WHEN TO USE THE SOLUTION: PRESERVING EXPLORATION IN ON-POLICY SELF-DISTILLATION FOR REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Does knowing the solution always lead to better supervision for reasoning models? On-policy self-distillation (OPSD) uses a solution-conditioned copy of the model to provide dense token-level supervision on student-generated trajectories. However, preferences formed with solution access may not be appropriate at every reasoning state, particularly when the model is still exploring competing continuations. Our experiments show that uniformly applying such supervision can degrade performance in thinking mode. Our token-level analysis further reveals that solution conditioning frequently changes the unprivileged self-teacher's leading candidate at high-entropy *fork states*, whereas the leading candidate is usually preserved at low-entropy *lock states*. These observations motivate adapting the supervision source to the uncertainty of the current reasoning state. We therefore propose Fork-Sensitive On-Policy Self-Distillation (FS-OPSD), which separates the decision of *when to use privileged supervision* from *how to shape the resulting targets*. Specifically, the method consists of two components: (1) *Source selection* is proposed to mitigate exploration inhibition by selectively applying unprivileged self-teacher at forks and privileged self-teacher at locks. (2) *Target reshaping* is proposed to balance the weights of the selected source, ensuring supervision targets are appropriately distributed. Our evaluation across five thinking models and three mathematical reasoning benchmarks shows gains in averaged avg@16 up to 8.26% over vanilla OPSD and up to 2.36% over corresponding base models. Code is available at `https://anonymous.4open.science/r/FS-OPSD-14A3/README.md`.

1 INTRODUCTION

Advances in LLMs have shown that CoT reasoning can substantially improve performance on complex tasks by enabling models to decompose problems, explore alternatives, and revise decisions before answering (Wei et al., 2022; Guo et al., 2025; Muennighoff et al., 2025; Snell et al., 2025). However, translating these capabilities into reliable gains through post-training remains challenging. For instance, off-policy distillation suffers a mismatch between training and inference because it trains student on trajectories different from those encountered at inference time.(Kim & Rush, 2016; Agarwal et al., 2024). Reinforcement learning is hard to assign credit to individual tokens and resulting in unstable training for its sparse rewards. On-policy distillation addresses these challenges by training on the student's own trajectories and providing token-level supervision from a teacher at the states the student actually visits. Building on this idea, on-policy self-distillation (OPSD) uses a privileged copy of the same model, typically supplied with a worked solution, to transform task-level supervision into dense targets (Zhao et al., 2026; Shenfeld et al., 2026; He et al., 2026).

However, vanilla OPSD does not reliably improve thinking models. Both recent work (Kim et al., 2026; Kaur et al., 2026; Shen et al., 2026; Peng et al., 2026) and our experiments find that vanilla OPSD mostly reduce reasoning performance. Typically, vanilla OPSD uses the solution-conditioned distribution as the supervision target at every student-visited state. This can inhibit exploration of alternative paths, while thinking models rely on trajectories of exploration, verification, and self-correction (Gandhi et al., 2025; Venhoff et al., 2025). As a result, the performance is degraded. This raises the question: *which token positions should privileged information be applied to?*

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.

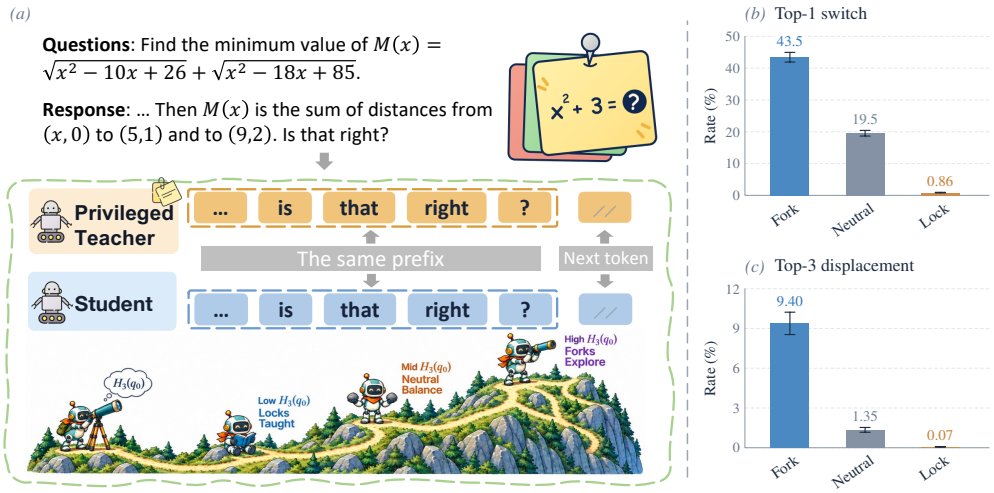


Figure 1: **State-dependent effects of solution conditioning.** **(a)** A reasoning example and the supervision principle. **(b)** Top-1 switching between q_0 and q_z. **(c)** Displacement of q_0's top token from q_z's top three. 92,160 positions from the first 1,536 response tokens are analyzed.

