. Existing methods instantiate this idea through teacher token entropy in EOPD [Jin et al., 2026], token-wise teacher-student probability log-ratios in ToDi [Jung et al., 2025], teacher-student log-likelihood-ratio rewards and entropy-guided token-level sampling in REOPOLD [Ko et al., 2026], and tutor-answer consensus in GATES [Stein et al., 2026]. These signals are useful for adapting objectives, rewards, or trajectory selection, but our diagnostic shows that they are not reliable proxies for branch viability: the same large or permissive signal can arise when the teacher is unreliable and when multiple continuations are genuinely viable. Position provides a complementary structural signal, capturing where local ambiguity becomes unreliable for the student.

3.3 Theoretical interpretation: Reliability-weighted distillation

Let $c_t = (x_{\mathrm{stu}}, x_{\mathrm{tch}}, y_{<t})$ denote the distillation context, let q_t be the privileged teacher target at that context, and let $p_t = \pi_\theta(\cdot \mid x_{\mathrm{stu}}, y_{<t})$ be the student distribution. For the interpretation, define the unclipped token divergence $D_t(\theta) = \mathrm{KL}(q_t \,\|\, p_t)$. Introduce a latent indicator $I_t \in \{0, 1\}$, where $I_t = 1$ means that the teacher's local target is reliable for the student prefix: matching it helps preserve or recover the correct solution rather than following an off-path alternative. The ideal reliability-filtered risk is

$$R(\theta) = \mathbb{E}[I_t \cdot D_t(\theta)]\,. \tag{2}$$

For fixed θ, $D_t(\theta)$ is determined by the distillation context, so the tower property gives

$$R(\theta) = \mathbb{E}[\rho^*(c_t) \cdot D_t(\theta)]\,, \qquad \rho^*(c_t) = \Pr(I_t = 1 \mid c_t). \tag{3}$$

The Bayes-optimal surrogate for this latent-reliability risk weights each token by the posterior probability that the teacher target is reliable. Appendix C gives the conditioning details. Related on-policy and adaptive distillation methods instead modify the sampling distribution, the divergence, or the weight functional [Agarwal et al., 2024, Gu et al., 2024, Ko et al., 2024, Wen et al., 2023, Wu et al., 2025, Jung et al., 2025, Jin et al., 2026]; PW-OPSD uses normalized position as a low-cost structural proxy for that posterior.

The branch issue can also be summarized by a mixture identity. Let $h_t = (x_{\mathrm{stu}}, y_{<t})$ denote the student-side prefix. Suppose the teacher target at that prefix is a mixture over latent successful branches Z,

$$q_t = \sum_z \alpha(z \mid h_t) q_t^z.$$

For any student distribution p_t,

$$\mathbb{E}_{z\sim\alpha(\cdot\mid h_t)}[\mathrm{KL}(q_t^z \,\|\, p_t)] = \mathrm{KL}(q_t \,\|\, p_t) + I_q(Y_t; Z \mid h_t), \tag{4}$$

where $I_q(\cdot;\cdot \mid \cdot)$ is conditional mutual information under the teacher mixture. The mutual-information term quantifies branch-specific variation hidden by the marginal teacher target; because it is independent of p_t, the identity is not by itself a different gradient objective. The corresponding sequence-level identity accumulates this conditional mutual information across time; Appendix D gives the details and interpretation.

The theory thus specifies the ideal weight, the reliability posterior $\rho^*(c_t)$, but not its shape; the diagnostic supplies an empirical estimate of how it varies along the trajectory. Early high-ambiguity positions are most often labeled real-uncertain, and fitting the sigmoid form of Eq. (5) to the diagnostic's failure curve recovers the transition of our default schedule ($\tau = 0.30$, $s = 0.10$; Appendix M.8), so PW-OPSD discounts early positions with a lower outer weight.

3.4 PW-OPSD: Position-weighted on-policy self-distillation

PW-OPSD uses a deterministic structural proxy for the reliability posterior $\rho^*(c_t)$: a token's relative position in its own student rollout. For sequence i with valid completion length L_i, define the one-based valid token index $t \in \{1, \dots, L_i\}$ and the position fraction

$$r_{i,t} = \frac{t - 0.5}{L_i}.$$

The position weight is

$$w_{i,t} = w_{\min} + (1 - w_{\min})\sigma\left(\frac{r_{i,t} - \tau}{s}\right), \tag{5}$$

with defaults $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$. The floor keeps early tokens partially supervised, the threshold places the transition early in the rollout, and the scale avoids a hard discontinuity. The schedule is increasing because raw position is positively associated with reliability in the branch-viability diagnostic.

The position fraction is defined relative to each sequence's own valid length, not the batch-padded maximum, and the loss averages within each sequence before averaging across valid sequences. The two choices interact: under a global token mean, the position weight would also reweight whole sequences through their length profile, and longer rollouts would receive more gradient mass purely because they contain more tokens. The 2×2 ablation in Appendix L separates the two factors and attributes most of the gain to position weighting; an explicitly entropy-gated variant is analyzed in Appendix N.

With B valid sequences in the batch, the implemented PW-OPSD objective is

$$\mathcal{L}_{\text{PW-OPSD}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{L_i}\sum_{t=1}^{L_i} w_{i,t}\sum_{j\in\mathcal{V}}\min\left(\mathbf{p}_{i,t}^{\text{opsd}}(j)\log\frac{\mathbf{p}_{i,t}^{\text{opsd}}(j)}{\pi_\theta(j \mid x_{\text{stu}}^i, y_{<t}^i)}, \tau_{\text{clip}}\right). \tag{6}$$

The inner term is the same per-vocabulary clipped forward KL as Eq. (1). PW-OPSD uses the ordinary single-pass teacher target $\mathbf{p}_{i,t}^{\text{opsd}}$; its additional computation over OPSD is only the scalar sigmoid in Eq. (5).

4 Experiments

Section 3.2 suggests a simple reliability prior: early high-ambiguity branch points are more likely to be teacher-unreliable, while later positions are more reliable under the branch-viability diagnostic. This section tests whether the corresponding position-weighted objective improves downstream reasoning behavior relative to OPSD under the same rollout, teacher pass, and evaluation protocol.

4.1 Setup

Models, baselines, and schedule. We train the Qwen3-4B checkpoint [Yang et al., 2025, Qwen Team, 2025] on OpenThoughts-Math-30k [Guha et al., 2025, siyanzhao, 2026] for the main comparison (Table 2), the position-schedule sweep (Table 3), and the reduction-positioning ablation (Table 6); the cross-model evidence in Table 4 additionally uses DeepSeek-R1-Distill-Llama-8B (DSR1-L8B) [Guo et al., 2025, DeepSeek-AI, 2025] and Olmo-3-7B-Think [Allen Institute for AI, 2026]. PW-OPSD uses the Moderate schedule $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$, the diagnostic-derived default. We compare against three baselines: OPSD [Zhao et al., 2026] as the uniform-weight reference, EOPD [Jin et al., 2026] as a representative entropy-conditioned adaptive-KL baseline, and REOPOLD [Ko et al., 2026] as a cross-family policy-gradient adaptive on-policy distillation baseline that controls for whether any adaptive per-token signal recovers the downstream pattern attributed to position. All methods share the OPSD privileged on-policy training setup with LoRA [Hu et al., 2022] (rank 64, $\alpha = 128$), are evaluated at the 100-step checkpoint following OPSD, and otherwise use their published defaults. Full training hyperparameters, the cross-family rationale, and the PW-OPSD pseudocode are in Appendices G and I.

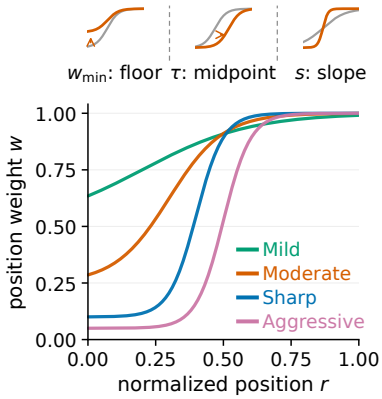
Table 2: Qwen3-4B results with maximum generation length 38,912 tokens. Entries report mean ± standard deviation across three evaluation seeds. Bold marks the best value for each benchmark-metric column and any value within 0.1 pp of the leader. The Aggressive row is included to show schedule sensitivity rather than as a separate method claim.

