FROM ROLLOUTS TO HINTS: SELECTIVE TRAJECTORY REUSE FOR ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) turns a reasoning model's own rollouts into dense training paths: a privileged teacher conditions on a reference solution and guides the student along prefixes the student actually visits. Yet the generated trajectory itself remains underused. The reference always occupies the teacher's privileged context, even when the student produces a verified solution that is shorter and better matched to its decoding distribution, and every generated token receives distillation despite large differences in decision relevance. We introduce SEL-OPSD, which treats generated trajectories as reusable supervision assets at two scales. HINTSELECT verifies a group of current-policy rollouts and lets a concise correct candidate serve as both the student trajectory and, when it clears a length margin, the teacher hint. KEYSTEP preserves the original OPSD divergence but restricts it to positions selected by student uncertainty, teacher–student disagreement, mathematical-token priors, and final-answer risk. Across three Qwen3 scales and three competition-level mathematics benchmarks, SEL-OPSD improves Avg@12 over dense OPSD in all nine model–benchmark settings, with a mean gain of 2.69 percentage points and a 95% bootstrap interval of $[2.41, 3.03]$. It also reduces deployment response length by approximately 30%. Factorial and matched-sparsity ablations isolate the contributions of hint reuse, verified-trajectory training, and token selection, showing that the gains cannot be explained by best-of-group filtering or sparse loss alone.

1 INTRODUCTION

Reasoning post-training is increasingly constrained by how well it uses generated trajectories. Reinforcement learning from verifiable rewards (RLVR) may sample several solutions to obtain useful reward contrast, while inference-time scaling spends compute on longer chains, repeated attempts, or verifier-ranked candidates (Shao et al., 2024; DeepSeek-AI, 2025; Wang et al., 2023). The resulting trajectories are expensive, structured records of the model's own computation. Treating them as disposable after one update leaves much of that information unused.

On-policy self-distillation (OPSD) already extracts more value from a sampled trajectory. The student generates a completion, and a privileged teacher conditions on a reference solution while scoring the same visited prefixes (Thinking Machines Lab, 2025; Zhao et al., 2026). This preserves the student's on-policy support while replacing a single outcome reward with dense token-level guidance, avoiding the exposure mismatch of teacher-forced distillation (Hinton et al., 2015; Agarwal et al., 2024; Gu et al., 2024). But dense OPSD still treats trajectory reuse as a fixed recipe: the reference is always the privileged context, and the loss is applied uniformly across the sampled sequence.

Figure 1 exposes the two allocation decisions hidden by this recipe. First, a generated candidate can contain a better matched privileged context than the static reference. The reference is reliable when the current model rarely solves a problem, but once a candidate is both answer-verified and concise, its prefixes and vocabulary are closer to the distribution being optimized. The decision is therefore not whether to discard the reference, but when a generated candidate has earned the right to replace it. Second, a reasoning trace is not homogeneous. Algebraic commitments, branch points, recovery steps, and the final answer coexist with predictable connective text. Uniform distillation allocates the same objective support to all of them, even though uncertainty and teacher–student disagreement are sharply concentrated.

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

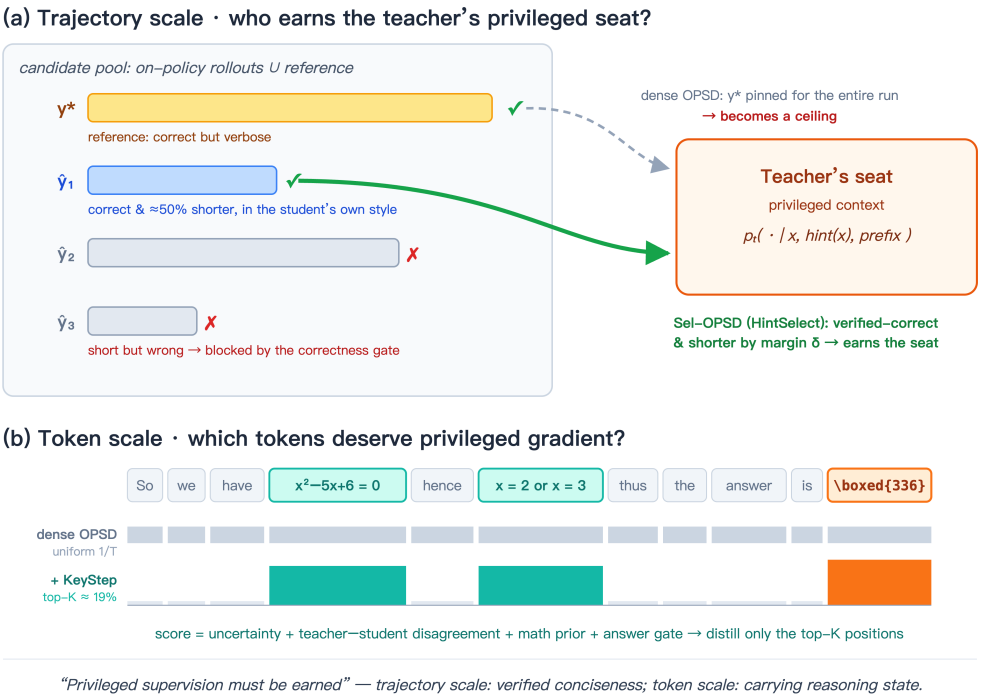


Figure 1: **Generated trajectories contain supervision at two scales.** Dense OPSD conditions the teacher on a fixed reference and supervises every generated position. SEL-OPSD re-evaluates candidate trajectories before reusing them as teacher hints and student paths, then concentrates the unchanged divergence on selected token positions.

We propose SEL-OPSD, a two-scale framework for selective trajectory reuse. At the trajectory scale, HINTSELECT samples a group of current-policy candidates and verifies their final answers. The shortest correct candidate becomes the student-side trajectory; it also replaces the reference as the teacher hint only when it improves the reference length by a fixed margin. When no candidate is verified, the method falls back exactly to the dense OPSD teacher context. At the token scale, KEYSTEP keeps the same teacher and divergence, but limits the loss support to positions ranked by student uncertainty, teacher–student disagreement, a lightweight mathematical-token prior, and answer-region risk. The two stages modify neither the base model nor the self-distillation objective. They change what the generated trajectories are allowed to supervise.

This design creates an attribution problem that a single headline table cannot resolve. Selecting the best verified trajectory can improve training even if that trajectory never becomes the teacher hint, and reducing the number of supervised tokens can act as regularization even if the selected tokens carry no special information. We therefore separate hint replacement from verified-trajectory training in a 2×2 factorial experiment, compare KEYSTEP with matched random sparsification and single-signal selectors, and measure the added rollout cost against deployment-time response length. The resulting evidence is consistent across model scales and task difficulty: SEL-OPSD improves Avg@12 in all nine Qwen3 model–benchmark cells, the mean gain over dense OPSD is 2.69 percentage points, and responses are approximately 30% shorter at deployment.

Contributions. We make three contributions. (i) We formulate on-policy self-distillation as supervision allocation over generated trajectories, separating trajectory selection, privileged-context selection, and token-level support. (ii) We instantiate this formulation in SEL-OPSD, whose verifier-gated HINTSELECT and score-based KEYSTEP modules reduce to dense OPSD when their selectors are disabled. (iii) We establish the effect through matched comparisons across three model scales and six mathematics benchmarks, factorial and matched-sparsity ablations, seed-level variation, and a training–deployment cost analysis.