To examine this, we compare two distributions from the same self-teacher at each student-generated prefix, namely (1) q_0, conditioned only on the problem and prefix, and (2) q_z, which additionally receives the supplied solution. The top-three entropy of q_0 measures competition among the leading candidates. Inspired by Zhang et al. (2026), we call states with competing candidates *forks*, those with a dominant candidate *locks*, and those with intermediate levels of uncertainty *neutral*, as illustrated in Figure 1 (a). Our analysis of 60 Qwen3-4B thinking rollouts reveals a state-dependent pattern (Section 3.2, Figure 1 (b) and (c)). At forks, solution conditioning frequently changes q_0's leading candidate (43.5%) and can push it outside q_z's top three. At neutral, switching is less frequent (19.5%) but remains far more common than at locks. At locks, the leading candidate is usually preserved (0.86%), while relative candidate weights can still change.

Building on this, we propose Fork-Sensitive On-Policy Self-Distillation (FS-OPSD). The core idea is to adapt supervision source based on the uncertainty of current reasoning state. Our method consists of two components: (1) *Source selection*, which retains q_0 at fork and neutral states and admits q_z at locks, and (2) *Target reshaping*, which balances candidate weights and restricts support. The overview of our method is in Figure 2. *Source selection* is proposed to mitigate exploration inhibition from applying solution-conditioned supervision everywhere. Specifically, we retain q_0 at forks and neutral. At locks, q_z is admitted for solution-conditioned reweighting due to the low rate of leading-candidate changes. *Target reshaping* is proposed to control candidate weights. To this end, fork targets are softened to balance competing candidates. Then, top-k truncation and renormalization are applied to limit tail mass by restricting each target to its source's leading candidates. To demonstrate the effectiveness of FS-OPSD, we evaluate it on 5 thinking models and 3 mathematical reasoning benchmarks. FS-OPSD's averaged accuracy improves by up to 8.26% over vanilla OPSD and up to 2.36% over base models. Our contributions are summarized as follows:

- To mitigate thinking collapse in vanilla OPSD, we adapt distillation targets to uncertainty-defined reasoning states. This design selectively applies solution-conditioned supervision to guide reasoning while preserving exploration.
- We introduce FS-OPSD which consists of two components: (1) *Source selection* for mitigating exploration inhibition, and (2) *Target reshaping* for controlling candidate weights and limiting support for better performance.
- We evaluate FS-OPSD on 5 thinking models and 3 mathematical reasoning benchmarks, achieving up to 8.26% absolute avg@16 gains over vanilla OPSD and up to 2.36% over base models.

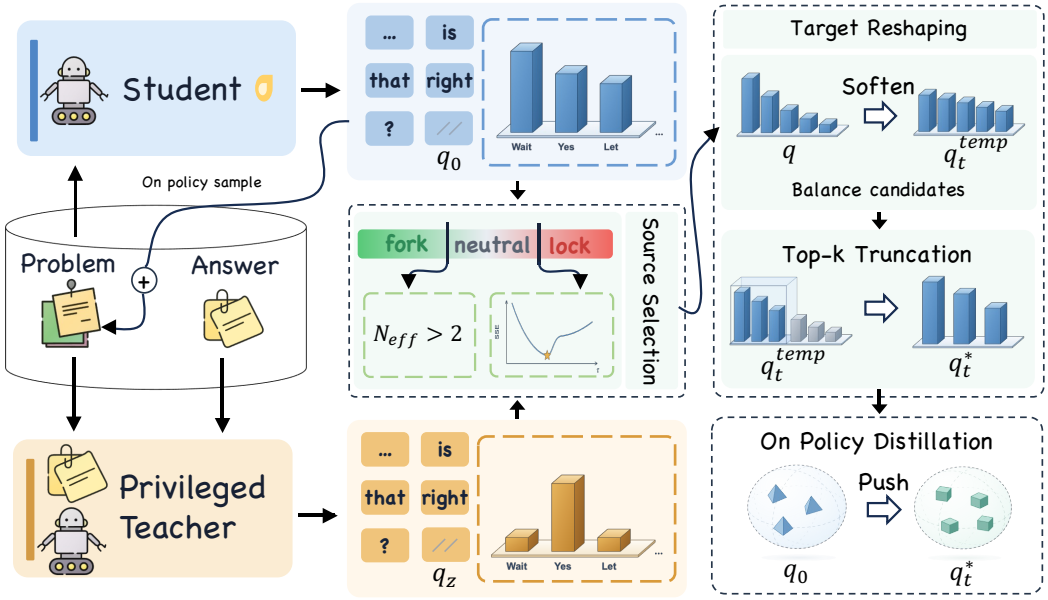


Figure 2: **Overview of FS-OPSD.** Prefixes are scored by an unprivileged and a solution-conditioned self-teacher. $H_3(q_0)$ selects q_0 at fork and neutral states and q_z at locks. Fork-specific softening and top-k truncation construct the target for the on-policy update. The distributions are schematic.

2 Related Work

Post-Training for Reasoning Models. Chain-of-thought (CoT) improves problem solving by enabling models to explore alternative paths(Wei et al., 2022; Wang et al., 2023). However, the benefits are not always preserved during post-training. For instance, (Wu et al., 2025) shows that sparse reinforcement learning can degrade intermediate reasoning despite improving final accuracy. Muennighoff et al. (2025) uses budget forcing to extend reasoning length. Peng et al. (2025) reduces redundant self-doubt by prompting models to assess question before answering. However, they do not address the risk that uniformly applied privileged supervision may suppress useful reasoning. We investigate this problem in on-policy self-distillation by examining whether solution-conditioned supervision should guide every position, and use privileged signals selectively to preserve exploration.