Method	Benchmark	Pass@12	Avg@12	Maj@12
OPSD [Zhao et al., 2026]	MATH-500	98.00 ± 0.35	95.33 ± 0.08	96.73 ± 0.12
	AIME 2024	86.67 ± 0.00	75.19 ± 0.42	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	$\mathbf{83.33 \pm 0.00}$	66.67 ± 1.27	73.33 ± 3.33
	HMMT 2025	$\mathbf{64.44 \pm 1.92}$	43.89 ± 0.73	51.11 ± 3.85
EOPD [Jin et al., 2026]	MATH-500	$\mathbf{98.33 \pm 0.23}$	95.33 ± 0.08	96.73 ± 0.23
	AIME 2024	85.56 ± 1.92	73.61 ± 2.17	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	82.22 ± 1.92	65.65 ± 0.89	$\mathbf{76.67 \pm 3.33}$
	HMMT 2025	63.33 ± 3.33	41.94 ± 2.00	50.00 ± 3.33
REOPOLD [Ko et al., 2026]	MATH-500	$\mathbf{98.40 \pm 0.20}$	95.09 ± 0.09	96.53 ± 0.31
	AIME 2024	86.67 ± 0.00	73.98 ± 0.42	$\mathbf{80.00 \pm 3.33}$
	AIME 2025	81.11 ± 1.92	62.13 ± 1.67	74.44 ± 5.09
	HMMT 2025	58.89 ± 1.92	41.39 ± 1.44	51.11 ± 3.85
PW-OPSD Moderate (**ours**)	MATH-500	$\mathbf{98.40 \pm 0.00}$	95.34 ± 0.10	96.67 ± 0.12
	AIME 2024	85.56 ± 1.92	$\mathbf{76.20 \pm 0.58}$	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	$\mathbf{83.33 \pm 0.00}$	$\mathbf{67.78 \pm 1.27}$	74.44 ± 1.92
	HMMT 2025	60.00 ± 3.33	43.33 ± 1.21	$\mathbf{52.22 \pm 1.92}$
PW-OPSD Aggressive (**ours**)	MATH-500	$\mathbf{98.40 \pm 0.20}$	$\mathbf{95.53 \pm 0.04}$	$\mathbf{97.07 \pm 0.12}$
	AIME 2024	$\mathbf{87.78 \pm 1.92}$	75.19 ± 0.85	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	$\mathbf{83.33 \pm 0.00}$	67.59 ± 0.85	75.56 ± 1.92
	HMMT 2025	60.00 ± 0.00	$\mathbf{45.37 \pm 0.58}$	$\mathbf{52.22 \pm 1.92}$

Evaluation. We follow the OPSD evaluation setting [Zhao et al., 2026] with maximum generation length 38,912 tokens, 12 samples per problem, temperature $T = 1.0$, top-$p = 0.95$, top-k disabled, and `enable_thinking=True`. Benchmarks are MATH-500 [Hendrycks et al., 2021, Lightman et al., 2024, HuggingFaceH4, 2024], AIME 2024 [HuggingFaceH4, 2025], AIME 2025 [yentinglin, 2025], and HMMT 2025 [Dekoninck et al., 2026]. MATH-500 serves as a saturation control (all methods lie within 0.5 pp of about 95% Avg@12). HMMT 2025 and the two additional model families are never used by the diagnostic of Section 3.2, and the three schedule constants stay fixed across all benchmarks and models. We report Pass@12, Avg@12, and Maj@12; Appendix J gives the formulas. For each method-benchmark pair we run three random evaluation seeds and report mean ± across-seed sample standard deviation; the cross-model assessment (Table 4) applies the same four-benchmark three-seed protocol to two additional checkpoints from different model families.

4.2 Main results on Qwen3-4B

Table 2 reports the Qwen3-4B comparison with maximum generation length 38,912 tokens. We report the diagnostic-derived *Moderate* schedule $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$ as the default PW-OPSD configuration. We also include *Aggressive* $(0.05, 0.50, 0.05)$ from the schedule sweep as a sensitivity variant that applies a stronger early discount. The main pattern is that position weighting improves per-sample reasoning accuracy on the AIME benchmarks while preserving near-saturated MATH-500 performance. On MATH-500, OPSD, EOPD, and PW-OPSD Moderate are tied within 0.01 pp Avg@12, while PW-OPSD Aggressive gives the highest MATH-500 Avg@12 and Maj@12 in the table. On AIME 2024 and AIME 2025, PW-OPSD Moderate improves Avg@12 over OPSD by +1.0 pp and +1.1 pp, respectively. The local or adaptive alternatives do not reproduce this Avg@12 pattern: EOPD trails OPSD by 1.6 pp on AIME 2024 and 1.0 pp on AIME 2025, while REOPOLD trails by 1.2 pp and 4.5 pp. HMMT 2025 highlights the role of the early-discount strength. The Moderate schedule improves Maj@12 over OPSD by +1.11 pp but trails by 0.6 pp Avg@12 and 4.4 pp Pass@12. Under the stronger early discount of the Aggressive schedule, HMMT 2025 Avg@12 rises to 45.37, a +1.48 pp gain over OPSD, and Maj@12 remains +1.11 pp above OPSD. Of the three methods compared with OPSD, only PW-OPSD improves Avg@12 over it: Moderate gains +1.0 pp on AIME 2024 and +1.1 pp on AIME 2025, and Aggressive gains +1.5 pp on HMMT 2025.



Config	Benchmark	Pass@12	Avg@12	Maj@12
Mild	MATH-500	98.33 ± 0.12	95.44 ± 0.05	96.87 ± 0.12
	AIME 2024	87.78 ± 3.85	73.98 ± 0.85	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	$\mathbf{84.44 \pm 1.92}$	67.59 ± 1.16	73.33 ± 0.00
	HMMT 2025	62.22 ± 1.92	45.28 ± 0.73	47.78 ± 1.92
Moderate	MATH-500	$\mathbf{98.40 \pm 0.00}$	95.34 ± 0.10	96.67 ± 0.12
	AIME 2024	85.56 ± 1.92	$\mathbf{76.20 \pm 0.58}$	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	83.33 ± 0.00	67.78 ± 1.27	74.44 ± 1.92
	HMMT 2025	60.00 ± 3.33	43.33 ± 1.21	$\mathbf{52.22 \pm 1.92}$
Sharp	MATH-500	98.13 ± 0.31	95.36 ± 0.15	96.60 ± 0.20
	AIME 2024	$\mathbf{88.89 \pm 7.70}$	75.19 ± 1.53	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	83.33 ± 0.00	$\mathbf{68.80 \pm 0.89}$	$\mathbf{75.56 \pm 3.85}$
	HMMT 2025	$\mathbf{64.44 \pm 1.92}$	44.26 ± 1.12	46.67 ± 3.33
Aggressive	MATH-500	$\mathbf{98.40 \pm 0.20}$	$\mathbf{95.53 \pm 0.04}$	$\mathbf{97.07 \pm 0.12}$
	AIME 2024	87.78 ± 1.92	75.19 ± 0.85	$\mathbf{80.00 \pm 0.00}$
	AIME 2025	83.33 ± 0.00	67.59 ± 0.85	$\mathbf{75.56 \pm 1.92}$
	HMMT 2025	60.00 ± 0.00	$\mathbf{45.37 \pm 0.58}$	$\mathbf{52.22 \pm 1.92}$

Figure 2: Per-token weight schedules for the four configurations; top panels label the role of each $(w_{\min}, \tau, s)$ parameter. Table 3: Position-schedule sweep on Qwen3-4B (maximum generation length 38,912 tokens). Each cell is mean $\pm$ standard deviation across three evaluation seeds. Bold marks the best value for each benchmark-metric column within the sweep. Configurations: Mild $(0.50, 0.20, 0.20)$, Moderate $(0.25, 0.30, 0.10)$, Sharp $(0.10, 0.40, 0.05)$, Aggressive $(0.05, 0.50, 0.05)$.

EOPD and REOPOLD instead regress on at least three of four benchmarks, losing 1.1 and 2.1 pp Avg@12 on average.

4.3 Position-schedule ablation

The main comparison fixes the PW-OPSD schedule to $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$. Since the method introduces three scalar parameters, we also run a schedule sweep over four settings to test whether the effect is tied to a single parameter tuple. Figure 2 shows the schedule shapes; Table 3 reports the Qwen3-4B results for all four configurations under the same maximum generation length as the main table.

The four settings vary how strongly early tokens are discounted. Mild $(0.50, 0.20, 0.20)$ keeps a high early floor and uses the softest transition; Moderate $(0.25, 0.30, 0.10)$ is the chosen configuration from Table 2; Sharp $(0.10, 0.40, 0.05)$ lowers the floor and makes the transition steeper; Aggressive $(0.05, 0.50, 0.05)$ applies the strongest early down-weighting and delays the transition furthest. Thus lower $w_{\min}$ makes the early-token discount more aggressive, larger τ moves the transition later, and smaller s makes it sharper.

On MATH-500, every configuration sits within 0.2 pp Avg@12, consistent with saturation. On AIME 2024, Moderate leads on Avg@12 but the other three trail by at most 2.2 pp and all four tie on Maj@12; AIME 2025 Avg@12 is similarly clustered (67.59–68.80). HMMT 2025 separates the configurations more clearly: Aggressive and Mild gain $+1.9$–2.0 pp Avg@12 over Moderate. No configuration dominates uniformly. We retain Moderate in the main table because it is the schedule chosen *a priori* from the diagnostic curves, not because it is the best across this sweep. Averaged across the four benchmarks, all four schedules improve Avg@12 over OPSD (70.27) by $+0.30$ to $+0.65$ pp and span only 0.35 pp among themselves, confirming PW-OPSD is robust to the choice of $(w_{\min}, \tau, s)$.

Reduction vs. positioning. PW-OPSD differs from OPSD along two axes: a position-dependent token weight w_t (positioning) and a per-rollout average of the token loss before the batch mean (per-sequence reduction). The full 2×2 factorial over all four benchmarks in Table 6 (Appendix L) attributes most of the gain to positioning: relative to OPSD, the reduction change alone raises the four-benchmark Avg@12 by $+0.13$ pp, position weighting alone by $+0.25$ pp, and the joint configuration by $+0.40$ pp (computed from unrounded means).