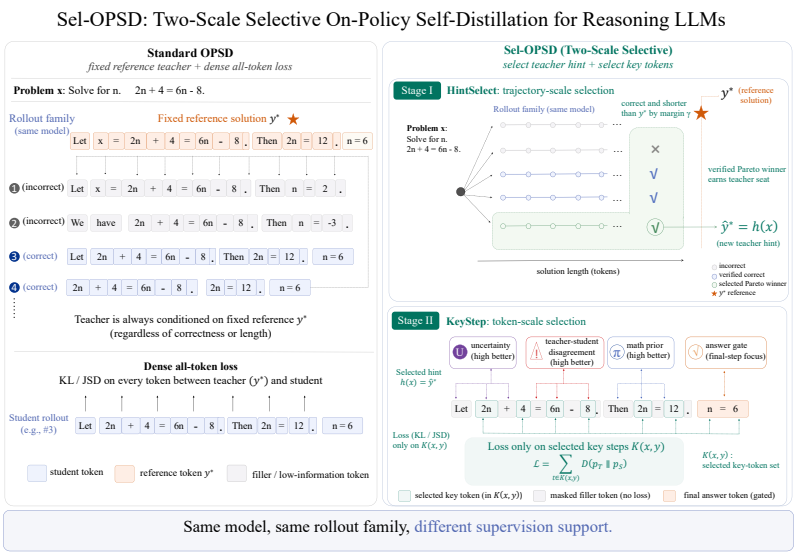


Figure 2: **SEL-OPSD overview.** Stage I (HINTSELECT) lets a verified shorter rollout replace the reference as the teacher hint; Stage II (KEYSTEP) distills only on high-value tokens. The base teacher, student, prompt, and loss are unchanged.

2 RELATED WORK

On-policy distillation. GKD and MiniLLM train on student-generated sequences to reduce exposure mismatch (Agarwal et al., 2024; Gu et al., 2024), and recent reasoning pipelines let the teacher score student-visited prefixes under richer context (Thinking Machines Lab, 2025; Zhao et al., 2026; Ye et al., 2026), with concurrent analyses tracing the gain to token-level overlap and thinking-pattern alignment (Li et al., 2026). SEL-OPSD keeps this on-policy support and the same divergence family but changes *which* trajectory conditions the teacher and *which* token gradients survive. Concurrent variants extend OPSD along orthogonal axes—conciseness-conditioned teachers (Sang et al., 2026), dual-path calibration (Zheng et al., 2026), verifier-reward coupling (Yang et al., 2026), multi-turn agents (Wang et al., 2026), and student-in-the-loop rationale choice (He et al., 2026)—complementary to our factorization.

Self-training, filtering, and reasoning supervision. STaR, ReST(-EM), RFT, SPIN, and expert iteration realize generate–filter–retrain loops over *complete* sequences (Zelikman et al., 2022; Gulcehre et al., 2023; Singh et al., 2024; Yuan et al., 2024; Chen et al., 2024; Anthony et al., 2017); we reuse correctness-gated filtering but move the chosen rollout to the *teacher* side, keeping supervision token-level. Math offers symbolic verification (Cobbe et al., 2021; Lightman et al., 2024; Wang et al., 2024), on top of which RLVR methods such as PPO and GRPO optimize sparse outcome rewards (Schulman et al., 2017; Shao et al., 2024). SEL-OPSD sits between the two: verification gates which trajectories may teach, while token-level distributions supply local signal.

Token-level selectivity. Tokens contribute unequally to reasoning gains (Lin et al., 2024a; Wang et al., 2025; Lin et al., 2024b); KEYSTEP ports this into privileged on-policy distillation by masking low-value positions inside an existing OPSD trainer, without changing the loss, model, or verifier.

3 BACKGROUND: DENSE OPSD

Let x be a problem, $y^\star$ its reference solution, and $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T)$ a completion sampled from the current student $p_S(\cdot \mid x)$. In dense OPSD, the teacher scores the same visited prefixes while

conditioning on privileged information:

$$\mathcal{L}_{\mathrm{OPSD}} = \mathbb{E}_{x,\hat{y}\sim p_S}\Bigg[\frac{1}{T}\sum_{t=1}^{T} \mathcal{D}\big(p_T(\cdot \mid x, y^\star, \hat{y}_{<t}) \,\big\|\, p_S(\cdot \mid x, \hat{y}_{<t})\big)\Bigg], \tag{1}$$

where $\mathcal{D}$ may be forward KL, reverse KL, JSD, or a clipped token-level surrogate. In self-distillation, p_T shares parameters with p_S and differs only in its privileged conditioning: the "teacher" is the student evaluated under a richer context. Equation 1 is dense in two senses. The teacher hint is fixed to the reference $y^\star$, and every generated position receives equal averaging weight. SEL-OPSD relaxes both assumptions.

4 METHOD: TWO-SCALE SELECTIVE DISTILLATION

4.1 OVERVIEW

A training step of SEL-OPSD proceeds in three blocks. We first sample G on-policy completions from the student for each problem. Given these rollouts, a verifier and a length-margin rule pick the privileged hint $h(x)$ and the distillation trajectory $\tilde{y}$. Each generated token in $\tilde{y}$ is then scored, the top positions form a small set $\mathcal{K}$, and the unchanged OPSD divergence is applied only on $\mathcal{K}$. The resulting objective is

$$\mathcal{L}_{\mathrm{Sel}} = \mathbb{E}_{x,\tilde{y}}\Bigg[\frac{1}{|\mathcal{K}|}\sum_{t\in\mathcal{K}(x,\tilde{y})} \mathcal{D}\big(p_T(\cdot \mid x, h(x), \tilde{y}_{<t}) \,\big\|\, p_S(\cdot \mid x, \tilde{y}_{<t})\big)\Bigg]. \tag{2}$$

If the trajectory selector does not replace $y^\star$ and $\mathcal{K} = \{1, \ldots, T\}$, Eq. 2 reduces to dense OPSD.

4.2 STAGE I: CORRECTNESS-GATED HINT SELECTION

The first stage decides whether the reference solution $y^\star$ continues to occupy the teacher's privileged-context slot, or whether a verified shorter rollout takes its place. The rule is lexicographic—correctness first, then length: among verifier-correct candidates we take the shortest, and it replaces $y^\star$ only when it wins by a margin. The reference $y^\star$ is merely one correct solution among many; during training, the student may produce correct rollouts that are shorter or closer to its decoding distribution.

For each prompt x, the trainer samples a group of on-policy rollouts $\hat{Y} = \{\hat{y}_i\}_{i=1}^{G}$ and forms the candidate pool $\mathcal{C} = \hat{Y} \cup \{y^\star\}$. A strict verifier defines correctness:

$$r_1(y) = \mathbf{1}[\mathrm{ans}(y) \equiv \mathrm{ans}(y^\star)]. \tag{3}$$

The answer extractor prefers the most recent $\texttt{\textbackslash boxed}\{\cdot\}$, falls back to the last numeric token, and compares strings after light normalization (optionally symbolically); the verifier is conservative.

Among the correct candidates $\mathcal{C}_+ = \{y \in \mathcal{C} : r_1(y) = 1\}$, the trainer selects the shortest candidate

$$\hat{y}^\star = \arg\min_{y\in\mathcal{C}_+} |y|, \tag{4}$$

measured in tokenized form so the comparison is robust to surface formatting. The shortest correct rollout dethrones the reference only if it wins by a relative margin:

$$h(x) = \begin{cases} \hat{y}^\star, & |\hat{y}^\star| < (1-\delta)|y^\star|, \\ y^\star, & \text{otherwise}, \end{cases} \tag{5}$$

with $\delta = 0.05$ by default; the deliberately conservative margin suppresses near-tie switching (which would inject teacher non-stationarity) and avoids replacing an already concise reference. A correct rollout, when present, also becomes the student-side trajectory $\tilde{y}$; if the whole group is incorrect the trainer falls back to dense OPSD on the first sampled rollout, so the hint is never less informative than under the dense baseline.