On-policy distillation. On-policy distillation trains students on its own trajectories, eliminating the mismatch between training and inference (Agarwal et al., 2024). Although exposure bias is avoided, recent work show that OPD does not guarantee positive gains. Li et al. (2026) identifies compatible thinking patterns and new capabilities as important conditions. Kim et al. (2026) associates self-distillation degradation with suppression of uncertainty expression. Kaur et al. (2026) shows that privileged context can reverse gains of OPD on thinking models. These findings indicate that matching the student's visited states does not by itself make privileged supervision appropriate throughout a trajectory. Our analysis moves beyond mismatch and shows that solution conditioning can disproportionately change candidate preferences. We address this issue through selectively using privileged supervision.

Token-level supervision and position selection. Lin et al. (2025) identify critical tokens and use token-level estimation to improve preference optimization. Wang et al. (2025) restrict RL updates to high-entropy tokens, achieving better performance than all tokens. In OPSD, Kaur et al. (2026) link privileged-context degradation to suppressed self-correction signals, but do not translate these diagnostics into a rule for selecting supervision sources. Shen et al. (2026) propose Purified OPSD, which uses a reference-only teacher to isolate a reference-induced component and constructs PMI-based targets to reduce reference-specific shortcuts. Peng et al. (2026) introduce AD-OPSD, which combines student entropy with sampled-token disagreement to adaptively blend privileged targets with a frozen base prior, mitigating thinking collapse. However, their corrected or blended targets can still reorder competing continuations, potentially discouraging exploration. We therefore treat whether to use privileged information as a separate decision from target reshaping.

3 FORK-SENSITIVE ON-POLICY SELF-DISTILLATION

To mitigate reasoning degradation, we propose Fork-Sensitive On-Policy Self-Distillation (FS-OPSD), which adapts the supervision source to the uncertainty of student-visited state (Figure 2). FS-OPSD consists of two components: (1) *Source selection* and (2) *Target reshaping*.

3.1 PRELIMINARIES

We first formalize OPSD and notation. Let $\mathcal{D} = \{(x, z)\}$ denote a training set. x is a problem and z is the solution. At each update, the current student policy p_θ samples an on-policy reasoning trajectory $y = (y_1, \ldots, y_T) \sim p_\theta(\cdot \mid x)$. The prefix generated before token t defines a student-visited state

$$s_t = (x, y_{<t}). \tag{1}$$

In OPSD, the student and teacher share the same model p_θ, while teacher receives solution z as privileged information. To distinguish teacher and student, we denote self-teacher model as $p_{\bar{\theta}}$, which is initialized from the same checkpoint as the student and remains frozen during training. For every token v, we define

$$q_0(v \mid s_t) \quad \equiv p_{\bar{\theta}}(v \mid x, y_{<t}), \tag{2}$$

$$q_z(v \mid s_t, z) \equiv p_{\bar{\theta}}(v \mid x, z, y_{<t}). \tag{3}$$

Vanilla OPSD treats q_z as the distillation target everywhere (Zhao et al., 2026; He et al., 2026). Using a token-level divergence D and the stop-gradient operator $\mathrm{sg}[\cdot]$, its objective is

$$\mathcal{L}_{\mathrm{OPSD}}(\theta) = \mathbb{E}_{(x,z)\sim\mathcal{D},\, y\sim p_\theta(\cdot|x)} \left[\frac{1}{T} \sum_{t=1}^{T} D(\mathrm{sg}[q_z(\cdot \mid s_t, z)] \,\|\, p_\theta(\cdot \mid s_t)) \right]. \tag{4}$$

3.2 UNCERTAINTY-DEFINED REASONING STATES

Prior studies find that particular token choices can strongly affect downstream outcomes (Bigelow et al., 2025; Lin et al., 2025), and associate high-entropy regions with branching and exploratory reasoning (Wang et al., 2025; Cheng et al., 2026). We characterize this local uncertainty through competition among the leading candidates under q_0. Let $\mathcal{V}_t^{(3)}$ contain the three tokens with the highest probability under $q_0(\cdot \mid s_t)$. We renormalize their probability mass and compute the top-three entropy using natural logarithms:

$$\begin{aligned} \widetilde{q}_0(v \mid s_t) &= \frac{q_0(v \mid s_t)}{\sum_{u\in\mathcal{V}_t^{(3)}} q_0(u \mid s_t)},\ v \in \mathcal{V}_t^{(3)}; \\ H_3(q_0) &= -\sum_{v\in\mathcal{V}_t^{(3)}} \widetilde{q}_0(v \mid s_t) \ln \widetilde{q}_0(v \mid s_t). \end{aligned} \tag{5}$$

This statistic focuses on the leading candidates while reducing sensitivity to the diffuse low-probability tail. We distinguish *fork* states with competing candidates, *lock* states with concentrated preferences, and an intermediate *neutral* regime:

$$r(s_t) = \begin{cases} \text{fork}, & H_3(q_0) \geq \tau_{\text{fork}}, \\ \text{neutral}, & \tau_{\text{lock}} < H_3(q_0) < \tau_{\text{fork}}, \\ \text{lock}, & H_3(q_0) \leq \tau_{\text{lock}}. \end{cases} \tag{6}$$