4.4 Cross-model transfer of the position schedule

To test whether the diagnostic-derived Moderate schedule transfers beyond Qwen3-4B, we evaluate PW-OPSD on two larger models from different families: DeepSeek-R1-Distill-Llama-8B (8B, Llama family) [Guo et al., 2025, DeepSeek-AI, 2025] and Olmo-3-7B-Think (7B, OLMo

Table 4: Cross-model Avg@12 evidence for PW-OPSD (Moderate, no per-model retuning) versus the OPSD baseline of Zhao et al. [2026] across three model families with maximum generation length $38{,}912$ tokens. Entries report mean $\pm$ across-seed sample standard deviation over three evaluation seeds; the Avg@12 column is the equal-weight mean across the four benchmarks, computed from unrounded seed-level values. Bold marks the within-model winner in each numeric column, including values within 0.1 pp of the leader.

Model	Method	MATH-500	AIME 2024	AIME 2025	HMMT 2025	Avg@12
Qwen3-4B	OPSD	$\mathbf{95.33 \pm 0.08}$	75.19 ± 0.42	66.67 ± 1.27	$\mathbf{43.89 \pm 0.73}$	70.27
	PW-OPSD (**ours**)	$\mathbf{95.34 \pm 0.10}$	$\mathbf{76.20 \pm 0.58}$	$\mathbf{67.78 \pm 1.27}$	43.33 ± 1.21	**70.66**
DSR1-L8B	OPSD	$\mathbf{89.02 \pm 0.19}$	40.74 ± 2.58	31.57 ± 1.28	20.93 ± 0.80	45.56
	PW-OPSD (**ours**)	88.72 ± 0.25	$\mathbf{41.20 \pm 0.32}$	$\mathbf{32.22 \pm 0.56}$	$\mathbf{21.48 \pm 0.42}$	**45.91**
Olmo-3-7B-Think	OPSD	$\mathbf{96.20 \pm 0.08}$	$\mathbf{72.78 \pm 0.28}$	67.41 ± 2.16	$\mathbf{43.89 \pm 0.73}$	70.07
	PW-OPSD (**ours**)	$\mathbf{96.26 \pm 0.19}$	72.59 ± 1.53	$\mathbf{69.63 \pm 1.12}$	$\mathbf{43.80 \pm 0.58}$	**70.57**

family) [Allen Institute for AI, 2026]. For each model we compare PW-OPSD (Moderate, $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$) against the OPSD baseline under the same training and evaluation protocols as the Qwen3-4B main experiments: the full four-benchmark suite (MATH-500, AIME 2024, AIME 2025, HMMT 2025), three independent evaluation seeds per cell, and the maximum generation length of $38{,}912$ tokens described in Section 4.1. The Moderate schedule is held fixed across all three models with no per-model retuning. For DSR1-L8B and Olmo-3-7B-Think, both trained as reasoning models, we additionally apply a tokenizer-level $\langle$think$\rangle$-closure that replicates `enable_thinking=False` on chat templates that ignore the flag, matching the training-time student prompt format used on Qwen3-4B (evaluation continues to use `enable_thinking=True` as described in Section 4.1).

Table 4 shows that the same Moderate schedule gives positive four-benchmark Avg@12 gains on every tested model (computed from unrounded means): $+0.40$ pp on Qwen3-4B (4B parameters, Qwen family), $+0.34$ pp on DeepSeek-R1-Distill-Llama-8B (8B, Llama family), and $+0.50$ pp on Olmo-3-7B-Think (7B, OLMo family). Under the same protocol, neither adaptive baseline improves on all three families (EOPD: -1.13, $+0.30$, and $+0.12$ pp; REOPOLD: -2.12, -0.15, and $+0.81$ pp). The pattern spans Qwen, Llama, and OLMo checkpoints across a 4B–8B parameter range, with no per-model schedule retuning, supporting the model-family portability of the positional reliability signal identified in Section 3.2.

5 Conclusion, limitations, and future work

In this paper, we study teacher-token reliability in on-policy self-distillation. To test whether high teacher entropy reflects low reliability or benign solution diversity, we design a branch-viability diagnostic, which shows that reliability is positionally structured (AUROC 0.83 vs. ≤ 0.57 for local uncertainty). Inspired by this finding, we introduce PW-OPSD, an increasing sigmoid position weight within the OPSD training framework. Our experiments show that PW-OPSD improves Avg@12 over OPSD on AIME 2024 ($+1.0$ pp), AIME 2025 ($+1.1$ pp), and in the four-benchmark average on DeepSeek-R1-Distill-Llama-8B and Olmo-3-7B-Think. One limitation is that PW-OPSD is plug-and-play yet its gains over OPSD remain modest, leaving room to design more sophisticated methods built on the positional reliability finding. To improve the proposed method, we aim to design more dedicated objectives that go beyond pure position weighting, such as position-conditioned mixing of forward and reverse KL. The schedule is calibrated on mathematical reasoning; the diagnostic's positional structure also appears on a chemistry benchmark (Appendix M.6), and extending the training evaluation to other domains is a natural next step.

Acknowledgments and Disclosure of Funding

This paper is partially supported by Schmidt Sciences AI2050, Amazon Nova AI Challenge: Trusted Software Agents, Amazon Initiative for Interactive AI (AI2AI), Samsung Research via the LEAP-U program.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024.

Allen Institute for AI. Olmo-3-7B-Think. `https://huggingface.co/allenai/Olmo-3-7B-Think`, 2026. Hugging Face model card.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, G. Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mo Bavarian, Clemens Winter, P. Tillet, F. Such, D. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, S. Balaji, Shantanu Jain, William Saunders, Christopher Hesse, A. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, M. Knight, Miles Brundage, M. Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, I. Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

DeepSeek-AI. DeepSeek-R1-Distill-Llama-8B. `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B`, 2025. Hugging Face model card.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond Benchmarks: MathArena as an Evaluation Platform for Mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026. URL `https://arxiv.org/abs/2605.00674`.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, 2016.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-Policy Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024.

Arnav Gudibande, Eric Wallace, Charles Burton Snell, Xinyang Geng, Hao Liu, P. Abbeel, S. Levine, and Dawn Song. The false promise of imitating proprietary LLMs. *arXiv preprint arXiv:2305.15717*, 2023.

E. Guha, Ryan Marten, Sedrick Scott Keh, Negin Raoof, G. Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean-Pierre Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Ben Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, K. Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, V. Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, K. Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, J. Jitsev, Reinhard Heckel, M. Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni,

Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. doi: 10.1038/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021.

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

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-OPD: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

HuggingFaceH4. MATH-500. `https://huggingface.co/datasets/HuggingFaceH4/MATH-500`, 2024. Hugging Face dataset.

HuggingFaceH4. AIME 2024. `https://huggingface.co/datasets/HuggingFaceH4/aime_2024`, 2025. Hugging Face dataset for the 2024 competition; derived from AI-MO/aimo-validation-aime.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-Aware On-Policy Distillation of Language Models. *arXiv preprint arXiv:2603.07079*, 2026. URL `https://arxiv.org/abs/2603.07079`.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. ToDi: Token-wise Distillation via Fine-Grained Divergence Control. In *Conference on Empirical Methods in Natural Language Processing*, 2025.

Alex Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Conference on Empirical Methods in Natural Language Processing*, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and SeYoung Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, 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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *ACM Symposium on Operating Systems Principles*, 2023.

Hynek Kydlíček. Math-Verify: Math verification library. https://github.com/huggingface/math-verify, 2025. Version 0.6.1.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems*, 2018.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 2017.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, S. Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, M. Simens, Amanda Askell, Peter Welinder, P. Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, E. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Qwen Team. Qwen3-4B. https://huggingface.co/Qwen/Qwen3-4B, 2025. Hugging Face model card.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. CRISP: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

M. Sensoy, Lance M. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, 2018.

siyanzhao. OpenThoughts-Math-30k OPSD. https://huggingface.co/datasets/siyanzhao/Openthoughts_math_30k_opsd, 2026. Hugging Face dataset; train split with 29,434 examples.

Mingyang Song and Mao Zheng. A Survey of On-Policy Distillation for Large Language Models. *arXiv preprint arXiv:2604.00626*, 2026. URL https://arxiv.org/abs/2604.00626.

Alex Stein, Furong Huang, and Tom Goldstein. GATES: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

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

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL `https://aclanthology.org/2020.emnlp-demos.6/`.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models. In *International Conference on Computational Linguistics*, 2025.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL `https://arxiv.org/abs/2505.09388`.

yentinglin. AIME 2025. `https://huggingface.co/datasets/yentinglin/aime_2025`, 2025. Hugging Face dataset.