Two coupled switches. Relative to dense OPSD, Stage I thus changes two things at once: a *hint switch* replaces the teacher-side context ($y^\star \to \hat{y}^\star$ upon dethroning), and a *trajectory switch* replaces the student-side trajectory ($\hat{y}_1 \to \hat{y}^\star$ whenever a correct rollout exists). The switches are separable—either can be applied alone—and Tables 2 and 3 report the full 2×2 factorial, so the teacher-seat claim is not confounded with the data-quality benefit of training on verified-correct rollouts.

By construction, only candidates in $\mathcal{C}_+$ may dethrone $y^\star$, so every selected hint is verifier-correct: conciseness is rewarded only after correctness, which rules out the short-but-wrong failure mode of naive length penalties.

Group size G trades rollout compute against the chance of finding a verified shorter winner; $G \in \{4, 8\}$ surfaces non-trivial replacement rates at acceptable per-step cost (Table 8), the replacement rate is logged for drift inspection, and fixed- or EMA-teacher stabilizers remain orthogonal.

4.3 STAGE II: KEY-STEP TOKEN SELECTION

The second stage decides which positions inside the selected trajectory $\tilde{y}$ contribute to the distillation loss. Long reasoning traces mix a few algebraic commitments and final-answer decisions with many stylistic connectives such as "so we have"; dense OPSD spends equal gradient on both, and divergence on the second kind can drown out the first. KEYSTEP scores each generated position and applies the divergence loss only on the top-scoring subset (Figure 3).

Given $\tilde{y}$, the student and teacher produce aligned logits z_t^S and z_t^T at each generated position t. The per-token score combines four signals:

$$s_t = \lambda_u U_t + \lambda_d D_t + \lambda_m M_t + A_t, \tag{6}$$

where U_t is student uncertainty, D_t teacher–student disagreement, M_t a math-token prior, and A_t an answer-aware bonus, with default weights $\lambda_u = \lambda_d = 1$ and $\lambda_m = 2$. The continuous signals U_t and D_t are standardized per trajectory (zero mean, unit variance over generated positions) before weighting, so the four terms combine on comparable scales and no raw logit magnitude dominates the score. The weights are set a priori rather than tuned: because U_t and D_t are standardized, equal unit weight keeps "how undecided the student is" and "how much the hint would change that decision" on par, and the binary prior carries $\lambda_m = 2$ so that a strong lexical cue can lift a token past a one-SD swing in either continuous signal. We validate the *combination* rather than the specific weights: the signal ablation (Table 3, bottom) shows each component contributes, while exact weight values sit inside one seed-level SD.

Student uncertainty. Let $z^S_{t,(1)}$ and $z^S_{t,(2)}$ be the largest and second-largest student logits. We define

$$U_t = -(z^S_{t,(1)} - z^S_{t,(2)}), \tag{7}$$

so tokens with smaller logit margins—branch points, variable substitutions, equation-direction choices—receive larger scores.

Teacher–student disagreement. Let $p_t^S = \mathrm{softmax}(z_t^S)$ and $p_t^T = \mathrm{softmax}(z_t^T)$. We use a symmetric KL,

$$D_t = \frac{1}{2}\left[\mathrm{KL}(p_t^T \| p_t^S) + \mathrm{KL}(p_t^S \| p_t^T)\right], \tag{8}$$

which is robust when one side places near-zero mass where the other is confident. D_t fires where the privileged hint shifts the local continuation distribution and complements U_t: uncertainty measures how undecided the student is alone, disagreement measures how much the hint would change that decision (Table 3).

Math prior. M_t is a lightweight binary indicator that fires on digits, fractions, single-letter variables, operators, equality and inequality signs, brackets, equation-environment markers, and final-answer wrappers such as `\boxed`. The indicator is computed from token surface form without a parser, and suppresses stylistic phrases that score high on the continuous signals but carry no algebraic state.

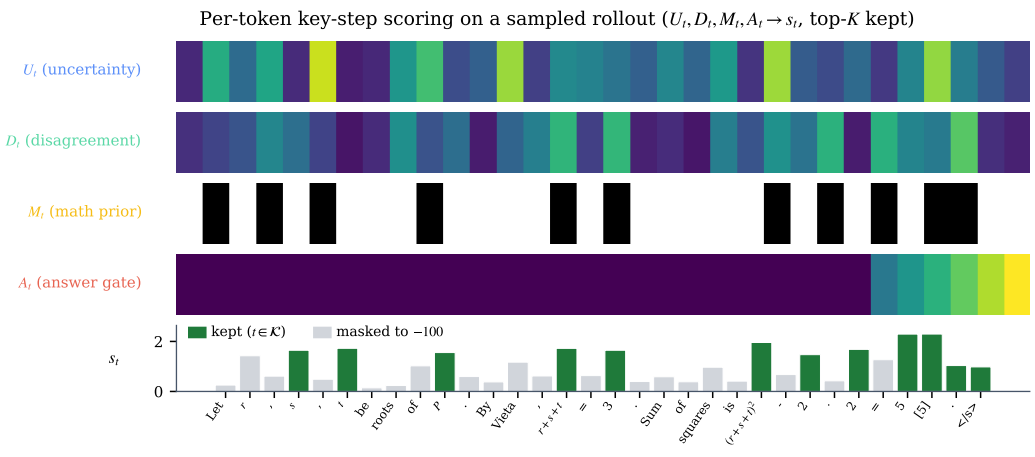


Figure 3: **Token-scale scoring signals.** KEYSTEP combines uncertainty, disagreement, a math prior, and answer-aware gating; the top-scoring positions concentrate on algebra, branch corrections, and final-answer commitments.

Answer-aware gate. When the sampled final answer disagrees with the reference, the last w generated tokens receive an additive bonus $A_t > 0$ covering the region where the model commits to a numeric answer (including the closing `\boxed` expression). The bonus raises the priority of the final commitment region exactly when the rollout is wrong; on correct rollouts, $A_t = 0$.

Budgeted selection. For each sample, the trainer keeps a fixed budget of the top-K positions by s_t, with three safeguards: a minimum-keep floor $K_{\min}$ for very short completions, a mandatory tail window K_{tail} so the answer region remains represented, and unconditional masking of prompt positions. The default configuration uses $K = 128$, $K_{\min} = 32$, $K_{\text{tail}} = 24$, answer bonus $A^* = 3.0$, and answer window $w = 48$; because our training rollouts average $\approx$0.7k generated tokens, this fixed $K = 128$ budget corresponds to keeping $\approx$18–19% of generated positions on average. Non-selected labels are set to -100 so the original OPSD loss ignores them unchanged.

Cost and composition with Stage I. KEYSTEP reuses the student and teacher forward passes from dense OPSD and adds an $O(T)$ score over materialized logits. The same scoring rule applies to selected and fallback trajectories, with no stage-specific parameters.

5 EXPERIMENTS

5.1 SETUP

Models and training. We evaluate Qwen3-1.7B/4B/8B (Yang et al., 2025) on the same OPSD math split with LoRA (Hu et al., 2022), grouped vLLM rollouts (Kwon et al., 2023), and a TRL-style on-policy trainer (von Werra et al., 2020); the student prompt holds only the problem, while the teacher additionally sees the selected hint on the same visited prefix. We train for 100 steps (LoRA rank/alpha 64/128, lr 8×10^{-6}, batch 4, grad. accum. 2, rollout cap 16,384 vs. a 38,912 evaluation budget) with three seeds per configuration; full generation, hardware, and fallback details are in the supplement. The method-specific hyperparameters (G, δ, K, w) are defaults inherited from the matched dense-OPSD configuration or set a priori, held constant across all scales and benchmarks, and never tuned on any evaluation benchmark; Appendix Table 6 documents the post-hoc sensitivity protocol.

Benchmarks and metrics. On AIME24, AIME25, and HMMT25 we sample $n = 12$ completions per problem and report Avg@12 (mean correctness), Pass@12 (solved at least once), and Maj@12 (majority vote under self-consistency (Wang et al., 2023)).

