SELD: Selective On-Policy Distillation for Mathematical Reasoning via Semantic Error Localization

Anonymous ACL submission

Abstract

On-policy distillation (OPD) has become a standard recipe for transferring mathematical reasoning from a strong teacher to a smaller student, and recent variants differ mainly in how they weight tokens of incorrect student rollouts: some apply uniform full-token knowledge distillation (KD), some discard incorrect rollouts entirely, and others reweight by teacher–student distributional disagreement. We revisit which parts of an incorrect rollout provide the most direct corrective signal. In an incorrect math rollout, the most informative region is often the point where the reasoning first goes wrong: the prefix before it is largely reasoning the student already performs correctly, while the suffix after it consists of next-token predictions conditioned on an already-wrong premise and may provide less reliable supervision. Building on this view, we propose **SELD** (Selective Error-Localized Distillation), an OPD method with two components: (i) a semantic first-error localization step in which the teacher reads the student's full wrong solution and identifies the step where the reasoning first breaks, a semantic judgment over the whole solution rather than a token-level distributional signal; and (ii) a hard selective KD objective that supervises only a small neighborhood around that step and masks the remaining incorrect-rollout tokens. Using Qwen3 models on four math benchmarks (AIME-25, AIME-24, AMC-23, MATH-500), SELD supervises only 23.28% of incorrect-rollout tokens yet improves avg@12 over full-token GKD/OPD by $+1.68$ points and over Correct-only KD by $+2.64$ points. A similar trend appears for a larger Qwen3-4B student.

1 Introduction

Knowledge distillation (Hinton et al., 2015) transfers the behavior of large language models to smaller students with lower inference cost. In mathematical reasoning, recent systems increasingly use on-policy distillation: the student generates its own rollout, and a stronger teacher supervises the trajectory token by token (Lin et al., 2020; Gu et al., 2023; Agarwal et al., 2024). This reduces the train–inference mismatch of teacher-forced distillation (Ranzato et al., 2016; Bengio et al., 2015; Kim and Rush, 2016), since the student is trained on states it actually visits.

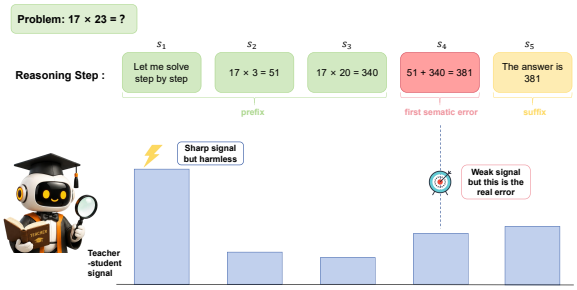

Figure 1: Signal–error mismatch in an incorrect math rollout. Upper track: reasoning steps colored as valid prefix (green), first semantic error (red), and wrong-premise suffix (yellow). Lower track: per-step teacher–student signal δ_t. The signal peaks at a stylistic paraphrase (**sharp but harmless**) yet is near-lowest at the first semantic error (**weak but wrong**).

Yet on-policy training also exposes the student to its own erroneous reasoning trajectories, and these trajectories are rarely uniformly wrong. A long solution typically has three regions: a valid prefix of correct reasoning before the error, the erroneous step itself, and a suffix of continuations derived from the wrong premise. The prefix is mostly reasoning the student already gets right, so supervision there is less targeted; the suffix has already drifted off the correct trajectory, so supervision there imitates continuations conditioned on the wrong premise. The neighborhood of the error therefore directly marks where the student should change its next reasoning step. From this perspective, Correct-only KD may discard the informative first-error region, while Full-token KD may dilute

its signal across prefix, error, and off-trajectory suffix. Consistent with recent evidence that a small subset of high-value tokens carries much of the useful training signal in reasoning models (Cheng et al., 2025; Wang et al., 2025), we explore whether supervision on incorrect rollouts can be concentrated at the transition from valid reasoning to erroneous continuation. This requires first locating that transition (Figure 1, upper track).

Existing on-policy KD methods that localize supervision often rely on sampled probability gaps between the teacher and student (Agarwal et al., 2024; Gu et al., 2023). These gaps are useful indicators of teacher–student disagreement, but the disagreement signal need not coincide with the first mathematical error. Consider a student asked "What is 17×23?" that splits 23 as $20 + 3$ and writes $17 \times 20 = 340$, then $17 \times 3 = 51$, and finally $340 + 51 = 381$. Two positions in this rollout illustrate the distinction. *Sharp but harmless:* an opening phrase such as "Let me solve this step by step" is stylistic boilerplate the teacher would phrase differently, producing a large probability gap on a span that carries no mathematical content. *Weak but wrong:* the arithmetic line is short and confidently expressed, so teacher and student agree token-by-token, yet $340 + 51 = 381$ is exactly the step that commits the student to the wrong answer of 381 instead of 391 (Figure 1, lower track). We quantify this mismatch on 300 real incorrect rollouts in Section 3.1.

We propose **SELD**, an on-policy KD method that addresses the two questions above (where the error is and where to supervise) by changing where supervision is placed on incorrect rollouts. To answer where the first mathematical error occurs, we ask the teacher to read the student's full wrong solution and identify the step where the reasoning first breaks; this provides a *semantic* judgment over the whole solution that complements token-level distributional signals. To answer where supervision should be placed, we restrict KD to that step and its immediate neighbors. The rest of the rollout receives zero loss, including both the correct reasoning before the error and the wrong-premise continuation after it. Correct rollouts keep standard uniform KD. This is hard selection, not reweighting: compared methods that use incorrect rollouts retain full-trajectory supervision with different per-token weights.

Across Qwen3-1.7B and Qwen3-4B students with a Qwen3-8B teacher (Yang et al., 2025) on AIME-25, AIME-24, AMC-23, and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), SELD shows the same improvement trend over full-token GKD/OPD; in the 1.7B setting, it improves over GKD/OPD by $+1.68$ points and over Correct-only KD by $+2.64$ points while supervising roughly one quarter of incorrect-rollout tokens. Ablations show that the gain is not from sparsity alone, that the useful mask location is specifically the first-error neighborhood, and that the teacher's semantic diagnosis outperforms distributional anchor signals at fixed mask shape (Section 5). We summarize our contributions as:

- We introduce the principle of *first-error-localized distillation*: for incorrect on-policy reasoning rollouts, supervision should concentrate near the first step where the reasoning goes wrong rather than be spread over the whole trajectory.
- We instantiate this principle with SELD, which asks the teacher to identify the student's first erroneous reasoning step and then applies a hard step-level KD mask around that step, while keeping standard uniform KD on correct rollouts.
- We validate both the principle and the method on Qwen3-1.7B and Qwen3-4B students across four math benchmarks: SELD improves over full-token GKD/OPD with a small incorrect-rollout token footprint, and ablations show that sparsity, first-error placement, and semantic anchoring each contribute to the gains.

2 Related Work

On-policy distillation for reasoning. Knowledge distillation transfers behavior from a stronger teacher to a smaller student (Hinton et al., 2015; Busbridge et al., 2025), and its sequence-level instantiation trains the student on teacher-forced contexts, producing an exposure-bias-style train–inference mismatch (Kim and Rush, 2016; Lin et al., 2020). On-policy distillation removes this mismatch by aligning teacher and student distributions on student-sampled trajectories: MiniLLM optimizes a reverse Kullback–Leibler (KL) objective via policy gradient for mode-seeking behavior (Gu et al., 2023), and Generalized Knowledge Distillation (GKD) generalizes this view by interpolating on- and off-policy data under a family of

divergences (Agarwal et al., 2024), a setup now widely used for distilling chain-of-thought reasoning (Wei et al., 2022; Hsieh et al., 2023; Lu, 2025; Hübotter et al., 2026). SELD keeps this on-policy KD setup, but changes only the loss support on verified-incorrect rollouts: correct rollouts still receive uniform full-token KD, while incorrect rollouts are supervised only in the neighborhood of a grounded first-error step.