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

A Branch-viability protocol details

Models, prompts, and software. The diagnostic uses one `Qwen/Qwen3-4B` checkpoint [Qwen Team, 2025] (Hugging Face snapshot `1cfa9a72...3b3df60c`, dtype `bfloat16`) with no adapters loaded. "Teacher" and "student" denote two prompt templates applied to this single checkpoint: the teacher template includes the privileged ground-truth answer, and the student template is the ordinary problem prompt. vLLM and Hugging Face load this checkpoint as separate software backends, not as different trained weights. Problems are drawn from three sources, sampled without replacement (see attrition below): `MATH-500` test split [Hendrycks et al., 2021, Lightman et al., 2024, HuggingFaceH4, 2024], `AIME 2024` [HuggingFaceH4, 2025], and `AIME 2025` [yentinglin, 2025]. We use `vllm 0.11.0` with `tensor_parallel_size=4` (TP=4) for generation [Kwon et al., 2023], `transformers 4.57.1` for the Hugging Face (HF) teacher forward passes [Wolf et al., 2020], and `torch 2.8.0+cu128` on $4\times$H100 80 GB GPUs [Paszke et al., 2019].

Problem attrition. Phase A samples 24 MATH-500, 30 AIME 2024, and 30 AIME 2025 problems (84 total). Phase B produces $23 + 21 + 18 = 62$ correct-spine problems. After Phase F labeling, one AIME 2024 correct-spine problem has only gray candidates, so the binary-labeled pool used in Table 1 contains $23 + 20 + 18 = 61$ usable problems. Phase C then proposes up to 5 high-truncated-entropy candidates per problem (with the spacing / plausibility filters below); Phase E rolls out 6 student continuations per (candidate, alternative) pair. Table 1 is restricted to high-truncated-entropy candidate positions and to the 8 candidates labeled *real-uncertain* plus 271 *diversity* candidates across these 61 problems. Figure 1 uses the broader continuous-viability subset before binary thresholding.

Random seeds. Phase A fallback sampling, Phase E forced continuations, and the $\widetilde{r}$ residualization use vLLM seed `0`; Phase D MC-dropout masks use a per-position random state derived from the model's default generator. The cluster bootstrap in Phase G uses NumPy seed `0`.

Score residualization and AUROC. Within each problem p, every raw uncertainty score u (or its sign-oriented form for position) is mean-centered: $u' = u - \bar{u}_p$. We compute AUROC on the residualized scores against the binary label defined in Phase F, using the standard Mann-Whitney U formula with mid-rank tie handling. The cluster bootstrap resamples problems (not candidates) with replacement and recomputes the within-problem residualization on each resample, giving the 95% intervals reported in Table 1.

- **Phase A.** vLLM (TP=4) student greedy decode of 84 problems in total (24 from MATH-500, 30 from AIME 2024, and 30 from AIME 2025; the protocol is run independently per dataset and the labeled candidates are pooled in Phase G), with fallback to T=0.7 for problems where greedy does not reach `\boxed` within 16K tokens.
- **Phase B.** HF forward of the same checkpoint under the teacher template (privileged-answer prompt + spine), giving per-position teacher distributions and top-16 valid-token truncated entropy.
- **Phase C.** Apply junk / EOS / special filters, plausibility filter $p_2 \geq 0.02 \wedge p_2/p_1 \geq 0.10$, 64-token spacing constraint, and skip post-`\boxed` positions. Keep up to 5 high-truncated-entropy candidate positions per problem; for each, record the teacher pass's top-3 valid child tokens as forced alternatives.
- **Phase D.** HF teacher MC dropout (M=5, p=0.1, last 4 layers) at each candidate; record H_{full}, MI, and $\log \hat{\kappa}$.
- **Phase E.** vLLM (TP=4) *student*-context forced continuation: each forced child token from Phase C is appended to the student-template prompt + spine truncated to the candidate position, and rollouts continue under the student template (no privileged information). We use student-template continuation rather than teacher-template continuation because a teacher-template continuation can re-use the privileged answer in its prompt and recover from almost any forced child, collapsing the labels toward diversity. Across the three datasets, up to $(24 + 30 + 30) \times 5 \times 3 \times 6 = 7560$ attempted rollouts at T=1.0, `top_p=0.95`, with `max_tokens` dynamically clipped per request to fit `max_model_len=32K`.
- **Phase F.** For each forced child, define viability as the fraction of its 6 student-context continuations whose extracted boxed answer matches the ground truth. Label a candidate *diversity* if at least two of its children have viability $\geq V_{\text{high}}$=0.75; label it *real-uncertain*

if every child has viability $< V_{\text{low}}{=}0.40$ and the mean child viability is $< V_{\text{low}}$; otherwise label "gray" and exclude from the AUROC.
- **Phase G.** AUROC + area-under-precision-recall-curve (AUPRC) + cluster-bootstrap by problem (2,000 resamples, multiplicity-preserved).

Privileged teacher prompt. We use the following template (wrapped in the Qwen3 chat template with `enable_thinking=True`):

> *You are a privileged teacher. Solve the problem and end your solution with the correct boxed answer.*
> *Problem: {problem}*
> *Privileged ground-truth final answer: {answer}*
> *Use the ground-truth answer above. Produce a complete step-by-step solution that ends with* $\backslash\text{boxed}\{\{$*answer*$\}\}$.

This template simplifies the training-time teacher prompt of OPSD and conditions the teacher's autoregressive distribution on the ground-truth answer throughout generation.

B MC-dropout diagnostic implementation

MC dropout is diagnostic-only in this paper. It is used to compute H_{full}, MI, and $\log \hat{\kappa}$ scores for Table 1; it is not used to construct a training target or a token weight. All training methods use the ordinary single-pass teacher target $\mathbf{p}_t^{\text{opsd}}$.

MC dropout via forward hooks. Modern Qwen / Llama transformer blocks do not expose an `nn.Dropout` submodule. We inject MC dropout at inference time via forward hooks on the last 4 transformer layers; the hook applies $\text{F.dropout}(h, p, \text{training}{=}\text{True})$ to the layer output. We use $p = 0.1$ throughout and $M = 5$ MC samples. The unperturbed privileged-teacher forward provides $\mathbf{p}_t^{\text{opsd}}$; the perturbed forwards are used only for the diagnostic uncertainty scores. Hooks are removed in a `try/finally` block to prevent leaking dropout into subsequent forward passes.

Numerical stability of $\hat{\kappa}_t$. The moment-matching estimator for $\hat{\kappa}_t$ can produce negative values when the sample variance trace exceeds the categorical variance trace $1 - \|\bar{\mathbf{p}}_t\|_2^2$ (an artifact of small M). For Table 1, we do not apply any floor or log-normalization. We store the raw $\hat{\kappa}_t$ estimate and compute $\log \max(\hat{\kappa}_t, \epsilon)$ only at score time, with $\epsilon = 10^{-6}$ for the log transform.

Incremental moment statistics. For very large vocabularies (Qwen3 vocab ≈ 152K), computing the full softmax per MC sample at every position is memory-bound. We compute moment statistics token-by-token and aggregate $\sum_m \|\mathbf{p}_t^{(m)}\|_2^2$ incrementally to avoid materializing the full per-sample distribution beyond one forward at a time. In Phase B of the diagnostic we still materialize the per-position softmax once for top-16 selection; this is a one-time pass per problem.

C Reliability-weighted surrogate details

Section 3.3 conditions the reliability posterior on the distillation context $c_t = (x_{\text{stu}}, x_{\text{tch}}, y_{<t})$. This conditioning makes the tower-property step explicit. For fixed θ and a fixed teacher, the token divergence $D_t(\theta) = \text{KL}(q_t \,\|\, p_t)$ is determined by c_t, so

$$\mathbb{E}[I_t D_t(\theta)] = \mathbb{E}[\mathbb{E}[I_t D_t(\theta) \mid c_t]] = \mathbb{E}[D_t(\theta)\,\Pr(I_t = 1 \mid c_t)]\,.$$

Conditioning only on the student prefix $h_t = (x_{\text{stu}}, y_{<t})$ would require an additional assumption, because the privileged teacher target can vary with the teacher prompt and reference information even when the student prefix is fixed.

The risk in Eq. (2) is also a distillation surrogate on sampled rollouts. As in standard OPSD implementations, the rollout is sampled from the current student and then treated as a fixed training example for the KL update; gradients pass through the student probabilities on visited prefixes, not through the sampling operation that produced those prefixes. The main-text interpretation uses the unclipped divergence $D_t(\theta)$ for notational clarity, while the implemented objectives use the element-wise clipped forward-KL surrogate described in Eq. (1).

D Branch-mixture identity and interpretation

Equation (4) gives the token-level identity used in the main text. With joint distribution $q(z, y_t \mid h_t) = \alpha(z \mid h_t) q_t^z(y_t)$ and marginal $q_t(y_t) = \sum_z \alpha(z \mid h_t) q_t^z(y_t)$,

$$\begin{aligned}\mathbb{E}_{z\sim\alpha}\,\mathrm{KL}(q_t^z \,\|\, p_t) &= \sum_{z,y_t} q(z, y_t \mid h_t) \log \frac{q_t^z(y_t)}{p_t(y_t)} \\ &= \mathrm{KL}(q_t \,\|\, p_t) + I_q(Y_t; Z \mid h_t).\end{aligned}$$

The mutual-information term is independent of p_t. The identity therefore decomposes branch-specific variation hidden by the marginal teacher target; it is not, by itself, a proof that marginal forward KL has a different gradient objective. The empirical branch-viability diagnostic supplies the task-specific direction used by PW-OPSD: early high-ambiguity positions more often correspond to unreliable supervision.