Table 1: Main results across three Qwen3 scales; each cell is Avg@12 / Pass@12 / Maj@12 over 30 problems. SFT, GRPO, and OPSD reproduce Zhao et al. (2026) under our evaluator; Pass/Maj lie on a $k/30$ grid, so close methods can tie.

Dataset	Method	Qwen3-1.7B			Qwen3-4B			Qwen3-8B		
		Avg	Pass	Maj	Avg	Pass	Maj	Avg	Pass	Maj
AIME24	SFT on $y^\star$	48.33	70.00	53.33	70.28	83.33	70.00	72.22	86.67	73.33
	GRPO	51.11	73.33	56.67	75.56	86.67	73.33	76.39	90.00	80.00
	OPSD	57.22	76.67	60.00	76.39	90.00	76.67	77.78	90.00	80.00
	HINTSELECT only	58.33	80.00	63.33	77.78	90.00	80.00	79.17	93.33	83.33
	SEL-OPSD	**59.44**	**80.00**	**66.67**	**78.89**	**93.33**	**80.00**	**80.28**	**93.33**	**86.67**
AIME25	SFT on $y^\star$	36.39	60.00	46.67	62.22	73.33	66.67	64.17	80.00	70.00
	GRPO	38.33	63.33	50.00	68.06	80.00	70.00	68.89	83.33	76.67
	OPSD	43.89	66.67	53.33	68.33	80.00	73.33	70.83	86.67	80.00
	HINTSELECT only	45.28	70.00	56.67	70.00	83.33	73.33	72.22	86.67	80.00
	SEL-OPSD	**46.39**	**73.33**	**60.00**	**70.83**	**86.67**	**76.67**	**73.06**	**90.00**	**83.33**
HMMT25	SFT on $y^\star$	22.78	46.67	23.33	43.33	66.67	43.33	42.78	66.67	56.67
	GRPO	23.61	50.00	26.67	44.44	70.00	46.67	46.67	73.33	60.00
	OPSD	29.17	56.67	30.00	46.11	73.33	46.67	45.83	73.33	60.00
	HINTSELECT only	30.83	60.00	33.33	48.06	73.33	50.00	48.61	76.67	63.33
	SEL-OPSD	**31.94**	**63.33**	**36.67**	**49.17**	**76.67**	**53.33**	**49.72**	**80.00**	**66.67**

Baselines. We compare SEL-OPSD against three families: *(i) self-training on verified rollouts*—SFT on $y^\star$, STaR (Zelikman et al., 2022), RFT (Yuan et al., 2024), ReST-EM (Singh et al., 2024); *(ii) verifier-rewarded RL*—GRPO (Shao et al., 2024), DeepSeek-R1 (DeepSeek-AI, 2025), and DAPO (Yu et al., 2025); these systems are not training-compute matched to our distillation runs, so we report GRPO as a reference row and cite DeepSeek-R1 and DAPO for context only; *(iii) on-policy distillation*—MiniLLM (Gu et al., 2024), GKD (Agarwal et al., 2024), the Thinking Machines Lab recipe (Thinking Machines Lab, 2025), OPCD (Ye et al., 2026), and dense OPSD (Zhao et al., 2026), with concurrent analyses (Li et al., 2026). Only dense OPSD, HINTSELECT-only, KEYSTEP-only, and full SEL-OPSD are training-matched—one shared teacher–student rollout pair, differing solely in the two selection modules; SFT/GRPO/OPSD reproduce Zhao et al. (2026) under our evaluator, and the other baselines use our prompt, sampling, and verifier with their own recipe.

5.2 MAIN RESULTS

Relative to dense OPSD, SEL-OPSD raises Avg@12 by 2.2–3.9 pp across the nine cells—HINTSELECT contributing most and KEYSTEP a steady 0.8–1.1 pp on top—with the largest gains on HMMT25 and at 1.7B, where verified-shorter rollouts are common and filler most dilutes dense supervision. Our reproduced OPSD tracks Zhao et al. (2026) (including its published dip below GRPO on HMMT25-8B); SEL-OPSD exceeds both in every cell, never lowers Pass@12 or Maj@12 (ties reflect the $k/30$ grid), and leads every baseline in Table 2 under fixed backbone, prompt, and evaluator.

5.3 ABLATIONS

Table 3 (top) decomposes HINTSELECT: the hint switch alone adds $+0.55$ Avg, the trajectory switch $+0.83$ Avg / $+3.33$ Maj, and they combine near-additively ($+1.39$/$+3.33$), so the trajectory gain is not merely the data-quality effect of training on verified rollouts. KEYSTEP adds a further $+1.11$ Avg / $+3.34$ Maj. On the token side (bottom), random sparsification at the same keep ratio underperforms dense OPSD, and the hybrid-plus-answer-gate score beats every single-signal variant in all three seeds; single-signal rows differ within one seed-level SD (± 0.4 pp), so we do not rank them.

Table 2: Cross-method comparison on Qwen3-8B. Hint: dynamic teacher hint; Traj.: verified-trajectory substitution (sequence-level filtering for RFT/ReST-EM); Mask: token-level masking. HINTSELECT$_{\text{hint}}$/HINTSELECT$_{\text{traj}}$ apply one switch each, completing the 2×2 factorial with OPSD and HINTSELECT. SFT/GRPO/OPSD reproduce Zhao et al. (2026); other baselines use our prompt, sampling, and verifier.

Method	Hint	Traj.	Mask	AIME24			AIME25			HMMT25		
				Avg	Pass	Maj	Avg	Pass	Maj	Avg	Pass	Maj
SFT on $y^\star$	×	×	×	72.22	86.67	73.33	64.17	80.00	70.00	42.78	66.67	56.67
RFT	×	✓	×	73.61	86.67	76.67	65.83	80.00	73.33	43.89	70.00	56.67
ReST-EM	×	✓	×	74.44	86.67	76.67	66.94	83.33	73.33	44.72	70.00	60.00
MiniLLM	×	×	×	75.28	90.00	76.67	67.50	83.33	73.33	45.28	73.33	56.67
GKD	×	×	×	75.83	90.00	80.00	68.06	83.33	76.67	45.00	73.33	60.00
GRPO	×	×	×	76.39	90.00	80.00	68.89	83.33	76.67	46.67	73.33	60.00
OPSD	×	×	×	77.78	90.00	80.00	70.83	86.67	80.00	45.83	73.33	60.00
HINTSELECT$_{\text{hint}}$	✓	×	×	78.33	90.00	80.00	71.39	86.67	80.00	47.22	73.33	60.00
HINTSELECT$_{\text{traj}}$	×	✓	×	78.61	90.00	83.33	71.67	86.67	80.00	47.50	73.33	63.33
HINTSELECT only	✓	✓	×	79.17	93.33	83.33	72.22	86.67	80.00	48.61	76.67	63.33
KEYSTEP only	×	×	✓	78.61	90.00	83.33	71.39	86.67	80.00	47.78	73.33	63.33
SEL-OPSD	✓	✓	✓	**80.28**	**93.33**	**86.67**	**73.06**	**90.00**	**83.33**	**49.72**	**80.00**	**66.67**

Table 3: Ablations on Qwen3-8B AIME24. **Top:** 2×2 factorial over the hint and trajectory switches, plus token masking. **Bottom:** KEYSTEP signal ablation.