Section 3.3 specifies these thresholds and their selection rationale. We then compare q_0 and q_z on identical prefixes to measure how solution conditioning changes these candidate preferences. Let $a_t^0 = \arg\max_v q_0(v \mid s_t)$ be the unprivileged teacher's preferred next token, and let $\mathcal{V}_{z,t}^{(3)}$ contain top three tokens under q_z. We define two complementary indicators:

$$I_t^{\text{switch}} = \mathbf{1}\left[\arg\max_v q_z(v \mid s_t, z) \neq a_t^0 \right], \tag{7}$$

$$I_t^{\text{drop}} = \mathbf{1}\left[a_t^0 \notin \mathcal{V}_{z,t}^{(3)} \right]. \tag{8}$$

Switch records a change in the preferred token; *drop* captures the stronger displacement of that token outside the privileged teacher's top three. We analyze 60 Qwen3-4B thinking rollouts from OpenMathReasoning, using the first 1,536 response tokens of each rollout for 92,160 matched-prefix positions and reporting prompt-level means within each regime. Figure 1 shows a clear gradient: *switch* falls from 43.5% at forks to 19.5% at neutral states and 0.86% at locks, while *drop* falls from 9.40% to 1.35% and 0.07%. We complement these statistics with a case study comparing reasoning continuations under q_0 and q_z (Details are in Appendix A). These observations identify where solution access most strongly changes local supervision targets which can inhibit exploration and motivate conditioning source selection on solution-free uncertainty.

3.3 Source Selection

Solution-conditioned preferences at unresolved states can downweight continuations through which the student would otherwise explore, verify, or revise its reasoning. This motivates retaining q_0 at fork and neutral states, while admitting q_z at locks. We implement this with a binary privilege gate:

$$\beta_t = \mathbf{1}[H_3(q_0) \le \tau_{\text{lock}}], \tag{9}$$

$$q_t^{\text{src}}(v) = (1-\beta_t) q_0(v \mid s_t) + \beta_t q_z(v \mid s_t, z). \tag{10}$$

Fork. We require fork states to have more than two effective candidates, which means that $N_{\text{eff}} = \exp(H_3(q_0)) > 2$. We set $\tau_{\text{fork}} = 0.75$, slightly above the corresponding entropy boundary $\ln 2$ for stricter decision. The detailed derivation is given in Appendix B.1.

Lock. Let $p = \widetilde{q}_0(a_t^0 \mid s_t)$ be the leading probability within the renormalized top-three distribution. Then, we adopt the log-odds criterion $\ln[p/(1-p)] \ge 3$, giving $p \ge p^* = (1+e^{-3})^{-1} \approx 0.9526$ (Kass & Raftery (1995)). For fixed p, splitting the residual mass equally maximizes entropy. The resulting bound $f(p) = h_{\text{b}}(p) + (1-p)\ln 2$, where h_{b} is binary entropy, decreases for $p > 1/3$. Hence,

$$H_3(q_0) \le f(p) \le f(p^*) = h_{\text{b}}(p^*) + (1-p^*)\ln 2 \approx 0.2237. \tag{11}$$

This yields a reference near 0.22 for the entropy gate. Appendix B.2 gives the full derivation. After theoretical derivation, we calibrate this value by fitting a change point in leading-candidate concentration. For each position i, let $h_i = H_3(q_0(\cdot \mid s_i))$ and $c_i = \widetilde{q}_0(a_i^0 \mid s_i)$ denoting the top-1 probability within the renormalized top-three distribution. For each candidate τ, we fit a constant mean on either side, $\bar{c}_L(\tau)$ and $\bar{c}_R(\tau)$, and compute the residual sum of squares:

$$S(\tau) = \sum_{i:h_i<\tau} \big(c_i - \bar{c}_L(\tau)\big)^2 + \sum_{i:h_i\ge\tau} \big(c_i - \bar{c}_R(\tau)\big)^2. \tag{12}$$

Figure 3(a) reports $S(\tau)$ for three Qwen3-4B trace pools: OpenMathReasoning and two OpenThoughts samples (selection seeds 42 and 43), each with 60 prompts and up to 1,536 response tokens per prompt. All three curves attain their minimum at $\tau = 0.22$, the same as our theoretical prediction. As a result, we select $\tau_{\text{lock}} = 0.22$ in our main experiments.

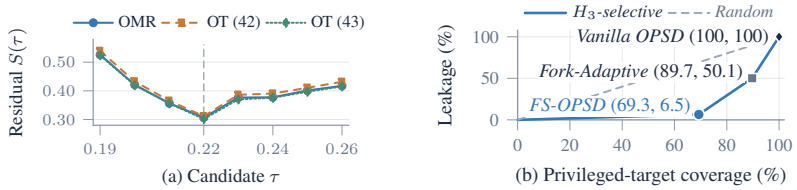
3.4 Target Reshaping

With the source fixed, we reshape its relative candidate weights and retained support (Zhang et al., 2026). At forks, softening reduces the dominance of the leading candidate to maintain competition among alternatives, which is beneficial for exploration. We use temperature $T_t = T_f = 1.5$ at fork states and $T_t = 1$ at neutral and lock states:

$$q_t^{\text{temp}}(v) = \frac{q_t^{\text{src}}(v)^{1/T_t}}{\sum_{u\in\mathcal{V}} q_t^{\text{src}}(u)^{1/T_t}}. \tag{13}$$

Temperature scaling preserves the selected source's candidate ranking, allowing fork softening to make alternatives more competitive while retaining q_0's preferred next token. For any two candidates u, v, their log-probability ratio changes according to

$$\log \frac{q_t^{\text{temp}}(u)}{q_t^{\text{temp}}(v)} = \frac{1}{T_t} \log \frac{q_t^{\text{src}}(u)}{q_t^{\text{src}}(v)}. \tag{14}$$



(a) Candidate τ (b) Privileged-target coverage (%)

Figure 3: **Source selection.** (a) Threshold calibration. (b) Privilege admission.