The same branch-mixture view yields a sequence-level form. Let the teacher's branch-conditioned next-token distribution be $q_t^z(y_t \mid y_{<t})$, and let the corresponding sequence distribution factorize autoregressively as

$$q^z(y_{1:T}) = \prod_{t=1}^{T} q_t^z(y_t \mid y_{<t}).$$

Its marginal under the latent branch prior α is $q(y_{1:T}) = \sum_z \alpha(z)\, q^z(y_{1:T})$. For a student sequence distribution $p(y_{1:T}) = \prod_{t=1}^{T} p_t(y_t \mid y_{<t})$,

$$\mathbb{E}_{z\sim\alpha}[\mathrm{KL}(q^z \,\|\, p)] = \mathrm{KL}(q \,\|\, p) + \sum_t I_q(Y_t; Z \mid Y_{<t}). \tag{7}$$

Thus the value of branch-conditioned sequence forward KL differs from marginal sequence forward KL by the sum of conditional mutual-information terms between the next token and the latent branch. As in the token-level identity, this sum quantifies branch ambiguity but is independent of the student distribution.

E Position-schedule hyperparameters

The Moderate schedule used in the main table sets $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$, placing the transition near the empirical early-to-late change in the branch-viability diagnostic while keeping a nonzero floor on early-token supervision. The full position sweep in Table 3 covers four schedules: **Mild** $(0.50, 0.20, 0.20)$, **Moderate** $(0.25, 0.30, 0.10)$ (the a priori choice from the diagnostic curve in Fig. 1), **Sharp** $(0.10, 0.40, 0.05)$, and **Aggressive** $(0.05, 0.50, 0.05)$. The main table reports both Moderate (the a priori default) and Aggressive (the sweep configuration with the highest HMMT 2025 Avg@12). We did not tune $(w_{\min}, \tau, s)$ on any evaluation result; the four configurations were chosen before observing any of the Avg@12/Pass@12/Maj@12 values.

F Adaptive-loss template and method comparison

Table 5 compares the three per-token distribution-matching objectives evaluated in Table 2 along the dimensions that the template makes explicit: target distribution, adaptive signal or reliability proxy, and reduction over tokens. REOPOLD [Ko et al., 2026] is also evaluated in Table 2; it does not fit this template because its gradient flows through the rolled-out token's log-prob rather than through a distributional forward KL. Its role in the comparison is summarized in Appendix F.1.

F.1 Comparison to related adaptive losses

Algorithm 1 (Appendix G) gives the explicit one-step pseudocode. For reference, the scalar-weighted clipped-FKL template is

$$\mathcal{L}_{\mathrm{wFKL}} = \frac{1}{|\mathcal{M}|} \sum_{t\in\mathcal{M}} w_t \sum_{j\in\mathcal{V}} \min\left(\mathbf{p}_t(j) \log \frac{\mathbf{p}_t(j)}{\pi_\theta(j \mid x_{\mathrm{stu}}, y_{<t})}, \tau_{\mathrm{clip}} \right). \tag{8}$$

Different adaptive losses instantiate or modify this template through the choice of target, reliability proxy, inner divergence, and reduction.

OPSD uses a uniform weight [Zhao et al., 2026]. EOPD [Jin et al., 2026] gates a forward-KL augmentation to reverse KL using teacher entropy. PW-OPSD keeps the OPSD forward-KL inner loss and uses the position schedule $w_{i,t}$ from Eq. (5) in its outer token aggregation. REOPOLD is also included in the experimental comparison but is a policy-gradient distillation variant rather than a per-token forward-KL weighting rule [Ko et al., 2026], so it does not instantiate the template above; Appendix I gives the rationale for including it.

Table 5: Per-token distribution-matching distillation objectives compared in Table 2, parametrized by target distribution, adaptive signal or reliability proxy, and reduction over tokens. REOPOLD is also evaluated in Table 2 but follows a policy-gradient route rather than a per-token forward-KL weighting, so it is not instantiated in this template.

Method	Target / inner loss	Adaptive signal / proxy	Reduction
OPSD [Zhao et al., 2026]	clipped FKL to $\mathbf{p}_t^{\mathrm{opsd}}$	uniform ($w_t = 1$)	global token mean
EOPD [Jin et al., 2026]	entropy-gated RKL/FKL mixture	teacher-entropy gate	global token mean
PW-OPSD (**ours**)	clipped FKL to $\mathbf{p}_{i,t}^{\mathrm{opsd}}$	position $w_{i,t}$	per-sequence mean

G PW-OPSD training pseudocode

Algorithm 1 gives one training step of PW-OPSD. The procedure computes student and teacher log-probabilities at the distillation temperature, forms the forward-KL tensor with no reduction, clamps each vocabulary element, sums over the vocabulary, applies the position weight, averages over valid tokens within each sequence, and then averages over valid sequences. The sampled rollout is fixed for this update, as discussed in Appendix C.

Algorithm 1 One training step of PW-OPSD.

Require: Parameters θ, prompts $\{(x^i_{\mathrm{stu}}, x^i_{\mathrm{tch}})\}_{i=1}^{B}$, schedule $(w_{\min}, \tau, s)$, distillation temperature T_{distill}, clip τ_{clip}.

Sample student rollouts $y^i \sim \pi_\theta(\cdot \mid x^i_{\mathrm{stu}})$.
Let L_i be the number of valid generated tokens before EOS or truncation.
Score visited prefixes at T_{distill} under the privileged teacher context and ordinary student context to obtain $\mathbf{p}_{i,t}^{\mathrm{opsd}}$ and $p_{i,t}$ for each valid t.
for each sequence i and valid token $t \in \{1, \dots, L_i\}$ **do**
    $r_{i,t} \leftarrow (t - 0.5)/L_i$
    $w_{i,t} \leftarrow w_{\min} + (1 - w_{\min})\sigma((r_{i,t} - \tau)/s)$
    $\ell_{i,t} \leftarrow \sum_{j \in \mathcal{V}} \min\left(\mathbf{p}_{i,t}^{\mathrm{opsd}}(j) \log \frac{\mathbf{p}_{i,t}^{\mathrm{opsd}}(j)}{p_{i,t}(j)},\ \tau_{\mathrm{clip}} \right)$
end for
$\mathcal{L}_{\text{PW-OPSD}} \leftarrow B^{-1} \sum_{i=1}^{B} L_i^{-1} \sum_{t=1}^{L_i} w_{i,t} \ell_{i,t}$
Update θ using $\nabla_\theta \mathcal{L}_{\text{PW-OPSD}}$.

H Implementation conventions inherited from OPSD

The following implementation conventions are held fixed across the methods reported in this paper.

Right-padded prompts. The upstream OPSD data collator right-pads each batch of prompts to the length of its longest prompt, so every completion starts at the same offset and shorter prompts are followed by padding tokens; the loss covers every completion token and excludes the padding. This is held constant across all methods in Table 2. OpenThoughts-Math-30k [siyanzhao, 2026] prompts vary in length (median 93 tokens, max 826), so the amount of padding varies by batch; all methods share it.

Train/eval prompt template gap. Training prompts use the OPSD reference student template `Problem: {problem}\n\n Please reason...` with `enable_thinking=False`; evaluation prompts use the simpler form `{problem}\n\n Please reason...` with `enable_thinking=True`. This training/evaluation prompt-template difference is shared across all methods reported in this paper.

Per-vocabulary clip semantics. The clipped forward KL in Eq. (1) clamps each token/vocabulary entry of $\mathbf{p}_t^{\text{opsd}}(j) \log(\mathbf{p}_t^{\text{opsd}}(j)/p_t(j))$ element-wise via `F.kl_div(reduction='none').clamp(max=tau_clip)` before the inner sum over the vocabulary. This matches the OPSD reference implementation.

Gradient-accumulation token-mean. With per-microbatch token-mean reduction and gradient accumulation $= 2$, the OPSD loss computes the mean of two per-microbatch token-means rather than the exact token-weighted mean over the effective batch. This is the standard Hugging Face Trainer behavior and is held fixed for the OPSD baseline. PW-OPSD's per-sequence reduction does not have this issue because each microbatch contains the same number of sequences.

vLLM rollout seed. The vLLM colocate-mode rollout sampler is seeded as `accelerator.process_index // tp_size`, independent of the trainer `--seed` flag. Reruns of the same `--seed N` therefore produce slightly different rollouts. All evaluation results use seeded vLLM `SamplingParams.seed`.

I Evaluation setup

Models. The Qwen3-4B checkpoint [Yang et al., 2025, Qwen Team, 2025] is used in the main comparison (Table 2), the position-schedule sweep (Table 3), and the reduction-positioning ablation (Table 6); the cross-model evidence in Table 4 additionally uses DeepSeek-R1-Distill-Llama-8B [Guo et al., 2025, DeepSeek-AI, 2025] and Olmo-3-7B-Think [Allen Institute for AI, 2026], two larger models from different families. In every comparison we train LoRA adapters on a fixed checkpoint and merge the selected adapter before evaluation, so within each model block, differences between method rows come only from the distillation objective and not from the checkpoint weights.