Variant	Hint	Traj.	Mask	Avg	Maj
OPSD dense	×	×	×	77.78	80.00
HINTSELECT$_{\text{hint}}$	✓	×	×	78.33	80.00
HINTSELECT$_{\text{traj}}$	×	✓	×	78.61	83.33
HINTSELECT only	✓	✓	×	79.17	83.33
KEYSTEP only	×	×	✓	78.61	83.33
SEL-OPSD	✓	✓	✓	**80.28**	**86.67**

Variant	Kept tok.	Avg	Pass	Maj
Dense OPSD	100%	77.78	90.00	80.00
Random mask	18%	77.22	86.67	80.00
Uncertainty only	18%	78.06	90.00	80.00
Disagreement only	18%	78.33	90.00	83.33
Hybrid	18%	79.17	90.00	83.33
Hybrid + answer gate	19%	**80.28**	**93.33**	**86.67**

Table 4: Generalization across three Qwen3 scales, Avg@n (GSM8K/MATH-500 (Cobbe et al., 2021; Hendrycks et al., 2021) use $n = 4$; OlympiadBench (He et al., 2024) uses $n = 12$). Pass/Maj: Tables 1, 2.

Dataset	Method	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
GSM8K	OPSD	92.40	93.80	94.41
	SEL-OPSD	**93.10**	**94.50**	**94.64**
MATH-500	OPSD	88.70	91.20	93.50
	SEL-OPSD	**90.10**	**92.60**	**94.50**
OlympiadBench	OPSD	43.20	48.10	69.51
	SEL-OPSD	**44.80**	**49.70**	**71.03**

5.4 GENERALIZATION AND INFERENCE EFFICIENCY

Beyond the three competition sets, gains grow with difficulty—modest on GSM8K, larger on MATH-500, largest on OlympiadBench (Table 4). The compute trade has two sides: SEL-OPSD cuts mean response length by $\approx 30\%$ on all three benchmarks (Table 5), while grouped rollouts raise step time from 15.5 to 45.1 s and KEYSTEP itself adds little overhead; Appendix E reports the full breakdown and break-even analysis.

Table 5: Deployment response length on Qwen3-8B (n=12); Tok./correct = total generated tokens / correct samples.

Benchmark	Method	Mean tok.	Tok./correct
AIME24	OPSD	11,540	14,839
	SEL-OPSD	**7,980**	**9,941**
AIME25	OPSD	12,210	17,238
	SEL-OPSD	**8,630**	**11,812**
HMMT25	OPSD	13,620	29,721
	SEL-OPSD	**9,480**	**19,067**

6 LIMITATIONS

Our testbed is competition math with a cheap symbolic verifier (matched to Zhao et al. (2026)): a false positive can promote a concise but wrong hint, and the math-token prior needs a domain-specific replacement elsewhere. We cover up to Qwen3-8B with LoRA under a 16,384-token rollout cap; full-parameter tuning and learned-verifier domains are future work.

7 CONCLUSION

Generated reasoning trajectories are more than transient paths to a terminal reward. SEL-OPSD turns them into supervision assets: HINTSELECT decides when a verified candidate should become the student path and privileged teacher hint, while KEYSTEP decides which positions merit dense guidance. Across three Qwen3 scales and three competition benchmarks, both decisions contribute independently and together, improving Avg@12 in every model–benchmark cell while shortening deployment responses by approximately 30%. The broader design principle is that self-distillation should allocate privileged supervision explicitly—over trajectories, contexts, and tokens—instead of treating a sampled sequence as an indivisible training example.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript restructuring, English drafting and editing, and LaTeX formatting. They were not used to generate experimental data or execute experiments. All numerical results and technical claims in the AI-assisted text were drawn from the author-supplied manuscript, tables, and supplementary records. The authors take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies training efficiency and accuracy on mathematical benchmarks and does not involve human subjects or personal data. Improved mathematical reasoning should not be interpreted as evidence of truthfulness or safety in open-domain deployment. In domains without exact verification, a noisy verifier may promote concise but incorrect trajectories; such use requires domain-specific validation and monitoring.

REPRODUCIBILITY STATEMENT

Section 5 specifies the evaluation protocol and matched baselines. Appendix A reports training, decoding, hardware, software, and seed configurations; Appendix B provides implementation pseudocode; and Appendix D gives seed-level variation and statistical analyses. The anonymous supplementary materials submitted through OpenReview include resolved configurations, per-seed predictions, and scripts for reproducing tables and figures.

REFERENCES

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

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning (ICML)*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645(8081):633–638, 2025. arXiv:2501.12948.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, et al. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Chaoqun He, Yingfa Chen, Chaojun Xiao, Xu Han, and Lijie Wen. Student-in-the-loop chain-of-thought distillation via generation-time selection. *arXiv preprint arXiv:2604.02819*, 2026. URL https://arxiv.org/abs/2604.02819.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

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

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 (ICLR)*, 2022.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016.

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 (ICLR)*, 2024.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*, 2024a.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances LLM's reasoning capability. *arXiv preprint arXiv:2411.19943*, 2024b. URL https://arxiv.org/abs/2411.19943.

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. URL https://arxiv.org/abs/2603.05433.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, K. Li, Y. Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. arXiv:2312.06585.

Thinking Machines Lab. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation, 2025. Technical report, Thinking Machines Lab.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouedec. TRL: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv preprint arXiv:2604.10674*, 2026. URL https://arxiv.org/abs/2604.10674.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2024. Published in ACL 2024.

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

Xuezhi Wang, Jason Wei, Dale 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 (ICLR)*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3775.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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. URL https://arxiv.org/abs/2601.18734.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026. URL https://arxiv.org/abs/2604.10688.

A Reproducibility Details

Training configuration. All runs use Qwen3 base models (1.7B, 4B, and 8B) (Yang et al., 2025) with LoRA rank 64 and alpha 128 on the standard attention and MLP projections (Hu et al., 2022). We train for 100 steps with learning rate 8×10^{-6}, gradient clipping at 0.1, gradient checkpointing, gradient accumulation of 2, and per-device batch size 4. The loss uses $\beta = 0$, $\lambda = 1$, and JSD token clipping at 0.05. Student rollouts use temperature 1.05, top-p 0.95, top-k 20, and a training rollout cap of 16,384 tokens (evaluation uses a 38,912-token budget). Training-time generation runs through vLLM in colocated mode (Kwon et al., 2023).

Selection and evaluation. HintSelect uses group size $G = 8$ and length margin $\delta = 0.05$. KeyStep uses $K = 128$, $K_{\min} = 32$, $K_{\text{tail}} = 24$, uncertainty and disagreement weights $\lambda_u = \lambda_d = 1.0$, math bonus $\lambda_m = 2.0$, answer bonus $A^* = 3.0$, and answer window $w = 48$. Evaluation uses temperature 1.0, top-p 0.95, maximum new-token budget 38,912, and $n = 12$ samples per problem on the three primary benchmarks. GSM8K and MATH-500 use $n = 4$ because of their larger test sets.

Hyperparameter development protocol. The four method-specific hyperparameters (G, δ, K, w) are set to defaults rather than tuned on any evaluation benchmark. Optimizer, LoRA, decoding, and loss parameters are inherited from the matched dense-OPSD configuration, and $G = 8$, $\delta = 0.05$, $K = 128$, $w = 48$, together with $K_{\min} = 32$, $K_{\text{tail}} = 24$, $\lambda_u = \lambda_d = 1$, and $\lambda_m = 2$, are fixed a priori and held constant across all model scales and benchmarks. Table 6 lists, for transparency, the candidate values we examined in *post-hoc* sensitivity sweeps on Qwen3-8B AIME24 (three seeds each); these sweeps characterize robustness around the defaults and are not a model-selection procedure.

Consequently no evaluation benchmark—AIME24 included—is used to select hyperparameters. All model scales and benchmarks (the 1.7B, 4B, and 8B scales; AIME24, AIME25, HMMT25; and the generalization suite GSM8K, MATH-500, OlympiadBench) are evaluated once under this single fixed configuration.