Localizing supervision: disagreement vs. wrongness. When on-policy KD methods move beyond uniform weighting, they typically localize supervision via teacher–student distributional signals such as sampled probability gaps or KL / Jensen–Shannon divergence (JSD) style divergence (Gu et al., 2023; Agarwal et al., 2024). A closely related line argues that uniform full-trajectory supervision is itself suboptimal. Liu et al. (2026) observe a "local teachability collapse" in strong-to-weak OPD, and keep supervision only on segments where the teacher–student preference gap is still informative. Recent OPD diagnostics make a similar case from the distribution side, flagging positions where dense teacher supervision is reliable versus where it should be down-weighted (Armandpour et al., 2026; Li et al., 2026), and a small fraction of high-loss, high-entropy, or high-information tokens has repeatedly been shown to carry most of the useful training signal (Jiang et al., 2026; Cheng et al., 2025; Wang et al., 2025). SELD shares this locality premise but answers *where* differently. All of the above read "where" off distributional or loss-based signals on the student's trajectory, yet Figure 1 shows that disagreement and wrongness can be misaligned at the step level, where a large probability gap may sit on a stylistic paraphrase while a confidently written step can hide the first real mathematical error. SELD therefore asks the teacher to read the student's full wrong solution and identify the step where the reasoning first breaks, and turns this semantic diagnosis into a hard step-level KD mask, without training any additional scoring model.

3 SELD: Selective Localized Distillation

3.1 Preliminaries

SELD operates in the on-policy KD setting where the student trains on both verified-correct and verified-incorrect rollouts; the design question is how incorrect rollouts should contribute to token-level KD.

Observation: distributional peaks miss the first semantic error. A natural way to localize supervision on an incorrect rollout is to upweight tokens where the teacher and student disagree, using sampled probability gaps or KL/JSD-style divergence (Agarwal et al., 2024; Gu et al., 2023). We audit 300 incorrect rollouts using the sampled teacher–student probability gap and find two directions of mismatch. **Where signal is high, the rollout is rarely wrong**: the top per-step peak coincides with the audited first-error step in only 5% of rollouts, and lands in the first-third "setup" band in 55%. **Where the rollout is wrong, signal is often weak**: the audited error step is missing from the top-3 peaks in 55% and falls into the bottom quartile of step signals in 15%. A distributional peak is therefore not a reliable proxy for the first semantic error, even though it remains a useful measure of teacher–student disagreement. A concurrent line of work, Prefix-KD (Liu et al., 2026), localizes supervision via a different distributional signal, namely a change point in the teacher's local discriminative power along the rollout, and similarly lands well before the audited first-error step (release point at $\rho \approx 0.13$ vs. audited first error at $\rho \approx 0.41$): distributional locality and semantic locality are different objectives. The full audit protocol and an exponential moving average (EMA) smoothed variant are in Appendix C.

Notation and base KD objective. Let π_s be the student, π_t the frozen teacher, and $y = (y_1, \ldots, y_T)$ a student-sampled rollout for a prompt. Standard on-policy KD minimizes a token-level divergence along the rollout:

$$d_t(y) = D\big(\pi_t(\cdot \mid y_{<t}) \,\|\, \pi_s(\cdot \mid y_{<t})\big),$$
$$\mathcal{L}_{\text{KD}}(y) = \frac{1}{\sum_t w_t} \sum_{t=1}^{T} w_t \, d_t(y). \tag{1}$$

where D is a KL/JSD-style divergence. Existing OPD/GKD-style methods usually set $w_t = 1$ for all completion tokens. SELD keeps the same distillation objective, but changes the support of w_t on verified-incorrect rollouts.

3.2 Overview

For each prompt x, the student samples an on-policy rollout $y \sim \pi_s(\cdot \mid x)$, and a math-answer verifier compares the final answer in y against the ground truth. Verified-correct rollouts are handed to standard uniform KD with $w_t = 1$ for all completion tokens; verified-incorrect rollouts are the

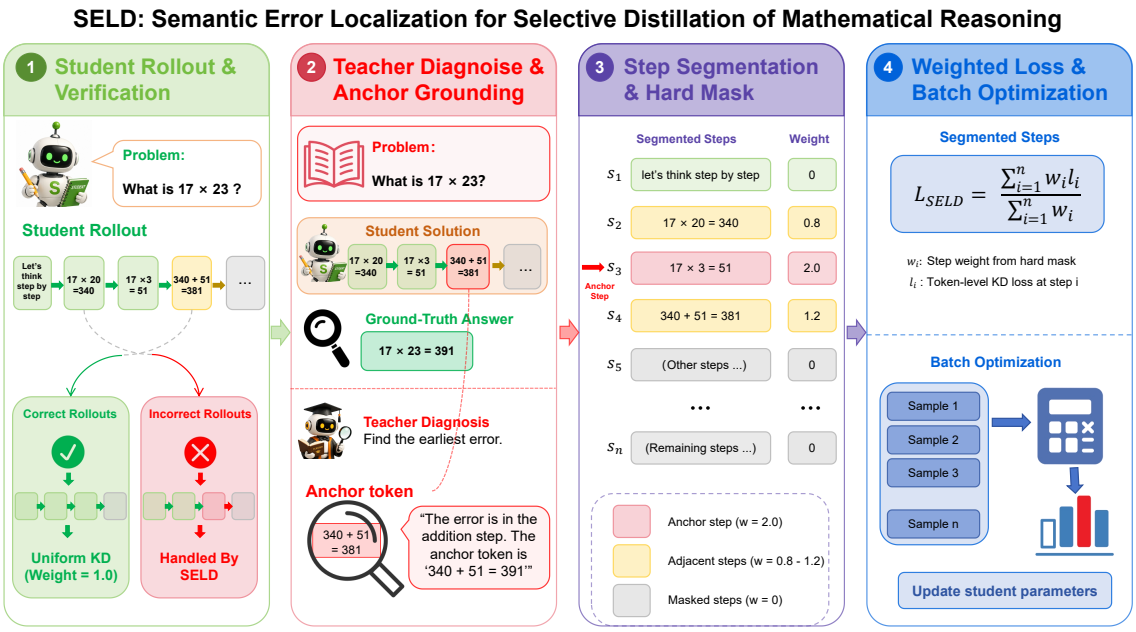

Figure 2: SELD pipeline. (1) The student rollout is verified; correct rollouts use uniform KD. (2) For incorrect rollouts, the teacher returns a verbatim error quote, grounded to an anchor step $s^\star$ via hierarchical step segmentation (ungrounded $\Rightarrow$ discarded). (3) A hard step-offset weight table $\{-1{:}0.8,\ 0{:}2.0,\ +1{:}1.2\}$ restricts KD to a ± 1-step window around $s^\star$. (4) Tokens are normalized within each valid rollout and then averaged across the batch, matching Eq. 3.

only branch on which SELD's loss support differs from OPD/GKD. On the incorrect branch, SELD proceeds in three further stages, shown left-to-right in panels 2–4 of Figure 2: (i) the teacher diagnoses the first semantic error and SELD grounds the diagnosis to an anchor via a verbatim quote (Section 3.3); (ii) the rollout is segmented into reasoning steps and a hard step-offset mask is built around the anchor step $s^\star$ (Section 3.4); and (iii) the per-token weights are aggregated into a per-sample-normalized KD loss over the surviving batch (Section 3.5).