Fork softening therefore narrows preference gaps, while $T_t = 1$ preserves the selected source's relative weights at neutral and lock states. Appendix C provides a detailed derivation. This gives neutral states a distinct role: withholding privilege does not automatically trigger additional softening.

Because softening also redistributes mass toward the vocabulary tail, we restrict the target to its k leading candidates and renormalize. Support restriction has been studied in decoding as desmoothing (Hewitt et al., 2022) and in OPD as teacher-supported local distribution matching (Fu et al., 2026). Since temperature scaling preserves rankings, the retained set can be written as $\mathcal{K}_t = \mathrm{TopK}(q_t^{\mathrm{src}}, k)$. The final target is

$$q_t^\star(v) = \frac{q_t^{\mathrm{temp}}(v)\,\mathbf{1}[v \in \mathcal{K}_t]}{\sum_{u \in \mathcal{K}_t} q_t^{\mathrm{temp}}(u)}. \tag{15}$$

We fix $k = 20$ in all three regimes. The resulting fork target balances weights within the retained candidate set, while neutral and lock targets preserve their sources' relative weights within that set.

3.5 Objective and Training Procedure

FS-OPSD matches the current student to the stop-gradient target at every generated position:

$$\mathcal{L}_{\text{FS-OPSD}}(\theta) = \mathbb{E}_{(x,z)\sim\mathcal{D},\, y\sim p_\theta(\cdot|x)}\left[\frac{1}{T}\sum_{t=1}^{T} \mathrm{JSD}(\mathrm{sg}[q_t^\star] \,\|\, p_\theta(\cdot \mid s_t))\right], \tag{16}$$

where $\mathrm{sg}[\cdot]$ stops gradients and JSD is the symmetric Jensen–Shannon divergence. For each sampled minibatch, generated tokens and teacher targets are treated as fixed; gradients are taken only through the student, and the teacher remains frozen. Algorithm 1 summarizes the procedure. The deployment policy requires only the problem. The solution affects training through targets selected at lock states.

Algorithm 1 FS-OPSD

Require: Dataset $\mathcal{D} = \{(x, z)\}$; model p_θ; divergence D(e.g. JSD)

Initialize frozen teacher $p_{\bar{\theta}}$ from p_θ; let $p_S = p_\theta$ and $p_T = p_{\bar{\theta}}$.
for each training iteration **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{D}$
  for all $(x, z) \in \mathcal{B}$ **do**
    Sample $y \sim p_S(\cdot \mid x)$
    for each token position t **do**
      $q_0 \leftarrow p_T(\cdot \mid x, y_{<t})$, $\quad q_z \leftarrow p_T(\cdot \mid x, z, y_{<t})$
      if $H_3(q_0) \geq \tau_{\text{fork}}$ **then** $q_t^{\mathrm{src}} \leftarrow q_0$, $\quad T_t \leftarrow T_f$ ▷ fork
      else if $H_3(q_0) > \tau_{\text{lock}}$ **then** $q_t^{\mathrm{src}} \leftarrow q_0$, $\quad T_t \leftarrow 1$ ▷ neutral
      else $q_t^{\mathrm{src}} \leftarrow q_z$, $\quad T_t \leftarrow 1$ ▷ lock
      Retain the top-k tokens of q_t^{src} as $\mathcal{K}_t$
      $q_t^\star(v) \leftarrow \dfrac{q_t^{\mathrm{src}}(v)^{1/T_t}\,\mathbf{1}[v \in \mathcal{K}_t]}{\sum_{u \in \mathcal{K}_t} q_t^{\mathrm{src}}(u)^{1/T_t}}$
    Compute token-wise divergence $\ell(x, z) \leftarrow \dfrac{1}{|y|}\sum_{t=1}^{|y|} D(\mathrm{sg}[q_t^\star] \,\|\, p_S(\cdot \mid x, y_{<t}))$
  Calculate loss $\mathcal{L}(\theta) \leftarrow \dfrac{1}{|\mathcal{B}|}\sum_{(x,z)\in\mathcal{B}} \ell(x, z)$ and update θ

4 EXPERIMENTS

We conduct experiments to answer the following research questions:

- RQ 1: How does FS-OPSD compare to base model and OPSD in reasoning performance? (§4.2)
- RQ 2: Does retaining solution-free supervision at forks improve reasoning? (§4.4.1)
- RQ 3: How does divergence (JSD, forward KL, or reverse KL) affect FS-OPSD? (§4.4.2)
- RQ 4: How sensitive is FS-OPSD's performance to τ_{lock} and T_f? (§4.4.3)

4.1 EXPERIMENTAL SETUP

Models and Datasets. We evaluate reasoning models spanning multiple architectures and scales, including Qwen3 (Yang et al., 2025), DeepSeek-R1-Distill (Guo et al., 2025), and GLM-Z1 (GLM et al., 2024). We use the same fixed 15,000-example subset of `Openthoughts_math_filtered_30K` (Ashkchamp, 2025) as training data. In this training set, each problem is paired with a solution that provides privileged information during training.