Implementation and fallback behavior. The data collator retains raw problem and reference-solution strings alongside tokenized prompts. For each prompt, HintSelect verifies the sampled group and selects the shortest correct rollout. If no rollout is correct, the method uses the original reference hint and first sampled trajectory, recovering the dense OPSD teacher-seat behavior. KeyStep writes -100 to non-selected generated-token labels, so the existing token-level divergence ignores them. Setting the generated-token mask to one everywhere recovers dense token supervision.

Table 6: Method-specific hyperparameters: the fixed default (used across all scales and benchmarks) and the candidate values examined in post-hoc sensitivity sweeps on Qwen3-8B AIME24. These sweeps characterize robustness and are not used for model selection.

Parameter	Candidate values	Default	# tried	Sensitivity axis
Group size G	$\{1, 2, 4, 8, 16\}$	8	5	accuracy–rollout cost
Length margin δ	$\{0, .05, .10, .20, \infty\}$	.05	5	correct-shorter replacement
Token budget K	$\{32, 64, 128, 256, \text{dense}\}$	128	5	accuracy–keep ratio
Answer window w	$\{0, 16, 32, 48, 64, 96\}$	48	6	final-answer coverage

Algorithm 1 Augmented data collator. The main additional operation compared with dense OPSD is retaining raw strings for re-prompting and verification.

Require: Batch B with fields PROBLEM and SOLUTION.
Ensure: Trainer batch $\mathcal{B}$.
1: $\mathcal{B}_{\text{stu}} \leftarrow$ TOKENIZEPROMPTONLY($\{f.\text{PROBLEM} : f \in B\}$)
2: $\mathcal{B}_{\text{tch}} \leftarrow$ TOKENIZEPROMPTWITHHINT($\{(f.\text{PROBLEM}, f.\text{SOLUTION}) : f \in B\}$)
3: $\mathcal{B} \leftarrow \text{merge}(\mathcal{B}_{\text{stu}}, \mathcal{B}_{\text{tch}})$
4: $\mathcal{B}.\text{PROBLEMTEXTS} \leftarrow [f.\text{PROBLEM} : f \in B]$
5: $\mathcal{B}.\text{SOLUTIONTEXTS} \leftarrow [f.\text{SOLUTION} : f \in B]$
6: **return** $\mathcal{B}$

Code availability. The anonymous supplementary materials submitted through OpenReview include implementations for dataset parsing, answer normalization, training, rollout generation, both selection modules, evaluation, statistical analysis, and figure/table generation. The experiments do not use a separate private preprocessing pipeline. The new-method functions are commented with references to the corresponding equations and Algorithms 2–4. Upon publication, the code, configuration files, and reproduction commands will be released under the MIT license.

Randomness and deterministic setup. We use three training seeds, $\{42, 43, 44\}$, for measuring run-to-run variation. For seed s, the launcher initializes the process-level, Python, NumPy, PyTorch CPU, and all CUDA random-number generators to s. Data-loader shuffling uses a seeded `torch.Generator`; worker seeds are initialized from its PyTorch worker seed. We enable deterministic settings where supported and explicitly control major randomness sources. To improve generation reproducibility, every vLLM request receives an explicit seed: training requests use $42 + 10^7 r + 10^3 q + j$ and evaluation requests use $10^8 + 42 + 10^7 r + 10^3 q + j$, where $r \in \{0, 1, 2\}$ is the training-run index, q is the zero-based problem index, and j is the rollout index. The archive stores the resolved configuration and seed map for each run.

Hardware and software. Our experiments are conducted on one Ubuntu 22.04.4 LTS node with eight NVIDIA A800-SXM4 GPUs (80 GB each; 640 GB total GPU memory), two AMD EPYC 7543 CPUs (64 physical cores total), and 512 GB system RAM. The software environment is Python 3.10.14, NVIDIA driver 550.54.15, CUDA 12.4, PyTorch 2.6.0, Transformers 4.51.3, PEFT 0.15.2, vLLM 0.8.5, and TRL 0.17.0. Training uses bfloat16 model computation and float32 optimizer states. The OpenReview supplementary materials include a pinned environment file and the corresponding launch commands for training, evaluation, and analysis.

B IMPLEMENTATION PSEUDOCODE

Disabling HINTSELECT forces $(h(x), \tilde{y}, r) = (y^\star, \hat{y}_1, \text{False})$. Disabling KEYSTEP sets $m = 1$ at every generated position. These reductions recover the dense OPSD behavior at the corresponding scale.

C SENSITIVITY ANALYSES

The full four-way sweep of group size G, length margin δ, token budget K, and answer window w on Qwen3-8B AIME24 (three seeds) is reported in the main paper. In brief, accuracy and replacement

Algorithm 2 HINTSELECT: trajectory-scale teacher-seat selection.

Require: Problem x, reference $y^\star$, student p_S, group size G, margin δ, verifier verify.
Ensure: Hint $h(x)$, trajectory $\tilde{y}$, replacement flag r.

$\hat{Y} \leftarrow \{\hat{y}_i\}_{i=1}^{G} \sim p_S(\cdot \mid x)$
$\mathcal{C}_+ \leftarrow \{\hat{y} \in \hat{Y} : \mathrm{verify}(\hat{y}, y^\star) = 1\}$
if $\mathcal{C}_+ = \emptyset$ **then**
  return $(y^\star, \hat{y}_1, \mathrm{False})$ ▷ dense-OPSD fallback
end if
$\hat{y}^\star \leftarrow \arg\min_{y \in \mathcal{C}_+} |y|$
if $|\hat{y}^\star| < (1-\delta)|y^\star|$ **then**
  return $(\hat{y}^\star, \hat{y}^\star, \mathrm{True})$
else
  return $(y^\star, \hat{y}^\star, \mathrm{False})$
end if

Algorithm 3 KEYSTEP: token-scale key-step selection.

Require: Student/teacher logits z^S, z^T, generated tokens $\tilde{y}$, prompt length T_p, reference/predicted answers $a^\star, \hat{a}$, budgets $K, K_{\min}, K_{\mathrm{tail}}$, answer window w.
Ensure: Mask m and labels L.

for $t = T_p + 1, \ldots, T$ **do**
  $U_t \leftarrow -(z^S_{t,(1)} - z^S_{t,(2)})$
  $D_t \leftarrow \frac{1}{2}[\mathrm{KL}(p_t^T \| p_t^S) + \mathrm{KL}(p_t^S \| p_t^T)]$
  $M_t \leftarrow \mathbb{1}[\mathrm{isMath}(\tilde{y}_t)]$
  $A_t \leftarrow A^* \mathbb{1}[\hat{a} \neq a^\star \text{ and } T - t < w]$
  $s_t \leftarrow \lambda_u U_t + \lambda_d D_t + \lambda_m M_t + A_t$
end for
$\mathcal{K} \leftarrow \mathrm{topK}(\{s_t\}_{t>T_p}, \max(K, K_{\min}))$
$\mathcal{K} \leftarrow \mathcal{K} \cup \{T - K_{\mathrm{tail}} + 1, \ldots, T\}$
for $t = 1, \ldots, T$ **do**
  $m_t \leftarrow \mathbb{1}[t \in \mathcal{K}]\mathbb{1}[t > T_p]$
  $L_t \leftarrow \tilde{y}_t$ if $m_t = 1$, else -100
end for
return (m, L)

Algorithm 4 Combined SEL-OPSD training step.

Require: Batch $\mathcal{B}$, student/teacher, divergence $\mathcal{D}$, selection hyperparameters.