Training. We use the OPSD privileged on-policy setup [Zhao et al., 2026]. Across methods, we hold fixed preprocessing and prompt templates, full-vocabulary clipped forward-KL conventions, LoRA [Hu et al., 2022] rank 64, $\alpha = 128$, dropout 0.05, learning rate 5×10^{-6}, `max_completion_length=1024`, distillation temperature $T_{\text{distill}} = 1.1$, KL clip $\tau_{\text{clip}} = 0.05$, a fixed teacher with LoRA disabled in the privileged forward pass, and seed 42. Training uses $4\times$H100 80 GB GPUs with effective batch size 32 (per-device batch 4 with gradient accumulation 2). All methods are evaluated at the 100-step checkpoint, following the OPSD evaluation horizon of Zhao et al. [2026]. PW-OPSD uses $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$ for the diagnostic-derived default schedule.

Baselines. We compare PW-OPSD against three baselines run under our common training and evaluation protocol: OPSD [Zhao et al., 2026] as the uniform-weight reference, EOPD [Jin et al., 2026] as a representative entropy-conditioned adaptive-KL alternative, and REOPOLD [Ko et al., 2026] as a representative policy-gradient adaptive on-policy distillation method. REOPOLD differs from the per-token forward-KL family in gradient form, because its gradient flows through the rolled-out token's log-prob rather than through a distributional forward KL. It nevertheless answers the same operational question of how to derive a per-token training signal from teacher-student log-likelihood-ratio rewards on sampled rollout tokens. Including REOPOLD therefore provides a cross-family reference point that controls for the possibility that any adaptive token weighting recovers the downstream pattern attributed to position. All baselines use their published default hyperparameters under the common training and evaluation protocol described above.

Evaluation setting. We use a maximum generation length of $38{,}912$ tokens: `max_new_tokens=38912`, $N = 12$ samples per problem, temperature $T = 1.0$, top-$p = 0.95$, top-k disabled, and `enable_thinking=True`, matching the OPSD evaluation setup [Zhao et al., 2026].

Metrics. We report Avg@12, Pass@12, and Maj@12, which together expose per-sample accuracy, search-style success under repeated attempts, and stability under aggregation. Majority vote uses math-equivalence clustering with `sympy` [Meurer et al., 2017] and `math_verify` [Kydlíček, 2025]; unformatted predictions are placed in a single `INVALID` cluster that participates in the plurality count and is scored incorrect when selected. Formal per-problem definitions are deferred to Appendix J.

Benchmarks. We evaluate on MATH-500 [Hendrycks et al., 2021, Lightman et al., 2024, HuggingFaceH4, 2024], AIME 2024 [HuggingFaceH4, 2025], AIME 2025 [yentinglin, 2025], and HMMT February 2025. The HMMT set is a cleaned copy of MathArena's `hmmt_feb_2025` release [Dekoninck et al., 2026], with its SHA-256 hash given in Appendix K. For each method-benchmark pair we run three random evaluation seeds and report mean $\pm$ across-seed sample standard deviation. The cross-model assessment (Table 4) extends this protocol to DeepSeek-R1-Distill-Llama-8B and Olmo-3-7B-Think under the same four benchmarks.

J Evaluation metric definitions

Multi-sample evaluation for reasoning. Reasoning models are commonly evaluated with repeated sampling because a single completion can understate the chance that the model finds a correct solution. Pass@N measures whether any of N samples succeeds [Chen et al., 2021, Brown et al., 2024], while self-consistency and majority vote aggregate multiple reasoning paths [Wang et al., 2023]. We report Avg@12, Pass@12, and Maj@12 throughout. These metrics expose different behaviors of a distillation objective: per-sample accuracy, search-style success under repeated attempts, and stability under aggregation.

Formal definitions. For a single problem with $N{=}12$ generated solutions $\{y^{(1)}, \ldots, y^{(N)}\}$, the predicted answer is extracted as the content of the last `\boxed{...}` in $y^{(i)}$; samples without a parseable boxed answer are assigned the cluster key `INVALID`. Let $c_i \in \{0, 1\}$ indicate whether $y^{(i)}$ is graded correct against the gold answer (using `math_verify` [Kydlíček, 2025] with a normalized string-equality fallback for parsing failures), and let a_i be the math-equivalence cluster key obtained by grouping the extracted answers under the `sympy` [Meurer et al., 2017] and `math_verify` normalization pipeline. The per-problem metrics are

$$\mathrm{Avg@}N = \frac{1}{N}\sum_{i=1}^{N} c_i, \tag{9}$$

$$\mathrm{Pass@}N = \mathbf{1}\Big[\textstyle\sum_{i=1}^{N} c_i \geq 1\Big], \tag{10}$$

$$\mathrm{Maj@}N = \begin{cases} 0, & a^\star = \texttt{INVALID}, \\ c_{i^\star}, & \text{otherwise}, \end{cases} \tag{11}$$

where the plurality cluster key is $a^\star \in \arg\max_{a \in \{a_1,\ldots,a_N\}} \#\{j : a_j = a\}$, with ties broken by smallest first occurrence $a^\star = a_{\min\{i : a_i \in \arg\max\}}$, and the representative index $i^\star = \min\{i : a_i = a^\star\}$. The `INVALID` cluster *participates in the plurality count* but is scored incorrect when selected, so a problem on which the model produces no parseable answer in the majority of samples scores $\mathrm{Maj@}N = 0$ even if some minority of samples were correct. Reported numbers are means of these per-problem metrics across the benchmark, averaged across the three evaluation seeds with a sample (Bessel-corrected) standard deviation.

K Reproducibility notes

The training and evaluation code accompanying this paper is available at `https://github.com/SaFo-Lab/PW-OPSD`. The main dataset sources and evaluation conventions are as follows.

- **Datasets.** MATH-500 is Hugging Face `HuggingFaceH4/MATH-500` [Hendrycks et al., 2021, Lightman et al., 2024, HuggingFaceH4, 2024]; AIME 2024 is Hugging Face `HuggingFaceH4/aime_2024` [HuggingFaceH4, 2025]; AIME 2025 is Hugging Face `yentinglin/aime_2025` [yentinglin, 2025]; HMMT February 2025 is a cleaned copy of

the MathArena `hmmt_feb_2025` release [Dekoninck et al., 2026], SHA-256 `87bfb23d2c887fab12b42fcc2b2dd8cb5a9d1070e591490dac8755bc366ea25e`; the training corpus is OpenThoughts-Math-30k [Guha et al., 2025, siyanzhao, 2026], the same corpus used by Zhao et al. [2026].

- **Maj@N evaluation.** Maj@N clusters all N predictions by math-equivalence using `sympy` [Meurer et al., 2017] and `math_verify` [Kydlíček, 2025], with unformatted predictions placed in a single `INVALID` cluster that participates in the plurality count and is scored incorrect when selected (see Appendix J for the formal definition).

L Reduction $\times$ positioning ablation

PW-OPSD differs from OPSD along two axes that are easy to conflate. The first is *positioning*: PW-OPSD multiplies the per-token loss by a position-dependent scalar $w_t \in [w_{\min}, 1]$, while OPSD uses $w_t \equiv 1$. The second is *reduction*: PW-OPSD averages the per-token loss within each rollout before averaging across the batch (a per-sequence reduction), while OPSD averages the loss over all valid tokens in the batch (a global token mean). Both changes alter how single-rollout positions contribute to the gradient, and a naive comparison conflates them. We run the full 2×2 factorial under identical training and evaluation settings; Table 6 reports Avg@12 on all four benchmarks.

Table 6: 2×2 ablation on Qwen3-4B: reduction (global token mean vs. per-sequence mean) by positioning (none vs. position-weighted). Avg@12 mean $\pm$ sample standard deviation across three evaluation seeds; the last column is the equal-weight four-benchmark mean computed from unrounded values. Bold marks the column maximum. The first and last rows (OPSD and PW-OPSD Moderate) report the same evaluation runs as the corresponding rows of Table 2. The position-weighted rows fix the schedule to Moderate $(w_{\min}, \tau, s) = (0.25, 0.30, 0.10)$.

Configuration	MATH-500	AIME 2024	AIME 2025	HMMT 2025	Average
Uniform / global (OPSD)	95.33 ± 0.08	75.19 ± 0.42	66.67 ± 1.27	43.89 ± 0.73	70.27
Uniform / per-seq	$\mathbf{95.46 \pm 0.03}$	74.72 ± 0.96	66.85 ± 1.40	$\mathbf{44.54 \pm 1.40}$	70.39
Pos-w / global	95.32 ± 0.08	74.54 ± 0.64	$\mathbf{67.87 \pm 0.98}$	44.35 ± 1.16	70.52
Pos-w / per-seq (PW-OPSD, **ours**)	95.34 ± 0.10	$\mathbf{76.20 \pm 0.58}$	67.78 ± 1.27	43.33 ± 1.21	**70.66**

Relative to OPSD, the reduction change alone raises the four-benchmark average by $+0.13$ pp, position weighting alone by $+0.25$ pp, and the joint configuration by $+0.40$ pp (computed from unrounded means), so position weighting accounts for most of the gain and the joint configuration attains the best average. The two axes interact rather than add: under a global token mean, the position weight also reweights whole sequences through their length profile, and the per-sequence mean removes this length confound. On AIME 2024 neither single axis reproduces the joint result (74.72 and 74.54 vs. 76.20), and a per-problem permutation test confirms that the joint configuration improves over position weighting with a global mean ($+1.67$ pp, $p = 0.038$).