Baselines and Evaluation. We compare (1) **Base**, the unmodified checkpoint; (2) **OPSD**, using q_z at every student-visited state with symmetric JSD; and (3) **FS-OPSD**. All models are evaluated in thinking mode on AIME 2024 (Hugging Face H4, 2025), AIME 2025 (Zhang & Math-AI, 2025), and HMMT 2025 (Dekoninck et al., 2026). We report *avg@16* for each benchmark; see Appendix D for training and evaluation details.

4.2 MAIN RESULTS

Table 1 presents the main results on five thinking models and three benchmarks. We highlight three key findings below.

Table 1: **Mathematical reasoning performance (avg@16, %) across five models.** Thinking mode is enabled for all models. Average is the mean across benchmarks, and bolded scores indicate the best-performing method for each model and metric. Small arrows show the absolute percentage-point change from Base. Additional results are reported in Appendix E.

Method	AIME 2024	AIME 2025	HMMT 2025	Average
`Qwen3-4B`				
Base	72.08	64.38	41.04	59.17
OPSD	67.92 ↓4.16	55.42 ↓8.96	36.04 ↓5.00	53.13 ↓6.04
FS-OPSD (Ours)	**74.58** ↑2.50	**66.46** ↑2.08	**43.13** ↑2.09	**61.39** ↑2.22
`Qwen3-8B`				
Base	77.29	66.46	44.58	62.78
OPSD	71.67 ↓5.62	59.38 ↓7.08	37.29 ↓7.29	56.11 ↓6.67
FS-OPSD (Ours)	**77.50** ↑0.21	**66.88** ↑0.42	**45.21** ↑0.63	**63.19** ↑0.41
`DS-R1-Distill-Qwen-7B`				
Base	52.29	40.00	22.92	38.40
OPSD	55.00 ↑2.71	34.38 ↓5.62	21.46 ↓1.46	36.94 ↓1.46
FS-OPSD (Ours)	**55.21** ↑2.92	**42.08** ↑2.08	**25.00** ↑2.08	**40.76** ↑2.36
`DS-R1-Distill-Llama-8B`				
Base	43.75	26.88	20.00	30.21
OPSD	43.13 ↓0.62	26.46 ↓0.42	18.96 ↓1.04	29.52 ↓0.69
FS-OPSD (Ours)	**46.04** ↑2.29	**30.63** ↑3.75	**20.42** ↑0.42	**32.36** ↑2.15
`GLM-Z1-9B-0414`				
Base	66.25	**50.00**	39.17	51.81
OPSD	62.92 ↓3.33	45.83 ↓4.17	35.00 ↓4.17	47.92 ↓3.89
FS-OPSD (Ours)	**67.92** ↑1.67	49.79 ↓0.21	**39.58** ↑0.41	**52.43** ↑0.62

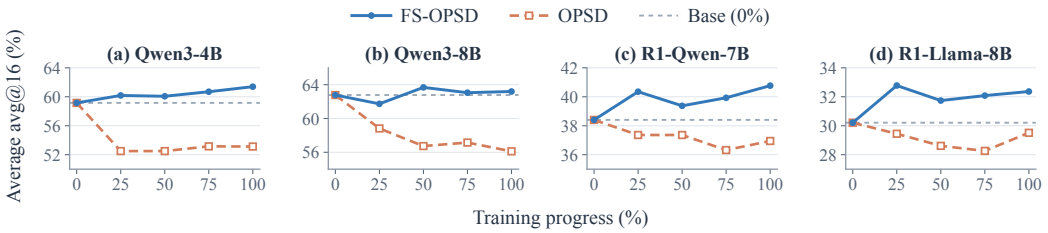


Figure 4: **Training dynamics.** Mean avg@16 across the three benchmarks.

(1) Standard OPSD does not reliably improve thinking models. Consistent with the analysis in Section 3.2, standard OPSD lowers average accuracy on all five models. The results reveal that uniformly applying privileged supervision regardless of the student's state is unreliable even when the reference solution is correct.

(2) FS-OPSD consistently improves average performance over the Base model. FS-OPSD achieves better performance than Base on 14 of 15 model-benchmark pairs and improves the average for all five models. The gains span Qwen3, DeepSeek-R1-Distill, and GLM-Z1, supporting generalization across architectures.

(3) The gap between standard OPSD and FS-OPSD is large and consistent. FS-OPSD beats standard OPSD on all 15 model-benchmark pairs, with gains of up to 8.26 points in average accuracy, supporting state-dependent rather than uniform privileged supervision.

4.3 Training Dynamics

While Table 1 reports final performance, Figure 4 traces performance throughout full training, contrasting the degradation of vanilla OPSD with the stability of FS-OPSD. **(1) The performance gap emerges early and persists.** FS-OPSD outperforms OPSD at every checkpoint across all four models and OPSD falls below Base by 25% of training and remains below it thereafter. **(2) FS-OPSD is robust to checkpoint selection.** Across all four models, almost all checkpoints stably outperform both Base and OPSD, indicating that gains do not depend on selecting a favorable checkpoint.

4.4 Ablation Studies

In this section, we conduct extensive ablation studies to study key design choices in FS-OPSD, including (1) the effect of supervision sources, (2) comparison of distillation objectives, and (3) sensitivity to hyperparameters.