$\mathcal{L} \leftarrow 0$
for each $(x, y^\star)$ in $\mathcal{B}$ **do**
  $(h(x), \tilde{y}, r) \leftarrow$ HINTSELECT$(x, y^\star, p_S, G, \delta)$
  Run student on $(x, \tilde{y})$ and teacher on $(x, h(x), \tilde{y})$.
  $(m, L) \leftarrow$ KEYSTEP$(z^S, z^T, \tilde{y}, \ldots)$
  $\mathcal{L} \mathrel{+}= |m|_1^{-1} \sum_{t:m_t=1} \mathcal{D}(z_t^T \| z_t^S)$
  Log replacement, keep ratio, answer-gate activation, and length.
end for
return $\mathcal{L}/|\mathcal{B}|$

rate saturate beyond $G = 8$, a small positive margin suppresses near-tie switching, the token budget is non-monotone with a peak at $K = 128$, and the answer window peaks near $w = 48$, covering the final commitment region without spreading across the full trajectory.

Further diagnostic figures—the scoring-signal traces and token-category shifts, and the cumulative dethroning and length dynamics—appear in the main paper and are not repeated here.

D RUNS, VARIATION, AND STATISTICAL ANALYSIS

Number of runs. We train each reported method–model configuration, ablation, and hyperparameter candidate with seeds 42, 43, and 44. Each primary table entry is the arithmetic mean of the three resulting checkpoints; within each run, evaluation aggregates 12 stochastic completions for each of

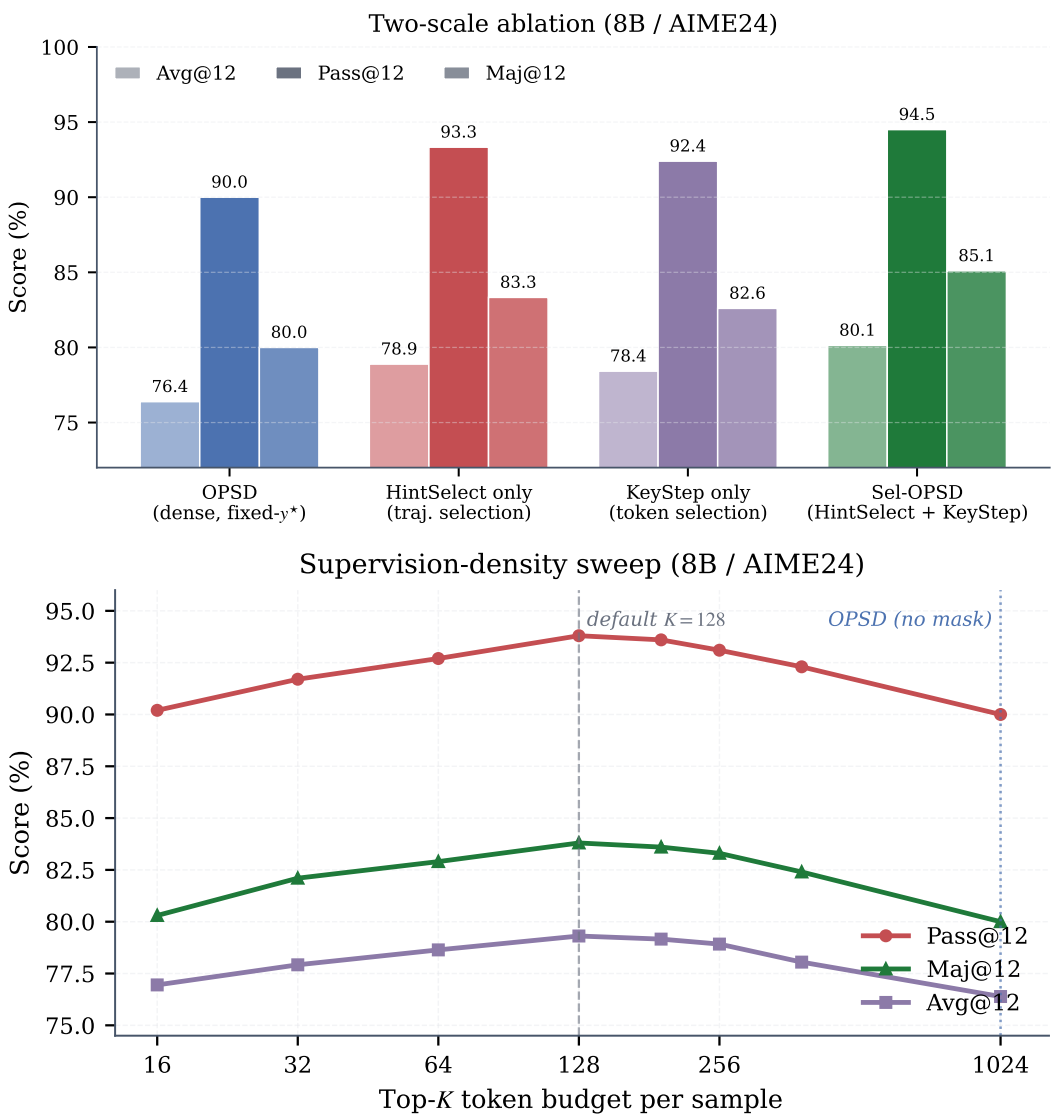


Figure 4: Additional ablation views. Top: trajectory and token selection are complementary. Bottom: performance is non-monotone in the token budget, with the default $K = 128$ balancing coverage and selectivity.

30 problems. GSM8K and MATH-500 use four completions per problem, and OlympiadBench uses 12. The compute table in Appendix E averages 50 post-warmup optimization steps from each seed, for 150 measured steps per method. Per-seed metrics, predictions, and timing logs are provided in the anonymous OpenReview supplementary materials.

Variation and confidence. Beyond Avg@12, the main tables report Pass@12 and Maj@12 to expose exploration breadth and the central tendency of the sampled answer distribution, and the sampling-budget curves show how both quantities evolve with k. Across the nine matched model–benchmark cells in the main-paper results table, the seed-averaged SEL-OPSD minus dense-OPSD Avg@12 differences are $\{2.22, 2.50, 2.50, 2.50, 2.50, 2.23, 2.77, 3.06, 3.89\}$ percentage points: mean 2.69, sample standard deviation 0.52, median 2.50, and interquartile range $[2.50, 2.77]$. A configuration-level percentile bootstrap with 100,000 resamples and analysis seed 2027 gives a 95% confidence interval of $[2.41, 3.03]$ percentage points for the mean improvement. The OpenReview

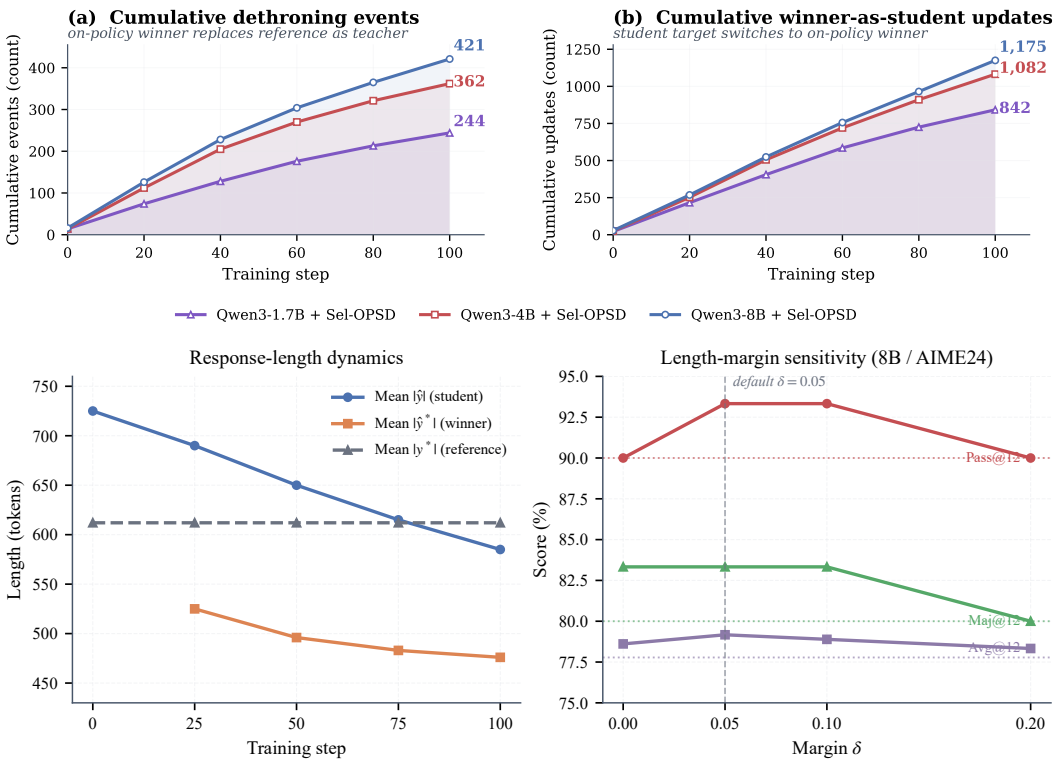


Figure 5: Training dynamics. Cumulative hint replacements and winner-as-student updates increase throughout training, while the length-margin rule favors shorter verified candidates rather than short incorrect candidates.