3.3 Teacher Diagnosis and Anchor Grounding

For an incorrect rollout, the teacher receives the problem, the student's wrong solution, and the ground-truth answer, and returns a short JSON diagnosis whose central field is a verbatim `error_quote` copied from the student solution (Saunders et al., 2022; Madaan et al., 2023; Zheng et al., 2023; Uesato et al., 2022; Wang et al., 2024). The verbatim constraint is what makes the diagnosis checkable: the quote must literally appear in y, so an ungrounded quote is treated as a failed diagnosis rather than a noisy supervision signal.

SELD then grounds the quote to a token span in y using exact match, whitespace- and punctuation-normalized match, and a short-context-assisted fallback. A successful match yields an *anchor token* that Section 3.4 lifts to a step. If grounding fails, the incorrect rollout is *discarded* from the KD objective rather than being force-supervised at an estimated position; the surviving batch size is denoted B' in Eq. 3. On our training run this filter removes 13.95% of incorrect rollouts, the vast majority of which (Appendix D) stem from the teacher failing to follow the verbatim-quote instruction. Discarded and grounded rollouts have similar length and first-error-position distributions, suggesting that the filter mainly reflects quote-grounding compliance rather than an obvious difficulty bias.

3.4 Step Segmentation and Hard Mask

SELD weights tokens at the granularity of reasoning steps rather than fixed-width token windows, since mathematical errors typically occupy a complete semantic unit such as a case split or an equation transformation. Each completion is segmented by a lightweight hierarchy: explicit step or case markers when present, unambiguous numbered-list boundaries when available, and paragraph or format-tag boundaries otherwise (Appendix B.2). The anchor token is then mapped to the unique step

$s^\star$ that contains it.

Let $\mathrm{step}(t)$ denote the step containing token t and $\Delta s_t = \mathrm{step}(t) - s^\star$. For incorrect rollouts with a grounded anchor, SELD assigns:

$$w_t = \begin{cases} 0.8 & \text{if } \Delta s_t = -1, \\ 2.0 & \text{if } \Delta s_t = 0, \\ 1.2 & \text{if } \Delta s_t = +1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We use this asymmetric three-step window as a simple heuristic: the anchor step receives the strongest corrective signal, the preceding step preserves local context, and the following step exposes the immediate recovery continuation. The defining design choice is the last line: all non-neighborhood tokens in incorrect rollouts are fully masked, so KD is supported only on a three-step window around the diagnosed error.

3.5 Weighted Loss and Batch Optimization

Given the per-token weights w_t from Eq. 2 (incorrect, grounded) or $w_t = 1$ (correct), SELD optimizes:

$$\begin{aligned} \mathcal{L}_{\text{SELD}} &= \frac{1}{B'} \sum_{b=1}^{B'} \frac{1}{Z_b} \sum_{t=1}^{T_b} w_t^{(b)} d_t^{(b)}, \\ d_t^{(b)} &= D\big(\pi_t(\cdot \mid y_{<t}^{(b)}) \,\|\, \pi_s(\cdot \mid y_{<t}^{(b)})\big), \\ Z_b &= \sum_t w_t^{(b)}. \end{aligned} \tag{3}$$

The normalizer Z_b is taken over the same support that $w_t^{(b)}$ uses, so $Z_b = T_b$ for correct rollouts and $Z_b \approx \sum_{\Delta s_t \in \{-1,0,+1\}} w_t^{(b)}$ for grounded incorrect rollouts. Two aggregation choices matter. First, incorrect rollouts with ungrounded anchors are *excluded* from B', so SELD's effective batch contains only correct rollouts and successfully grounded incorrect rollouts. Second, the per-sample normalization keeps a sparse-mask incorrect rollout and a fully-supervised correct rollout on the same loss scale. Without it, a SELD incorrect-rollout loss would scale roughly as $|\text{three-step window}|/T_b$ relative to the correct branch, and sparser masks would silently reduce the gradient contribution of the very samples the method targets.

4 Experiments

4.1 Setup

Models. Our main experiments use **Qwen3-1.7B** and **Qwen3-4B** as students and **Qwen3-8B** as the frozen teacher (Yang et al., 2025). This student-teacher size gap is in the regime where distillation gains are nontrivial and not guaranteed (Li et al., 2025). All compared methods share the same tokenizer, optimizer, student rollouts, and evaluation protocol; unless otherwise stated, the 1.7B student is trained for 2 epochs with max sequence length 8192 and the 4B student for 1 epoch with max sequence length 4096, both at learning rate 3×10^{-6} and effective batch size 112. Full engineering details, including DeepSpeed/vLLM settings and implementation `EACD_*` environment variables, are in Appendix A.

Data. Training prompts come from the OPSD training set (Zhao et al., 2026), released on Hugging Face[1], with 29,434 prompts. All methods use the same data and rollout pipeline. For each prompt, the student samples an on-policy rollout; correctness is determined by a math-answer verifier against the ground-truth answer.

Baselines. All distillation methods apply uniform KD to verified-correct rollouts and differ only in how they use verified-incorrect rollouts. **Correct-only KD** discards incorrect rollouts and trains only on verified-correct ones. **GKD/OPD** (Agarwal et al., 2024) applies uniform token-level KD to both correct and incorrect on-policy rollouts. **Fixed-prefix KD** applies KD only to the first 50% of tokens of each incorrect rollout and zeros out the rest, replacing SELD's teacher-diagnosed first-error anchor with a fixed prefix cut; it is the natural ablation of the localization signal against a non-trajectory-specific prefix, in the spirit of the prefix-teachability line of work (Liu et al., 2026). **GRPO** (Group Relative Policy Optimization) (Shao et al., 2024; Guo et al., 2025) is included as a non-KD reinforcement-learning reference point trained on the same prompts and rollouts, to indicate where a comparable reinforcement-learning-style optimizer lands on the same protocol rather than as a direct KD competitor. **SELD** (ours) keeps full-token KD on correct rollouts and applies the 3-step hard mask centered on the teacher-diagnosed first-error step to each incorrect rollout, zeroing out the rest (Section 3.4).

Evaluation. We evaluate on four mathematical reasoning benchmarks: **AIME-25**, **AIME-24**, **AMC-23**, and **MATH-500 L3–5**. We report two

[1] https://huggingface.co/datasets/siyanzhao/Openthoughts_math_30k_opsd

Method	AIME-25		AIME-24		AMC-23		MATH-500		Avg.	
	avg	pass	avg	pass	avg	pass	avg	pass	avg	pass
Student: Qwen3-1.7B										
Correct-only KD	0.0806	0.2333	0.1056	**0.3667**	0.4710	0.8478	0.6567	0.8801	0.3285	0.5820
GKD / OPD	0.0750	0.1667	0.1167	0.3000	0.4909	0.8478	0.6698	0.9047	0.3381	0.5548
Fixed-prefix KD	0.0833	0.1667	**0.1278**	0.3333	0.4819	**0.8696**	0.6653	0.8665	0.3396	0.5590
GRPO	0.0556	0.1667	0.0722	0.1667	0.4638	0.8261	**0.6932**	**0.9101**	0.3212	0.5174
SELD	**0.1056**	**0.3000**	0.1250	0.3333	**0.5018**	**0.8696**	0.6870	0.9085	**0.3549**	**0.6028**
Student: Qwen3-4B										
GKD / OPD	0.1889	**0.4333**	0.1944	**0.4667**	0.6069	0.8478	**0.7936**	0.9291	0.4459	0.6692
SELD	**0.2083**	0.4000	**0.2083**	**0.4667**	**0.6341**	**0.9130**	0.7929	**0.9346**	**0.4609**	**0.6786**

Table 1: Main comparison across Qwen3-1.7B and Qwen3-4B students. Metrics are avg@12 and pass@12 under the shared decoding protocol.

metrics: **avg@12**, the mean accuracy over 12 sampled responses per problem with temperature 0.7 and top-$p = 0.8$, and **pass@12**, the fraction of problems solved at least once across the 12 samples (Chen et al., 2021). The aggregate score is the simple unweighted average over the four benchmarks, giving each benchmark equal weight despite differing sizes.