M Additional branch-viability analyses

This section collects analyses that test the robustness of the positional finding in Section 3.2. Unless stated otherwise, each analysis uses the candidate pool of Table 1 (8 real-uncertain and 271 diversity candidates on the correct spines of 61 problems), the within-problem residualization, and the cluster bootstrap over problems ($2{,}000$ resamples) described in Appendix A.

M.1 Per-benchmark breakdown

All eight real-uncertain candidates occur on the two benchmarks with remaining headroom: four on AIME 2024 (from three problems) and four on AIME 2025 (from two problems). On MATH-500, where Qwen3-4B is close to saturation, all 113 high-entropy candidates on correct spines remain viable. Within each AIME benchmark the early-position score separates the two classes in the same direction, with residualized AUROC 0.76 $[0.53, 1.00]$ on AIME 2024 (20 problems) and 0.91 $[0.84, 0.97]$ on AIME 2025 (18 problems). Because the bootstrap resamples problems, the concentration of positives in five problems is already reflected in the pooled interval of Table 1.

M.2 Paired comparison against local scores

To compare position with each local score directly, we compute the paired AUROC difference on the same candidates and bootstrap the difference over problems (Table 7). Each baseline is scored in its a priori orientation: larger entropy, mutual information, or truncated entropy, and smaller Dirichlet precision, predict the real-uncertain label. Position exceeds every local score by 0.29–0.40 AUROC, and every interval excludes zero. Under the better-scoring orientation used in Table 1, the point differences are $+0.26$ to $+0.32$.

Table 7: Paired AUROC differences (position minus baseline) on the candidate pool of Table 1. Baselines use the a priori orientation; the position score attains 0.83. Intervals are 95% cluster-bootstrap intervals over problems.

Baseline	Baseline AUROC	Difference	95% CI
Predictive entropy $H(\mathbf{p}_t^{\mathrm{opsd}})$	0.51	$+0.318$	$[+0.138, +0.460]$
MC-dropout MI	0.51	$+0.315$	$[+0.189, +0.516]$
Dirichlet $\log \hat{\kappa}$	0.54	$+0.289$	$[+0.003, +0.594]$
Truncated entropy h_{trunc}	0.43	$+0.401$	$[+0.299, +0.574]$

M.3 Unfiltered candidate positions

The main diagnostic selects candidates by high teacher truncated entropy, which restricts the range of the local scores. To remove this restriction, we rerun the diagnostic on the same correct spines with candidate positions sampled uniformly at random (only junk, special-token, and spacing filters; no entropy ranking and no plausibility filter), keeping all other protocol elements unchanged. The resulting pool contains 8 real-uncertain and 274 diversity candidates over 61 problems; as in the filtered pool, all real-uncertain candidates occur on AIME 2024 and AIME 2025. Table 8 confirms the range restriction: once the pre-filter is removed, the local scores rise from 0.43–0.54 to 0.58–0.64. Position nevertheless remains the strongest predictor (0.78 $[0.53, 0.99]$), and its paired differences against predictive entropy ($+0.21$ $[+0.05, +0.36]$), MI ($+0.18$ $[+0.03, +0.33]$), and truncated entropy ($+0.19$ $[+0.02, +0.38]$) exclude zero.

Table 8: Residualized AUROC with and without the high-entropy pre-filter, in the a priori orientation (position is scored as $1-\widetilde{r}$). The pre-filtered column uses the candidate pool of Table 1, which instead reports each local score in its better-scoring orientation; the unfiltered column samples candidate positions uniformly on the same correct spines.

Score	Pre-filtered pool	Unfiltered pool
Predictive entropy $H(\mathbf{p}_t^{\mathrm{opsd}})$	0.51	0.58
MC-dropout MI	0.51	0.60
Dirichlet $\log \hat{\kappa}$	0.54	0.64
Truncated entropy h_{trunc}	0.43	0.59
Early-position score $1-\widetilde{r}$	$\mathbf{0.83}$	$\mathbf{0.78}$

M.4 Correct versus incorrect spines

The diagnostic is defined on correct spines because its label is measurable only there. On the 22 problems whose spines are incorrect, the same protocol labels 97 of the 101 non-gray candidates (96%) real-uncertain regardless of position, and the mean normalized position of real-uncertain candidates moves from 0.16 on correct spines to 0.41 on incorrect spines; pooling all spines lowers the position AUROC to 0.54 $[0.51, 0.58]$. When a spine never reaches the correct answer, forced continuations fail for reasons unrelated to the forced branch, so the label measures problem difficulty rather than branch viability. Training, in contrast, uses every sampled rollout without correctness-based selection (Sections 3.1 and 3.4), so the downstream results of Section 4 measure the effect of the position weight on the full rollout distribution, including incorrect and diverged rollouts.

M.5 Matched forced-token control

To separate a property of the teacher's preferred tokens from a property of the position, we change exactly one element of the diagnostic: instead of the teacher's top-3 alternatives, we force the tokens that the teacher ranks fourth to sixth, keeping the spines, the candidate-selection rule, and the viability protocol unchanged. Because teacher and student are the same checkpoint under two prompts, the control tokens come from the same model, differing only in whether the prompt contains the privileged answer. We run the control on AIME 2024 and AIME 2025, where all real-uncertain candidates occur (Appendix M.1). Table 9 shows that both token classes lose viability early and recover late with nearly identical position gradients. On the 178 positions selected in both runs, the control-minus-top-3 viability difference is $+0.015$ $[-0.026, +0.061]$ early and $+0.005$ $[-0.003, +0.014]$ late, and the difference in position gradients is -0.010 $[-0.056, +0.033]$; every interval includes zero. The early loss of viability is therefore a property of the position in the student trajectory rather than of the teacher's preference ranking, which is the trajectory-level structure that a position weight targets.

Table 9: Matched forced-token control on AIME 2024 and AIME 2025 (Qwen3-4B). Viability is the fraction of six student-context continuations that reach the correct answer; early means $\widetilde{r} < 0.35$ and late means $\widetilde{r} \geq 0.35$, near the schedule transition $\tau = 0.30$. Each row uses all positions evaluated in its own run; the paired differences in the text use the 178 positions selected in both runs.

Forced tokens	Early viability	Late viability	Early all-fail rate	Late minus early
Teacher top-3 (diagnostic)	0.728 ($n = 73$)	0.957 ($n = 122$)	0.110	$+0.230$
Teacher ranks 4–6 (control)	0.736 ($n = 73$)	0.960 ($n = 122$)	0.123	$+0.224$

M.6 A non-mathematical domain

To test whether the positional structure is specific to mathematics, we run the identical protocol on the science evaluation set released with the OPSD codebase [Zhao et al., 2026], which consists of four-way multiple-choice chemistry questions. From 108 sampled problems, 62 spines are correct, yielding 19 real-uncertain and 235 diversity candidates over 61 problems, a pool of the same scale as the mathematical one. Position is again the strongest predictor (Table 10). The four-way answer space implies a 25% guessing floor; the real-uncertain criterion $V_{\text{low}} = 0.40$ remains above this floor, although labels are noisier than in mathematics. The training schedule is calibrated on the mathematical diagnostic, and our training evaluation targets mathematical reasoning; this experiment shows that the trajectory-level structure motivating the schedule carries over to a different reasoning domain.

Table 10: Branch-viability classification on the chemistry multiple-choice set (Qwen3-4B; 19 real-uncertain vs. 235 diversity candidates over 61 problems). Entries are within-problem residualized AUROC with 95% cluster-bootstrap intervals over problems.

Score (residualized within problem)	Residual AUROC	95% CI
$H(\mathbf{p}_t^{\text{opsd}})$ (mean predictive entropy)	0.53	$[0.49, 0.59]$
MI (MC-dropout epistemic)	0.57	$[0.51, 0.65]$
$\log \hat{\kappa}$ (Dirichlet precision)	0.52	$[0.45, 0.59]$
h_{trunc} (top-16 valid entropy)	0.55	$[0.49, 0.61]$
$1 - \widetilde{r}$ (early-position score)	$\mathbf{0.62}$	$[\mathbf{0.53}, \mathbf{0.74}]$

M.7 Learned combinations of local statistics

We also ask whether position could be replaced by a richer function of local statistics learned from the diagnostic labels. Single additional statistics, including the top-2 probability gap, top-2 share, $\log(p_1/p_2)$, and child-token probability, entropy, and length statistics, reach at most 0.55 residualized AUROC even in their better-scoring orientation. We then fit logistic-regression combinations of four feature groups with leave-one-problem-out cross-validation and score the held-out predictions (Table 11). No combination approaches position: the best held-out AUROC is 0.64, and the combination of the five uncertainty scores falls below chance on held-out problems (0.16; a within-problem

label-permutation test gives $p = 0.98$ for a positive signal). Held-out values below 0.5 arise because removing a problem that contains positives lowers the training prevalence exactly when those positives are scored, so they indicate the absence of transferable signal rather than a reversed one. Besides its higher AUROC, position requires neither diagnostic labels nor the additional teacher passes that MC-dropout features need.