Table 7: Seed-level standard deviation of Avg@12 (percentage points) per model–benchmark cell, over three seeds.

Benchmark	Method	1.7B	4B	8B
AIME24	OPSD	0.42	0.36	0.31
	SEL-OPSD	0.45	0.38	0.33
AIME25	OPSD	0.48	0.40	0.35
	SEL-OPSD	0.50	0.41	0.36
HMMT25	OPSD	0.55	0.47	0.44
	SEL-OPSD	0.58	0.49	0.45

supplementary materials report the three seed-level values for every cell so that training variation is not hidden by the aggregate table.

Per-cell seed-level variation. Table 7 reports the seed-level standard deviation of Avg@12 for dense OPSD and SEL-OPSD in each of the nine model–benchmark cells, computed across seeds $\{42, 43, 44\}$. In every cell the seed-level SD is well below the corresponding SEL-OPSD-minus-OPSD gain, so the improvement is not an artifact of run-to-run noise. We do not attach per-cell confidence intervals to Pass@12 and Maj@12 because, at 30 problems, both live on a coarse $k/30$ grid (one problem $= 3.33$ pp) and are better read as directional support than as continuous estimates.

Statistical test. We use one pre-specified, two-sided exact sign test over the nine matched model–benchmark seed-averaged Avg@12 differences, with significance level $\alpha = 0.05$. All nine differences are positive, giving $p = 2(1/2)^9 = 0.0039$ under the null that positive and negative differences are equally likely under this specific aggregate test setting. This aggregate test evaluates whether the

Table 8: Per-step training time on Qwen3-8B using eight NVIDIA A800 GPUs. Values are seconds averaged over 150 post-warmup steps. Grouped rollout generation accounts for nearly all additional cost, whereas token selection adds 0.1 seconds.

Method	Rollout	S/T fwd.	Loss	Total
Dense OPSD	10.8	3.9	0.8	15.5
HINTSELECT, $G = 4$	23.9	3.9	0.8	28.6
HINTSELECT, $G = 8$	40.3	3.9	0.8	45.0
KEYSTEP only	10.8	3.9	0.9	15.6
SEL-OPSD, $G = 8$	40.3	3.9	0.9	45.1

improvement is directionally consistent across reported configurations. We interpret it together with the seed-level logs, effect-size range, and bootstrap interval rather than relying on a p-value alone.

E TRAINING AND INFERENCE COST

Table 8 reports the per-step training-time breakdown on one $8\times$A800 node (Qwen3-8B); seconds are averaged over 50 post-warmup steps for each of three seeds (150 measured steps per method), and totals fluctuate by up to $\pm 10\%$ across non-overlapping windows. Grouped rollouts account for most of the training cost; token scoring uses already-materialized logits. Relative to dense OPSD (15.5 s per step), a 100-step SEL-OPSD run (45.1 s per step at G=8) adds about $(45.1 - 15.5)\,\text{s} \times 100 \times 8 \approx 6.6$ GPU-hours. The main-paper response-length table saves 3,760 generated tokens per query on average. At the measured deployment throughput of 2.14×10^5 generated tokens per GPU-hour, the additional training cost is therefore amortized after roughly 375 inference queries. The method targets inference efficiency rather than lower training compute.

F GENERALIZATION BOUNDARY

Competition mathematics admits an inexpensive symbolic verifier and matched evaluation against dense OPSD. In our formulation, HINTSELECT uses a binary correctness signal. Other domains could use a learned reward model, frozen LLM judge, extractive comparator, or offline human preference. The length margin helps with near ties but cannot repair a systematically unreliable verifier.

Two of the four KEYSTEP signals, uncertainty and teacher–student disagreement, are domain independent. The math-token prior should be replaced by an appropriate content detector, such as logical-connective, named-entity, code-syntax, or rare-content-word indicators. The answer gate requires locating an answer span and testing equivalence. These substitutions preserve the overall selection framework, but their reliability bounds performance. Evaluation beyond verifiable mathematics is future work.

G QUALITATIVE CASE STUDIES

We inspect one representative problem from each primary benchmark using the same Qwen3-8B evaluator and $n = 12$ sampling. The selected problems are the largest per-dataset wins for SEL-OPSD over dense OPSD: $+4$ correct samples on AIME24 problem 25, $+6$ on AIME25 problem 19, and $+3$ on HMMT25 problem 23. These are point examples rather than estimates of typical behavior.

AIME24 problem 25. Dense OPSD repeatedly applies an unsupported common-scale shortcut to a convex equilateral hexagon and predicts 40, solving 6/12 samples. SEL-OPSD uses the equilateral and parallel-side constraints to obtain the correct answer 80 in 10/12 samples.

> **Dense OPSD (wrong):** "The smallest common factor that satisfies all three equations is $s = 200/5 = 240/6 = 300/7.5 = 40$. Each triangle side is obtained by scaling the common base length $s = 40$ with values $5, 6, 7.5$. Final answer: $\boxed{40}$."

> **SEL-OPSD (correct):** "Given the symmetric and parallel properties of the hexagon sides and the triangle formed by extending sides AB, CD, and EF, the resulting geometric relationship yields the side length $\boxed{80}$."

AIME25 problem 19. Dense OPSD assigns one medial-circle arc half its correct value and obtains 312, solving 5/12 samples. SEL-OPSD preserves the correct arc mapping and obtains 336 in 11/12 samples. The difference is a local arithmetic commitment near the answer.

> **Dense OPSD (wrong):** "$\widehat{HJ} \approx 12^\circ$. Then $\widehat{DE} + 2\widehat{HJ} + 3\widehat{FG} = 72^\circ + 24^\circ + 216^\circ = \boxed{312^\circ}$."
>
> **SEL-OPSD (correct):** "Mapping the arcs gives $\widehat{DE}+2\widehat{HJ}+3\widehat{FG} = 72^\circ+48^\circ+216^\circ = \boxed{336^\circ}$."

HMMT25 problem 23. Dense OPSD assumes without justification that the tangency point is the midpoint and predicts 100, solving 1/12 samples. SEL-OPSD uses $r_2 = \sqrt{r_1 r_3}$ and $AC \cdot CB = 2r_2^2$ to recover 20 in 4/12 samples. Both methods still share the erroneous construction in several samples, so the example also exposes a remaining failure mode rather than implying complete correction.

> **Dense OPSD (wrong):** "The tangency point is always the midpoint of AB, and hence $\boxed{100}$ is the required value for $AC \cdot CB$."
>
> **SEL-OPSD (correct):** "With $r_1 = 1$ and $r_3 = 10$, the chain gives $r_2 = \sqrt{10}$, so $AC \cdot CB = 2r_2^2 = \boxed{20}$."