4.2 Main Results

Table 1 reports avg@12 and pass@12 for Qwen3-1.7B and Qwen3-4B students over AIME-25, AIME-24, AMC-23, and MATH-500 L3–5. SELD (ours) follows the teacher-guided localized KD objective defined in Section 3.

SELD reaches the highest four-benchmark avg@12 (0.3549) with a small incorrect-rollout token footprint, improving over Correct-only KD (0.3285) by +2.64 points and over full-token GKD/OPD (0.3381) by +1.68 points. The Correct-only gap shows that incorrect rollouts still carry useful training signal; the GKD/OPD gap shows that this signal does not require keeping the whole trajectory in the loss. As a simple localization ablation, **Fixed-prefix KD** uses a hard-coded 50% prefix cut; it trails SELD on the four-benchmark average by 1.53 avg@12 and 4.38 pass@12 points, even though it edges SELD on AIME-24 alone. GRPO (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025) lands lowest on average (0.3212); it edges out KD methods on MATH-500 L3–5 alone, but this does not carry to the four-benchmark average. For the Qwen3-4B student, SELD shows the same trend over GKD/OPD: +1.50 pts avg@12 (0.4609 vs 0.4459) and +0.94 pts pass@12 under the same protocol.

Method	Incorrect KD	Total KD	avg@12
Correct-only KD	0%	58.23%	0.3285
SELD	23.28%	67.95%	**0.3549**
Fixed-prefix KD	50.00%	79.12%	0.3396
GKD / OPD	100%	100%	0.3381

Table 2: Token-level supervision footprint vs. accuracy, sorted by total KD-token budget. *Incorrect KD* is the fraction of incorrect-rollout tokens with nonzero KD weight; *Total KD* is the resulting fraction of all completion tokens with nonzero KD weight. Footprints are measured from training logs (Appendix B.3).

4.3 Supervised-Token Footprint

We now quantify the token-level cost of SELD's accuracy gains. Of all student rollouts in training, 58.23% are verified-correct and 41.77% are verified-incorrect; on the latter, SELD restricts KD to the teacher-diagnosed 3-step neighborhood. Table 2 compares the resulting supervision support against the three KD baselines that differ from SELD only in how they treat incorrect-rollout tokens.

Two footprint facts follow. First, SELD cuts the total KD-token budget by 32.05% relative to full-token GKD/OPD while improving avg@12 by +1.68 pts. Second, Fixed-prefix KD supervises roughly twice as many incorrect-rollout tokens as SELD yet trails by 1.53 pts, showing that the gains come from supervising the *right slice*, not more of the incorrect rollout.

5 Ablations

Sections 4.2 and 4.3 establish that SELD improves over Correct-only KD and full-token GKD/OPD at a fraction of the token budget. This section asks *why*, isolating the contributions of sparsity (Sec-

Method	avg@12	pass@12
GKD/OPD (full-token)	0.3381	0.5548
Uniform Error Upweight	0.3298	0.5594
Random-step (3-step mask)	0.3431	0.5884
SELD (ours)	**0.3549**	**0.6028**

Table 3: Sparsity controls. GKD/OPD and Uniform Error Upweight keep all incorrect-rollout tokens active (100%); Random-step and SELD share the same 3-step budget. Only the mask shape and placement differ. Per-benchmark breakdown in Appendix E.1.

3-step mask placement	avg@12	pass@12
Prefix-only (before anchor)	0.3289	0.5674
Suffix-only (after anchor)	0.3278	0.5590
Random step	0.3431	0.5884
Teacher-guided (on anchor)	**0.3549**	**0.6028**

Table 4: Placement controls. All four rows share the same 3-step budget; only the center step of the mask differs. Per-benchmark breakdown in Appendix E.1.

tion 5.1), first-error placement (Section 5.2), and the choice of teacher-guided vs. distributional localization signals (Section 5.3) under matched rollouts, optimizer, and evaluation protocol.

5.1 Is Sparse Masking Alone Enough?

A first hypothesis is that SELD's gain comes from *sparsity per se*: keeping a small region of each incorrect rollout active, regardless of where that region lands, and zeroing out the rest. We test this with two controls that hold the rollouts and optimizer fixed and only change how incorrect-rollout tokens are weighted. **Uniform Error Upweight** keeps every token in an incorrect rollout but multiplies its KD weight by $2.0\times$ (Ko et al., 2024), training more heavily on errors without any masking. **Random-Step Localization** uses the same 3-step hard mask and step-offset weights as SELD, but selects the center step uniformly at random from the segmented reasoning steps and never queries the teacher.

Two facts follow. Neither full-token control beats a uniformly random 3-step mask: GKD/OPD (0.3381) and Uniform Error Upweight (0.3298) cluster around avg@12 $\approx$ 0.33–0.34, while a random 3-step mask reaches 0.3431 at less than a quarter of the token budget. Reweighting incorrect-rollout tokens by a flat $2\times$ does not recover what masking already buys. Teacher-guided localization then adds another $+1.18$ pts avg@12 and $+1.44$ pts pass@12 over the random placement under an identical mask, so sparsity is necessary but not sufficient: *where* the supervised region lands matters at least as much as the fact that it is sparse.

5.2 Is the First-Error Neighborhood Specifically Useful?

Section 5.1 shows that the placement of the 3-step mask carries roughly as much weight as its sparsity. This subsection asks whether the privileged placement is specifically the first-error neighborhood, or whether nearby placements (its valid prefix, its wrong-premise suffix) would do as well. We hold the 3-step budget fixed and shift it along the rollout. **Prefix-only** applies the same 3-step mask to the three steps immediately before the anchor, supervising the valid prefix instead of the error itself. **Suffix-only** applies it to the three steps immediately after the anchor, supervising the wrong-premise continuation. The Random-step row from Section 5.1 is repeated here as a sparsity-only baseline.

The first-error neighborhood is not just *a* good placement but a specifically good one. Shifting the same 3-step mask one neighborhood earlier or later drops avg@12 to 0.3289 and 0.3278, below even a uniformly random 3-step placement (0.3431). The useful corrective signal is therefore neither in the valid prefix the student already gets right nor in the wrong-premise continuation it derives after the error, but specifically at the transition where reasoning first breaks. This points to the first-error neighborhood as the placement at which corrective supervision is most concentrated, and motivates the remaining ablation: *which signal* reliably approximates that placement.

5.3 Why Teacher-Guided, Not Distributional?

A central design choice in SELD is to localize supervision via a teacher-provided semantic diagnosis of the first error rather than via student–teacher distributional disagreement. We test whether this choice matters on downstream accuracy by retraining SELD with two distributional anchor signals from prior on-policy KD work, holding the rest of the pipeline fixed: rollouts, optimizer, evaluation protocol, the 3-step hard mask, and the step-offset weights. **Sampled-gap** anchors on a local teacher–student sampled-token probability gap along the student rollout. **Adaptive-EMA** anchors on an adaptive divergence statistic computed from teacher–student token distributions and normalized by a running exponential moving average. Implementation details are in Appendix A.2.

Localization signal	AIME-25		AIME-24		AMC-23		MATH-500		Avg.	
	avg	pass	avg	pass	avg	pass	avg	pass	avg	pass
Sampled-gap	0.1000	0.2667	0.1139	**0.3333**	0.4964	0.8478	0.6739	0.8992	0.3461	0.5867
Adaptive-EMA	0.1000	**0.3333**	**0.1417**	**0.3333**	0.4710	**0.8696**	0.6637	0.8992	0.3441	**0.6089**
Teacher-guided (SELD)	**0.1056**	0.3000	0.1250	**0.3333**	**0.5018**	**0.8696**	**0.6870**	**0.9085**	**0.3549**	0.6028

Table 5: Downstream accuracy of three localization signals under an otherwise matched pipeline (same rollouts, optimizer, evaluation protocol, 3-step hard mask, and step-offset weights). Only the anchor-selection signal varies across rows.