Table 11: Held-out AUROC of logistic-regression combinations of local statistics under leave-one-problem-out cross-validation, on the candidate pool of Table 1. Features are residualized within problem; the position row is the unfitted score of Table 1.

Feature group	Features	Held-out AUROC
Uncertainty scores	h_{trunc}, $H(\mathbf{p}_t^{\text{opsd}})$, MI, MC-dropout sample-variance trace, $\log \hat{\kappa}$	0.16
Top-2 and child probabilities	top-2 gap, $\log(p_1/p_2)$, top-2 share, top-2 entropy, entropy, gap, and mass of the three child probabilities	0.45
Child-token text	reflection-token, digit, and math-symbol indicators; mean decoded child length	0.34
All numeric statistics	every statistic above	0.64
Position (no fitting)	$1 - \widetilde{r}$	**0.83**

M.8 Calibrating the schedule from the diagnostic

Figure 3 checks whether the sigmoid schedule of Eq. (5) agrees with the position-dependent failure curve of the diagnostic. We bin all correct-spine candidates, including gray ones, into ten bins of normalized position, use the mean child viability in each bin as a continuous reliability estimate, rescale it to $[w_{\min}, 1]$, and fit Eq. (5) by count-weighted least squares over a grid ($w_{\min} \in \{0.05, \ldots, 0.60\}$, $\tau \in \{0.050, 0.075, \ldots, 0.725\}$, $s \in \{0.02, \ldots, 0.39\}$). The best fit has $\tau = 0.30$ and $s = 0.10$, which are the transition parameters of the Moderate schedule used throughout the paper (count-weighted RMSE 0.018; Moderate attains 0.033 against the same target). Because the target is rescaled to $[w_{\min}, 1]$, the fit constrains the location and sharpness of the transition but not the floor; the floor is examined by the schedule sweep in Table 3, where every floor from 0.05 to 0.50 improves the four-benchmark average over OPSD. The real-uncertain rate is 0.14, 0.12, and 0.05 in the first three bins and zero from $\widetilde{r} = 0.3$ onward.

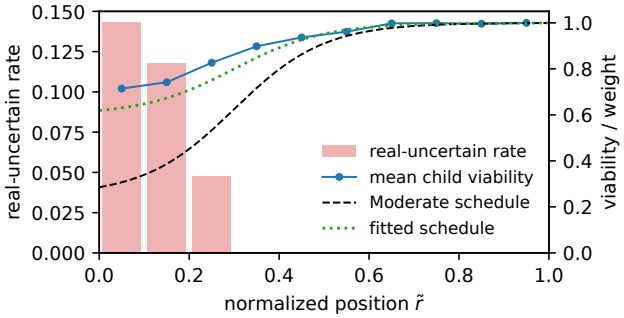


Figure 3: Real-uncertain rate (bars) and mean child viability (circles) by normalized position on correct spines, with the Moderate schedule (dashed) and the best-fitting sigmoid schedule (dotted).

N Entropy-gated position weighting

The diagnostic of Section 3.2 studies positions where the teacher is already ambiguous, whereas PW-OPSD discounts every early token. Because teacher and student share one checkpoint and differ only in the privileged prompt, the per-token forward KL is typically small where the teacher is confident, so the outer weight changes the gradient mainly at ambiguous positions. To test the

explicitly conditional design, we gate the position discount by the teacher's top-16 truncated entropy $h_{\text{trunc},t}$, the score used to select diagnostic candidates:

$$\gamma_t = \sigma\left(\frac{h_{\text{trunc},t} - \tau_H}{s_H}\right), \qquad w_{i,t}^{\text{gate}} = \gamma_t\, w_{i,t} + (1 - \gamma_t),$$

where $w_{i,t}$ is the Moderate schedule of Eq. (5), $\tau_H = 1.0$ nat, and $s_H = 0.3$; all other settings match PW-OPSD.

On Qwen3-4B (Table 12) the gated variant improves over OPSD on all four benchmarks, which indicates that the gain originates at ambiguous early positions, and across gate thresholds from 0.5 to 2.0 nats its four-benchmark average stays between 70.25 and 71.03 (OPSD: 70.27). Per problem, it is statistically indistinguishable from position-only PW-OPSD ($+0.15$ pp pooled over 590 problems; permutation $p = 0.43$). Its absolute entropy threshold, however, does not transfer across model families, whose entropy scales differ: with the same τ_H, the four-benchmark change over OPSD is -0.71 pp on DSR1-L8B and -0.03 pp on Olmo-3-7B-Think, whereas the scale-free position schedule improves all three families (Table 4). We therefore keep position-only weighting as the default proxy.

Table 12: Entropy-gated position weighting on Qwen3-4B. Avg@12 mean $\pm$ standard deviation across three evaluation seeds; the last column is the equal-weight four-benchmark mean. The lower block reports the four-benchmark mean as the gate threshold τ_H (in nats) varies with $s_H = 0.3$.

Method	MATH-500	AIME 2024	AIME 2025	HMMT 2025	Average
OPSD	95.33 ± 0.08	75.19 ± 0.42	66.67 ± 1.27	43.89 ± 0.73	70.27
PW-OPSD (position only)	95.34 ± 0.10	76.20 ± 0.58	67.78 ± 1.27	43.33 ± 1.21	70.66
Entropy-gated PW-OPSD ($\tau_H = 1.0$)	95.43 ± 0.15	76.11 ± 0.73	67.96 ± 2.10	44.63 ± 0.89	71.03
Gate threshold τ_H	0.5	1.0	1.5	2.0	
Four-benchmark average	70.85	71.03	70.25	70.42	

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 abstract and Section 1 state two claims, the branch-viability finding (positional reliability with AUROC 0.83 vs. ≤ 0.57 for local uncertainty) and the resulting PW-OPSD downstream gains; both are validated in Sections 3.2 and 4.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Section 5 explicitly discusses that PW-OPSD is plug-and-play yet its gains over OPSD remain modest, leaving room for more sophisticated methods built on the positional reliability finding.

Guidelines:

- The answer [N/A] 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: The paper does not state formal theorems, but the reliability-weighted surrogate and branch-mixture identities in Section 3.3 are accompanied by their assumptions and derivations in Appendices C and D; the downstream claims are empirical.

   Guidelines:

   - The answer [N/A] 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: Section 4.1 and Appendix I specify the training corpus, baselines, hyperparameters, decoding settings, benchmarks, and three-seed evaluation protocol; the code is available at `https://github.com/SaFo-Lab/PW-OPSD`.

Guidelines:

- The answer [N/A] 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 code release containing training and evaluation scripts plus launchers for all four reported methods is available at `https://github.com/SaFo-Lab/PW-OPSD`; all datasets used (MATH-500, AIME 2024, AIME 2025, HMMT 2025, OpenThoughts-Math-30k) are publicly available on Hugging Face and cited in Section 4.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 gives the main-body setup (models, baselines, schedule, evaluation protocol); Appendix I provides full hyperparameters, LoRA configuration, decoding parameters, and seed protocol.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: All Qwen3-4B, DSR1-L8B, and Olmo-3-7B-Think downstream tables report mean $\pm$ across-seed sample standard deviation over three evaluation seeds; the AUROC result (Section 3.2) is reported with a 95% cluster-bootstrap interval clustered at the problem level.

   Guidelines:

   - The answer [N/A] 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: Appendix I specifies the GPU type ($4\times$H100 80 GB) and effective batch size for the main training runs, and the evaluation settings (maximum generation length, sampling temperature, samples per problem) determine the per-job evaluation cost.

   Guidelines:

   - The answer [N/A] 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: The work uses only publicly available open-source models (Qwen3-4B, DeepSeek-R1-Distill-Llama-8B, Olmo-3-7B-Think) and public benchmarks; no human subjects, sensitive data, or scraped private data are involved.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: The work is foundational research on token-level distillation objectives for math reasoning models, with no direct deployment or specific application path that would create distinct societal impacts beyond those of the underlying open-source LLMs.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released artifact at `https://github.com/SaFo-Lab/PW-OPSD` contains only training and evaluation code; no new pretrained model checkpoints or high-risk datasets are released.

    Guidelines:

    - The answer [N/A] 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: All models (Qwen3-4B, DSR1-L8B, Olmo-3-7B-Think), datasets (MATH-500, AIME 2024/2025, HMMT 2025, OpenThoughts-Math-30k), and the upstream OPSD codebase are cited in Section 4.1 and Appendix I; the released code follows the licenses declared by the original asset owners on Hugging Face and GitHub.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The code release at `https://github.com/SaFo-Lab/PW-OPSD` ships with a README documenting environment setup, the four methods, training launchers, and evaluation scripts; no new datasets or model checkpoints are introduced.

    Guidelines:

    - The answer [N/A] 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 or research with human subjects.

    Guidelines:

    - The answer [N/A] 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 human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs (Qwen3-4B, DSR1-L8B, Olmo-3-7B-Think) are the subject of training and evaluation in this work, not an auxiliary component of the methodology; no separate LLM is used as a judge, annotator, or pipeline element.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.