Signal / Reference	Median ρ	Median norm. dist.
Sampled-gap	0.538	0.261
Adaptive-EMA	0.307	0.268
Human-audited first error	0.414	—
Teacher-guided (SELD)	**0.394**	**0.200**

Table 6: Anchor-position alignment on the 300-record audit. Median ρ is computed over all incorrect training rollouts. Median normalized distance is computed as $|s_{\text{signal}} - s_{\text{human}}|/(S-1)$, where S is the number of segmented reasoning steps.

Table 5 shows that teacher-guided localization is strongest on avg@12 (0.3549 vs 0.3461 for sampled-gap and 0.3441 for adaptive-EMA), while pass@12 is tightly clustered across the three signals (0.5867–0.6089).

Why semantic and distributional signals diverge. To see whether this accuracy gap traces back to where the signals actually place the mask, we record the normalized anchor position $\rho = t^\star/T$ for each incorrect rollout ($t^\star$ is the anchor token index, T the rollout length) and compare it to the human-audited first-error step from the 300-record audit mapped to the same scale.

Table 6 shows the same pattern at two levels. Globally, the teacher-guided anchor is essentially co-located with the audited first-error step, while distributional signals miss it on opposite sides: Sampled-gap fires late and Adaptive-EMA early; Prefix-KD's BIC release point sits earlier still ($\rho \approx 0.13$, Figure 3c of Liu et al., 2026). Per example, teacher-guided anchors also have the smallest normalized distance to the audited first-error step (0.200 vs 0.261–0.268), so the accuracy gap in Table 5 tracks both global position and local human-audit alignment.

The normalized distance complements the global median position because it is computed within each rollout's own step segmentation. This matters for mathematical solutions whose step granularity varies: one solution may combine an algebraic move into a single paragraph, while another may split the same move across several numbered steps. Measuring distance on the normalized step scale therefore asks whether the signal lands near the human first-error region, which is the criterion needed for placing a local KD mask. This distinction helps explain why the three signals behave differently under the same training recipe. Distributional signals track probability divergence, while teacher-guided anchoring uses the full solution context to target the first semantic break.

6 Conclusion

We introduced the idea of first-error-localized distillation for on-policy mathematical reasoning. Instead of treating an incorrect student rollout as either wholly useful or wholly harmful, this view treats it as a structured trajectory with a transition point: the reasoning is valid before the first error, breaks at the first erroneous step, and then continues under a wrong premise. The most useful corrective target is therefore the first-error neighborhood.

SELD instantiates this idea with a hard selective on-policy KD objective. Correct rollouts keep standard uniform KD, while incorrect rollouts are supervised only around the teacher-diagnosed first-error step. On Qwen3-1.7B and Qwen3-4B students with a Qwen3-8B teacher, SELD improves over both full-token GKD/OPD and Correct-only KD while supervising a small fraction of incorrect-rollout tokens.

The ablations separate the main ingredients behind this trend. Sparsity alone is not enough, the useful mask location is the first-error neighborhood, and semantic anchors better align with audited first-error steps than distributional ones under the same mask shape. Overall, SELD distills the corrective part of a wrong rollout.

Limitations

SELD targets step-by-step mathematical reasoning, where incorrect rollouts naturally factor into an identifiable first-error step. Domains with less step-local failures, such as long-form writing, multi-turn dialogue, or code generation, may require different segmentation rules. Our experiments also stay within the Qwen3 model family with a Qwen3-8B teacher, so cross-family robustness remains untested.

SELD also depends on the teacher diagnosis model. When the teacher misidentifies, paraphrases, or fails to ground the first error, the method either discards the rollout or supervises an imperfect anchor. Hybrid localizers that combine semantic diagnoses with distributional signals, or lightweight first-error detectors trained from teacher diagnoses, are natural extensions.

Finally, our footprint analysis measures tokens receiving nonzero KD weight, not realized compute savings. Wall-clock gains would require implementations that exploit this sparsity during computation.

Ethics Statement

The first-error annotations in Appendix C were produced by three author-team members, not external or paid annotators; no human-subject study was conducted and no IRB approval was required. We use the publicly released OpenThoughts-math-30k / OPSD dataset (Zhao et al., 2026) and Qwen3 student/teacher checkpoints (Yang et al., 2025) under their release terms. SELD is a training-side modification for mathematical reasoning and does not introduce new data collection, human-subject interaction, or capabilities beyond those of the underlying models.

References

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

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N. M. Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. 2026. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. 2025. Distillation scaling laws. In *International Conference on Machine Learning*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. MiniLLM: On-policy distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. 2026. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*.

Yuxuan Jiang, Runchao Li, Shubhashis Roy Dipta, Dawei Li, and Zhao Yang. 2026. Cornerstones or stumbling blocks? deciphering the rock tokens in on-policy distillation. *arXiv preprint arXiv:2605.09253*.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*.

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

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*.

Kaiyuan Liu, Ziyuan Zhuang, Yang Bai, Bing Wang, Rongxiang Weng, and Jieping Ye. 2026. Prefix teach, suffix fade: Local teachability collapse in strong-to-weak on-policy distillation. *arXiv preprint arXiv:2605.13643*.

Kevin Lu. 2025. On-policy distillation. Thinking Machines Lab: Connectionism (blog).

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

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

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *arXiv preprint arXiv:2506.01939*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Setting	Value
Student model	Qwen3-1.7B
Teacher model	Qwen3-8B
Training data	OPSD Hugging Face training set, 29,434 prompts
Training epochs / steps	2 epochs / 526 optimizer steps
Max training length	8192 tokens
Precision	bfloat16
Optimizer	AdamW (β_1=0.9, β_2=0.999)
Learning rate	3×10^{-6}
Schedule / warmup	cosine / 3% warmup
Weight decay	0
Max gradient norm	1.0
Microbatch / accumulation	2 per device / 8 steps
World size	7
Effective batch size	112
Gradient checkpointing	enabled
Attention implementation	FlashAttention
Student DeepSpeed stage	ZeRO-2
Teacher DeepSpeed stage	ZeRO-3
Checkpoint interval	100 steps
GKD on-policy ratio λ	1.0 (fully on-policy)
GKD KD temperature β	1.0
Student rollout temperature	1.0
Student rollout top-p	1.0
Student rollouts per prompt	1
Max student rollout length	8192 tokens
Anchor signal	teacher-guided
Mask outside anchor neighborhood	hard (weight 0)
Step weights (offset → weight)	$\{-1{:}0.8,\ 0{:}2.0,\ +1{:}1.2\}$
Ungrounded fallback anchor	discard rollout
Teacher diagnosis max length	200 tokens

Table 7: Training details for the main SELD run.

A Implementation Details

A.1 Training and evaluation setup

Table 7 summarizes the main SELD run. All compared distillation methods use the same student model, training data, tokenizer, optimizer family, and evaluation protocol unless otherwise stated.

Training runs on a single node with 8 NVIDIA A100 80GB GPUs, with 7 for the student and 1 reserved for the teacher diagnosis vLLM server, and the full 2-epoch / 526-step run completes in approximately 24 hours. The teacher is queried once per verified-incorrect student rollout, yielding ~ 25k diagnosis calls in total across training; because the teacher server is colocated on the same node, this is overlapped with the student forward/backward pass and does not become a throughput bottleneck.

Evaluation uses vLLM through the Swift/EvalScope native backend. For each problem we sample 12 completions with temperature 0.7, top-p 0.8, and a budget of 16,384 generated tokens. The evaluated datasets are AIME-25, AIME-24, AMC-23, and MATH-500 L3–5. We use zero-shot evaluation with the same long-reasoning system prompt across benchmarks and aggregate with mean accuracy (avg@12) and pass@12. Both training-time labeling and evaluation use the same final-answer verifier, which extracts the response with markers such as `\boxed{}`, normalizes it (LaTeX stripping, whitespace, numeric form), and compares it against the reference by exact string and numeric match; failed extractions are treated as incorrect.

A.2 Compared method details

All compared methods share the prompts, student rollouts, teacher model, tokenizer, optimizer, evaluation protocol, and correct-rollout supervision used by the main SELD run. The main baselines (Correct-only KD, GKD/OPD, Uniform Error Upweight, Fixed-prefix KD, and GRPO) follow the descriptions in Section 4. For the localization-signal ablation, only the anchor-selection signal varies; after an anchor is selected, the same 3-step hard mask and step-offset weights $\{-1{:}0.8, 0{:}2.0, +1{:}1.2\}$ from Table 7 are applied. The teacher-guided signal is described in Appendix B; below we spell out how the two distributional signals (Sampled-gap, Adaptive-EMA) score and select an anchor.

Sampled-gap signal. For every token t in the student completion of length N, let y_t be the token the student sampled. Write $p_t^{\mathrm{T}} = p^{\mathrm{teacher}}(y_t \mid y_{<t})$ and $p_t^{\mathrm{S}} = p^{\mathrm{student}}(y_t \mid y_{<t})$. The per-token gap is

$$g_t \;=\; \log p_t^{\mathrm{T}} \;-\; \log p_t^{\mathrm{S}}, \tag{4}$$

which is large where the teacher is more confident than the student on the token the student actually emitted. The signal is exponentially smoothed,

$$\tilde{g}_t \;=\; \alpha\, g_t + (1-\alpha)\, \tilde{g}_{t-1}, \tag{5}$$

with $\tilde{g}_0 = g_0$. The anchor selector scans $\tilde{g}_t$ from a skip prefix (the larger of the persistence window w and a 25%-of-length prefix, further extended past the chat-template / `<think>` prelude) and returns the first index $t^\star$ such that all three of the following hold:

$$\tilde{g}_{t^\star} \;\geq\; \tau_{\mathrm{abs}}, \tag{6}$$

$$\tilde{g}_{t^\star} - \tilde{g}_{t^\star-1} \;\geq\; \tau_{\mathrm{jump}}, \tag{7}$$

$$\frac{1}{w} \sum_{t^\star \leq t < t^\star + w} \tilde{g}_t \;\geq\; \tau_{\mathrm{abs}}. \tag{8}$$

We use the global defaults α=0.3, w=8, τ_{abs}=0.8 nats, τ_{jump}=0.5 nats. The selected token is mapped to the enclosing step under the segmentation rules in Appendix B.2.

Adaptive-EMA signal. The Adaptive-EMA selector keeps the same per-token gap g_t, the same EMA smoothing $\tilde{g}_t$, the same persistence-window test, and the same skip-prefix as Sampled-gap, but replaces the two fixed thresholds with data-adaptive ones derived from the rollout itself. Let $\mathrm{med}(\cdot)$ and $\mathrm{MAD}(\cdot)$ denote the median and the median-absolute-deviation, write $c = 1.4826$, and let $\delta_t = \tilde{g}_t - \tilde{g}_{t-1}$. Define rollout-level estimates

$$\hat{\sigma}_g = c \cdot \mathrm{MAD}(\tilde{g}), \tag{9}$$

$$\hat{\sigma}_\delta = c \cdot \mathrm{MAD}(\delta), \tag{10}$$

and set the thresholds as

$$\tau_{\text{abs}} = \max\big(\mathrm{med}(\tilde{g}) + k_{\text{abs}}\,\hat{\sigma}_g,\ \tau_{\text{abs}}^{\min}\big), \tag{11}$$

$$\tau_{\text{jump}} = \max\big(\mathrm{med}(\delta) + k_{\text{jump}}\,\hat{\sigma}_\delta,\ \tau_{\text{jump}}^{\min}\big). \tag{12}$$

We use k_{abs}=2.0, k_{jump}=1.5, $\tau_{\text{abs}}^{\min}$=0.3, $\tau_{\text{jump}}^{\min}$=0.15. Compared with Sampled-gap, the only change is the threshold rule: rather than a single global $(0.8,\ 0.5)$ pair, each rollout calibrates its own thresholds from its own gap distribution, so the same selector adapts to flat versus spiky gap profiles across rollouts.

A.3 Compute and throughput

We report per-step throughput and wall-clock for SELD and the on-policy KD baselines that share its training configuration (max length 8192, 2 epochs on `openthoughts_math_30k`, 8 A100-80GB total). SELD uses 7 student-training GPUs and 1 GPU for the teacher vLLM server; baselines use 8 training GPUs. SELD therefore takes 526 optimizer steps over the same dataset and epochs that baselines cover in 460 steps, because its effective batch on 7 GPUs is smaller by a factor of $7/8$.

Method	steps	s/it	wall (h)	GPU-h
Correct-only KD	460	250.8	32.0	256
Uniform err-upweight	460	248.7	31.8	254
Random-step anchor	460	255.4	32.7	261
SELD-main (hard)	526	**260.0**	**38.0**	**304**

Table 8: Training cost on 8 A100-80GB.

At matched sequence length, SELD's per-step time (260 s/it) is within 5% of the baselines (248–255 s/it): teacher diagnosis runs on its own GPU concurrently with student rollouts and is fully overlapped, and anchor grounding adds under 1 ms per record on CPU. End-to-end, SELD-main takes 38 wall-clock hours (304 GPU-h), 18% over the comparable baselines, with student rollout under `vllm-colocate` dominating the cost in every row.

B Method Pipeline Details

B.1 Teacher diagnosis and anchor grounding

For each verified-incorrect rollout, we strip chat-template wrappers to recover the readable student text, truncate it to 4,000 characters, and send it together with the problem and the reference answer (truncated to 2,000 characters) to the teacher. The system prompt instructs the teacher to find the first mathematical error and to copy a short exact substring from the student's solution whenever possible.

The core diagnosis instruction is:

Teacher diagnosis prompt

Find the earliest substantive mathematical error in the student's solution. Return JSON only. The `error_quote` must be a verbatim substring copied from the student's solution, preferably a short equation or phrase at the point where the reasoning first becomes wrong. Do not paraphrase the quote. If the solution is wrong from the start, quote the earliest unsupported setup or method choice. If the solution is incomplete, quote the first point where the solution stops making progress toward a valid answer.

The diagnosis output is constrained to JSON of the form

Diagnosis output schema

```
{
  "error_quote": "<verbatim substring>",
  "error_type": one of
      "wrong_from_start",
      "logic_error",
      "calculation_error",
      "wrong_method",
      "incomplete",
  "error_position_ratio": <float in [0,1]>
}
```

We parse either a full JSON response or the first brace-delimited JSON object containing `error_quote`; if parsing fails, the incorrect rollout is not used for SELD supervision.

Quote grounding then proceeds in two stages. First, we search for `error_quote` in the decoded completion-token sequence using exact substring matching followed by retries with whitespace, Unicode, and LaTeX-backslash normalization. Second, if the quote appears in the extracted student text

but not in the decoded token stream, we recover its position using a short surrounding context window. In the main setting, rollouts whose anchors can only be recovered from `error_position_ratio` (rather than from a grounded quote match) are discarded; only quote-grounded matches supervise training. Character positions are converted to token positions by binary searching decoded token prefixes, and the resulting token is mapped to a reasoning step for mask construction.

B.2 Step segmentation rules

The step segmenter operates on decoded text and maps character boundaries back to token indices. It always includes token 0 as the first boundary. Candidate boundaries are then collected in the following priority order.

1. **Major markers.** If present, SELD uses explicit reasoning markers such as Markdown headings, `Step`, `step`, `Case`, or `case`, optionally preceded by separator lines. These boundaries are accepted with a minimum spacing of 4 tokens.

2. **Numbered lists.** If no major markers are found, SELD considers numbered-list boundaries of the form `1.`, `1)`, etc. To avoid treating formula lists or option lists as reasoning steps, the list must begin with 1, the first boundary must occur within the first $\max(128, 0.2T)$ tokens, and the numbered sequence must contain a consecutive run of at least three items. Adjacent numbered spans longer than 512 tokens, or a final tail longer than 1024 tokens, cause the numbered-list fallback to be rejected.

3. **Paragraphs.** If neither major markers nor reliable numbered lists are available, SELD uses paragraph boundaries, defined as two or more newlines followed by non-whitespace text. Paragraph boundaries require a minimum spacing of 15 tokens.

Chat-template special tokens that mark the boundaries between the reasoning trace and the final-solution block are always added as candidate boundaries with a 4-token minimum spacing. After all candidates are collected, duplicate boundaries are merged, boundaries outside the token range are removed, and minimum-spacing constraints are applied in sorted order. This produces a monotone list of step-start token indices. The anchor token is assigned to the latest boundary not exceeding its token index, and the hard mask in Eq. 2 is applied to that step and its immediate neighbors.

Quantity	Value
Incorrect records	25,026
Grounded records	21,536
Grounding rate	86.05%
Ungrounded/discarded records	3,490
All incorrect tokens	71.91M
Grounded-record tokens	63.93M
Supervised tokens	16.74M
R_{err}	23.28%
R_{ground}	26.18%
Mean anchor position	0.446
Mean steps per grounded rollout	35.31
Mean active steps per grounded rollout	2.92

Table 9: Reconstruction statistics for the supervised-token ratios reported in Section 4.3.

B.3 Supervised-token reconstruction

The supervised-token ratios in Section 4.3 are reconstructed by replaying the segmentation-and-mask pipeline of Appendix B.2 on the main SELD anchor log, which records every incorrect rollout together with its grounded anchor token (when available), total completion-token count, teacher diagnosis metadata, and anchor source. Ungrounded or discarded fallback anchors contribute zero supervised tokens. Let N_{sup} be the number of completion tokens with nonzero mask weight and N_{tok} be the total completion-token count. Table 9 reports two ratios: $R_{\text{err}} = N_{\text{sup}}/N_{\text{tok}}$ uses an all-incorrect denominator, while R_{ground} restricts the denominator to grounded rollouts. The overall ratio over all training rollouts is not reported because this run did not log token totals for correct rollouts, which receive full-token KD.

C Anchor-Quality Audit

The human reference ρ used in Section 5.3 comes from 300 incorrect-rollout records (problem, student completion, reference solution) each labeled by three annotators (**A**, **B**, **C**) for the first reasoning step at which the student's solution becomes substantively wrong; ties are broken by majority vote, otherwise the three annotators adjudicate. The audited step is mapped to its token span and normalized by rollout length. The distributional and teacher-guided positions reported there are computed over all incorrect training rollouts, not this 300-record subset.

Inter-annotator agreement. All three annotators select the same step on 66.7% of records, exactly two agree on another 28.9%, and all three diverge on only 4.4%; average pairwise exact-step agreement is 76.3% and pairwise within-one-step agreement is 84.1%.

Per-annotator consistency. Each annotator individually matches the majority-vote consensus on at least 85% of records and lands within one step on at least 90%, with median offset 0 (Table 10). None of the three shows a systematic earlier/later bias relative to the consensus (mean normalized positions 0.460, 0.461, 0.434 vs. consensus 0.458), so the reference ρ is not an artifact of the majority-vote aggregation.

Annotator	Exact	Within ± 1	Med. $\lvert\Delta\rvert$
A	85.6%	91.1%	0
B	90.0%	94.4%	0
C	86.7%	90.0%	0

Table 10: Each annotator scored against the majority-vote consensus.

D Discarded vs. Grounded Incorrect Rollouts

Section 3.3 discards incorrect rollouts whose teacher-issued `error_quote` cannot be grounded. On our training log this filter removes 14.00% of incorrect rollouts. We check whether the discarded subset is a biased slice on length, truncation, first-error position, and teacher-assigned error type.

Distributional comparison. Completion length distributions are nearly identical (median 5,085 discarded vs. 5,023 grounded; rank-biserial $r = 0.04$). Truncation rate is 21.30% vs. 19.65% ($\Delta = +1.65$ pp). First-error position is essentially flat across the two subsets ($\max \lvert\Delta\rvert = 1.6$ pp across five bands). The largest shift is in teacher-assigned `error_type`: discarded rollouts are over-represented in 'Other' (21.0% vs. 9.2%) and under-represented in 'Computational Error' (28.7% vs. 35.4%); Cramér's $V = 0.147$, a small effect.

Root-cause of discarding. Classifying each discarded record by inspecting its `error_quote` against the student completion (Table 11), 86.0% of discards are teacher-side: the teacher returned a quote that is not a substring of the student's writing (paraphrase rather than copy), an empty quote, or a trivially short quote. The remaining 14.0% are cases where the quote does appear in the completion but grounding still fails.

Root cause	n	%
Teacher paraphrased (quote $\not\subset$ completion)	2781	79.10
Quote too short ($<$ 3 chars)	197	5.60
Empty quote	47	1.34
Quote in completion, grounding failed	491	13.96

Table 11: Root-cause breakdown of the discarded incorrect rollouts. The first three rows (86.0%) are teacher-side failures to comply with the verbatim quote instruction; the last row is pipeline-side.

Takeaway. Discarding therefore acts as a filter on teacher verbatim compliance, not on rollout difficulty.

E Additional Results and Cases

E.1 Per-benchmark ablation breakdown

Tables 12 and 13 report per-benchmark avg@12 and pass@12 for the ablation runs whose four-benchmark averages appear in Section 5.1 and Section 5.2. All rows use the same checkpoint-selection protocol, decoding configuration, and MATH-500 L3–5 subsetting as Table 1.

E.2 Teacher-diagnosed error types

To better understand what kinds of erroneous rollouts SELD supervises, Figure 3 summarizes the teacher-provided `error_type` labels from the same training anchor log used in Section 4.3. These labels are produced during teacher diagnosis rather than by an independent human taxonomy, but they provide a useful picture of the training signal available in incorrect trajectories: most incorrect rollouts are not simple arithmetic slips, with logic errors, wrong starts, and wrong methods accounting for nearly 90% of diagnosed failures.

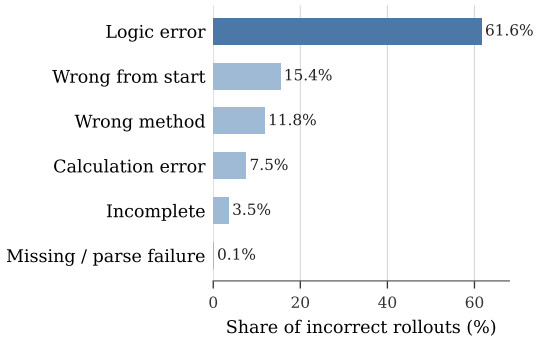

Figure 3: Teacher-diagnosed error-type distribution for incorrect training rollouts.

Method	AIME-25		AIME-24		AMC-23		MATH-500		Avg.	
	avg	pass	avg	pass	avg	pass	avg	pass	avg	pass
GKD/OPD (full-token)	0.0750	0.1667	0.1167	0.3000	0.4909	0.8478	0.6698	0.9047	0.3381	0.5548
Uniform Error Upweight	0.0833	0.2667	0.1167	0.3000	0.4583	0.7826	0.6610	0.8883	0.3298	0.5594
Random-step (3-step mask)	0.1000	0.3333	0.1250	0.3333	0.4891	0.8043	0.6583	0.8829	0.3431	0.5884
SELD (ours)	**0.1056**	0.3000	**0.1250**	**0.3333**	**0.5018**	**0.8696**	**0.6870**	**0.9085**	**0.3549**	**0.6028**

Table 12: Per-benchmark sparsity ablation (full version of Table 3 in Section 5.1). Four-benchmark averages match Table 3.

3-step mask placement	AIME-25		AIME-24		AMC-23		MATH-500		Avg.	
	avg	pass	avg	pass	avg	pass	avg	pass	avg	pass
Prefix-only (before anchor)	0.0889	0.2667	0.1111	0.2667	0.4565	0.8478	0.6592	0.8883	0.3289	0.5674
Suffix-only (after anchor)	0.1000	0.2667	0.1056	0.2333	0.4601	**0.8696**	0.6453	0.8665	0.3278	0.5590
Random step	0.1000	**0.3333**	**0.1250**	**0.3333**	0.4891	0.8043	0.6583	0.8829	0.3431	0.5884
Teacher-guided (on anchor)	**0.1056**	0.3000	**0.1250**	**0.3333**	**0.5018**	**0.8696**	**0.6870**	**0.9085**	**0.3549**	**0.6028**

Table 13: Per-benchmark locality ablation (full version of Table 4 in Section 5.2). All rows share the same 3-step budget; only the center step of the mask differs.

We also show one representative training rollout per category. Each row gives the problem prompt and the verbatim `error_quote` returned by the teacher; the quote is the span that SELD lifts to a reasoning step and centers its 3-step mask on.

Logic error. *Prompt.* A segment of length 1 is drawn with both endpoints on a unit circle, dividing the circle into two parts; compute the area of the larger region.
Quote. $A_{\text{larger}} = \pi - \frac{1}{2} + \frac{1}{2}\sin(\pi/3)$.
The closed-form mixes the chord-segment area formula incorrectly.

Wrong method. *Prompt.* A sidewalk 3 ft wide, 60 ft long, 3 in thick; concrete is ordered in whole cubic yards. How many?
Quote. "Since the contractor must order a whole number of cubic yards…"
The student names the integer-ordering constraint but then proceeds with continuous division instead of a ceiling.

Wrong start. *Prompt.* Show that $a^2 + b^2 + c^2 \geq ab + bc + ca$.
Quote. "Let me start by recalling some algebraic identities or inequalities that might help here."
The opening is a meta-comment about identities rather than the standard square-sum rearrangement.

Calculation error. *Prompt.* If a, b, c are n-digit positive integers, the number of digits in abc is: (A) $3n$, (B) $3n-1$, (C) $3n-2$, (D) None of the above.
Quote. "This is a very large number, but still has $3n - 3$ digits."
A local digit-count miscalculation in an otherwise correct case analysis.

Incomplete. *Prompt.* A triangle with perimeter 7 has integer side lengths; what is the maximum possible area?
Quote. "But wait, I should check if there are any ot…"
The rollout is cut off mid-sentence before the remaining integer-side-length cases are checked.

E.3 A confident fluent error invisible to distributional signals

This case study illustrates the failure mode introduced in Section 1 (Figure 1) on a real student rollout. The prompt is:

> *A cylinder with a volume of 9 is inscribed in a cone. The plane of the upper base of this cylinder truncates the original cone, forming a frustum with a volume of 63. Find the volume of the original cone.*

The student rollout (886 tokens, sampled by Qwen3-1.7B during training) produces a structurally clean derivation: it sets up similar-cone ratios, defines the cylinder-to-cone scale factor k, and proceeds in five labeled steps. In Step 4, however, it writes:

Position in completion	ratio	δ_t
Step 1 paraphrase of prompt (signal-anchor location)	0.420	$+1.62$
Step 4: V_{frustum} equation (true error)	0.685	-3.44

Table 14: Per-token Sampled-gap divergence on the cone–cylinder rollout. The signal-based anchor lands on a stylistic prompt paraphrase early in the trajectory; the genuine error in Step 4 is one of the quietest positions in the rollout.

> *"$V_{frustum}$ is the volume of the part of the original cone above the cylinder. The volume of the frustum is: $V_{frustum} = V_{cone} - V_{cylinder} = V_{cone} - 9$."*

This collapses the actual frustum (the lateral region surrounding the inscribed cylinder, not the full "cone minus cylinder" difference) into a single subtraction. The remaining steps execute that subtraction faithfully and arrive at the wrong final answer $\boxed{72}$. The substantive error is concentrated entirely in the equation above: every preceding step is locally valid, and every subsequent step is a deterministic consequence of the wrong premise.

What a distributional signal sees. We log the per-token sampled-gap signal δ_t used by our Sampled-gap anchor variant on the same trajectory. Table 14 summarizes the signal at three reference positions.

The signal-based anchor is selected at ratio 0.420, on the sentence "We are dealing with a cylinder inscribed in a cone, meaning the cylinder is entirely inside the cone…". This is a pure paraphrase of the prompt and contains no mathematical content, but it triggers a local divergence peak because phrasing choices at the start of a fresh section are high-entropy under both distributions. Meanwhile, at the position of the actual error ($V_{\text{frustum}} = V_{\text{cone}} - V_{\text{cylinder}}$), the signal is $\delta = -3.44$, among the quietest tokens in the entire trajectory.

What teacher diagnosis returns. On this same rollout, the teacher diagnosis pipeline returns a quote corresponding to the equation

$$V_{\text{frustum}} = V_{\text{cone}} - V_{\text{cylinder}},$$

classified as a `logic_error` at position ratio 0.685. The quote grounds exactly to the Step 4 equation, and SELD applies its three-step mask around that step: Step 4 receives the anchor weight, while Steps 3 and 5 provide local context and immediate recovery signal. Earlier setup steps remain masked.

Generalizing the failure mode. To quantify this failure mode, we reuse the Adaptive-EMA subset of the audit in Appendix C (300 incorrect rollouts, same annotators and adjudication protocol). We aggregate the logged per-token raw teacher–student gap (**Raw**) and its EMA-smoothed counterpart (**EMA**) to a per-step magnitude (max over tokens in the step), and compare the per-step signal profile against the audited first-error step. Table 15 reports both directions of the mismatch under both signals; the Raw column matches the summary numbers cited in Section 3.1. Both directions are amplified under EMA smoothing: the top peak coincides with the audited error in only 10% of rollouts (vs. 5% raw) and lands in the first-third setup band in 80% (vs. 55%), while the audited error step is missing from the top-3 peaks in 65% (vs. 55%) and falls into the bottom quartile in 20% (vs. 15%). The two directions together rule out the most charitable reading of distributional anchoring on incorrect math rollouts: peaks systematically land on early setup steps rather than at later semantic errors, and a peak-driven anchor selector therefore not only misses the right step but sometimes selects from the steps where signal is highest while the right step's signal is among the lowest. We therefore treat distributional anchor selection as a different objective from semantic error localization, not a noisy approximation of it.

Statistic ($N = 300$)	**Raw**	**EMA**
High signal is not the wrong step		
Top peak step = audited error step	5%	10%
Top peak in first-third setup band	55%	80%
Median percentile of top peak position	0.26	0.04
The wrong step's signal can be weak		
Audited error step *not* in top-3 peaks	55%	65%
Audited error step in bottom quartile	15%	20%
Median rank of audited error step	4	5

Table 15: Sampled teacher–student gap signal vs. audited first-error step on 300 incorrect rollouts. **Raw** is the per-token sampled gap; **EMA** is its exponential-moving-average-smoothed counterpart. Raw-column summary statistics are cited in Section 3.1.