4.4.1 Effect of Supervision Sources

A key design choice in FS-OPSD is to use different supervision sources at fork and lock states. To examine state-dependent supervision, we compare FS-OPSD with four variants: (1) *Fork-q_z*: replaces q_0 with q_z only at fork states. (2) *All-q_0*: uses q_0 in all regimes. (3) *Fork-Adaptive*: uses q_0 at forks and q_z at neutral and lock states. (4) *Random Routing*: shuffles regime labels within each sequence at fixed regime counts.

Table 2: **Supervision-source ablations on Qwen3-4B (avg@16, %).** Bold marks column bests.

Method	AIME 2024	AIME 2025	HMMT 2025	Average
Fork-q_z	**75.00**	65.42	40.63	60.35 ↓1.04
All-q_0	72.91	65.63	42.29	60.28 ↓1.11
Fork-Adaptive	72.71	62.71	40.63	58.68 ↓2.71
Random Routing	73.33	61.88	36.25	57.15 ↓4.24
FS-OPSD	74.58	**66.46**	**43.13**	**61.39**

All four variants lower average accuracy relative to FS-OPSD (Table 2), with Random Routing yielding the largest drop of 4.24 points despite preserving per-sequence treatment counts. Figure 3(b) provides a complementary diagnostic: lock-only privilege admission covers 69.3% of positions while admitting only 6.5% of all q_0–q_z top-1 disagreements. These results support restricting privileged supervision to locks and aligning supervision and target reshaping with the student's state.

4.4.2 COMPARISON OF DISTILLATION OBJECTIVES

Another design choice is the divergence used to measure the student–teacher gap. We compare JSD, forward KL and reverse KL on Qwen3-4B in Table 3. All three reported objective configurations improve average performance over Base, suggesting that FS-OPSD's gains can extend across different distillation objectives. Across all configurations, JSD performs best on average, with forward KL slightly better on AIME 2024. We therefore adopt JSD in all remaining experiments.

Table 3: **Divergence configurations (avg@16, %).** Bold marks the best score in each column.

Objective	AIME 2024	AIME 2025	HMMT 2025	Average
forward KL	**75.21**	62.08	42.92	60.07
reverse KL	74.58	64.58	42.29	60.48
JSD	74.58	**66.46**	**43.13**	**61.39**

4.4.3 SENSITIVITY TO τ_{lock} AND T_f

Table 4 varies τ_{lock} with $T_f = 1.5$, and varies T_f with $\tau_{\text{lock}} = 0.22$. All variants use $\tau_{\text{fork}} = 0.75$, default source-selection rule and JSD objective.

Consistent with our derivation in Appendix B.2, $\tau_{\text{lock}} = 0.22$ achieves the highest average accuracy among the tested settings (Table 4). It also minimizes the calibration residual across all three trace pools in Figure 3(a). This agreement in derivation, calibration and performance supports our choice.

T_f. The unsoftened target underperforms on all three benchmarks, indicating that retaining q_0 at forks alone does not recover the full method's benefit. Softening narrows preference gaps without changing the candidate ranking, consistent with the intended role of maintaining competition among retained candidates.

Table 4: **Hyperparameter sensitivity (avg@16, %).** Bold marks the best score.

Setting	AIME 2024	AIME 2025	HMMT 2025	Average
τ_{lock}, with $T_f = 1.5$				
0.18	73.54	64.58	42.92	60.35 ↓1.04
0.20	74.17	65.63	40.63	60.14 ↓1.25
0.22 (default)	**74.58**	**66.46**	**43.13**	**61.39**
0.24	**74.58**	63.33	42.92	60.28 ↓1.11
0.26	74.38	65.00	40.63	60.00 ↓1.39
Fork temperature T_f, with $\tau_{\text{lock}} = 0.22$				
1.0	72.71	65.00	42.50	60.07 ↓1.32
1.5 (default)	**74.58**	**66.46**	43.13	**61.39**
1.8	73.54	66.04	**43.75**	61.11 ↓0.28

5 CONCLUSION

We introduced Fork-Sensitive On-Policy Self-Distillation (FS-OPSD), a framework that adapts privileged supervision to the uncertainty of the current reasoning state. Our analysis reveals that solution conditioning frequently replaces the solution-free teacher's preferred continuation at high-entropy forks, but usually preserves it at low-entropy locks. Motivated by this distinction, FS-OPSD combines state-dependent source selection with target reshaping to maintain competition among candidate continuations at forks while retaining privileged guidance at locks. Experiments across five thinking models and three mathematical reasoning benchmarks show consistent improvements over vanilla OPSD and improved average accuracy over all five base models, with gains of up to 8.26% and 2.36% in average accuracy. These findings suggest that knowing when to withhold privileged information is an important part of effective self-distillation for thinking models.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of mathematical proofs. Additionally, we used generative AI tools to modify scientific figures, create and edit software code, search for and retrieve information, edit the manuscript to improve readability, and format references. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

Our work does not raise ethical concerns.

Reproducibility statement

To facilitate reproducibility, Section 3 and Algorithm 1 describe the FS-OPSD method and training procedure. Section 4.1 identifies the models and dataset sources used in our experiments. Appendix D provides the training hyperparameters, random seed, hardware configuration, prompt construction, and evaluation and answer-scoring protocols. The derivations underlying the state thresholds and target reshaping are provided in Appendices B.1, B.2, and C. And our code is available at `https://anonymous.4open.science/r/FS-OPSD-14A3/README.md`.

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*, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Ashkchamp. Openthoughts_math_filtered_30K. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/Ashkchamp/Openthoughts_math_filtered_30K.

Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation. In *International Conference on Learning Representations*, pp. 17134–17167, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2b6a8ca3d06ffa2e044bff8c723863ff-Abstract-Conference.html.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30377–30385, 2026. doi: 10.1609/aaai.v40i36.40290. URL https://ojs.aaai.org/index.php/AAAI/article/view/40290.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026. URL https://arxiv.org/abs/2605.00674. HMMT February 2025 dataset: https://huggingface.co/datasets/MathArena/hmmt_feb_2025.

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

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=QGJ9ttXLTy.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. URL https://arxiv.org/abs/2406.12793.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng,

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

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026. URL https://arxiv.org/abs/2604.12002.

John Hewitt, Christopher Manning, and Percy Liang. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427, 2022. doi: 10.18653/v1/2022.findings-emnlp.249. URL https://aclanthology.org/2022.findings-emnlp.249/.

Hugging Face H4. AIME 2024. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024.

Harold Jeffreys. *Theory of Probability*. Oxford University Press, 3rd edition, 1961.

Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026. URL https://arxiv.org/abs/2607.05184.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026. URL https://arxiv.org/abs/2603.24472.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

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*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang 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.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Liu, 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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37906–37918. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25j.html.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, 2025. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Keqin Peng, Liang Ding, Yuanxin Ouyang, Meng Fang, and Dacheng Tao. Revisiting overthinking in long chain-of-thought from the perspective of self-doubt. *arXiv preprint arXiv:2505.23480*, 2025. URL https://arxiv.org/abs/2505.23480.

Keqin Peng, Chen Li, Yuanxin Ouyang, Yancheng Yuan, and Liang Ding. Diagnosing and mitigating thinking collapse in on-policy self-distillation. *arXiv preprint arXiv:2607.10805*, 2026. URL https://arxiv.org/abs/2607.10805.

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, Gang Chen, and Jieping Ye. Purified opsd: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026. URL https://arxiv.org/abs/2607.02234.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. URL https://arxiv.org/abs/2601.19897.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*, 2025. URL https://openreview.net/forum?id=OwhVWNOBcz.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 115452–115486, 2025. doi: 10.52202/085713-3850. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a797c2d2e0c1fdabf4d1ab8cd0b465c6-Abstract-Conference.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Yuchen Wu, Liang Ding, Li Shen, and Dacheng Tao. Reason-ke++: Aligning the process, not just the outcome, for faithful llm knowledge editing. *arXiv preprint arXiv:2511.12661*, 2025. URL https://arxiv.org/abs/2511.12661.

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

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026. URL https://arxiv.org/abs/2604.01193.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

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.

A A MATCHED-PREFIX ILLUSTRATION OF SOLUTION-CONDITIONED REASONING

We extend the fixed-prefix diagnostic with a qualitative comparison of continuations. From the existing 60 Qwen3-4B trajectories, we select ten fork states at readable reasoning boundaries with changes in the archived leading candidate; selection is completed before generating the new continuations. The example below is selected for illustration and does not estimate the prevalence or usefulness of either behavior.

Shared prefix and local preferences. The problem asks for the minimum of

$$M(x) = \sqrt{x^2 - 10x + 26} + \sqrt{x^2 - 18x + 85}.$$

After completing the square, the shared prefix ends with:

> "So, if I have two such terms, then M(x) is the sum of distances from (x, 0) to (5, 1) and to (9, 2). Is that right?"

At this state, after 442 generated tokens, the scoring gives $H_3(q_0) = 0.9083$ nats, above 0.75. The unprivileged continuation begins "Wait, let me check," whereas the privileged continuation begins "Yes, because". Table 5 reports the corresponding next-token probabilities.

Table 5: Next-token probabilities at the shared prefix in the continuation run, before target reshaping and without renormalizing over the displayed candidates.

Candidate	q_0, problem only	q_z, full reference
`Wait`	63.419%	0.090%
`Yes`	18.170%	99.207%
`Let`	18.170%	0.020%

Different uses of verification. Both continuations apply geometric reflection, obtain a minimum value of 5, and proceed to check it with calculus. The unprivileged continuation starts deriving the stationarity condition:

> "But let me check with calculus to make sure. Maybe I can take the derivative of M(x) and find its minimum."

The privileged continuation instead introduces a candidate minimizer when checking that condition:

> "However, since we know from the geometric approach that the minimum occurs at x = 19/3, let me check if that satisfies the equation."

The value $19/3$ is derived in the supplied reference solution but has not been derived in the shared prefix or the preceding privileged continuation. Thus, the visible difference concerns constructing a candidate versus checking a candidate available in the reference; privileged access does not eliminate verification in this example. The later excerpts come from different continuation positions, while the probabilities in Table 5 refer only to the initial shared state. Both continuations reach the length limit, so this comparison supports an illustration of conditional reasoning preferences, without establishing a correctness advantage or the downstream value of retaining q_0.

B THEORETICAL DERIVATION FOR τ_{fork} AND τ_{lock}

In this section, we give a theoretical derivation for values that can distinguish fork and lock states (namely τ_{fork} and τ_{lock}) used in FS-OPSD.

B.1 FORK

For completeness, we derive the effective-candidate interpretation used by both values. Fix a state s_t, and let $v_{(1)}, v_{(2)}, v_{(3)}$ be the three highest-probability tokens under $q_0(\cdot \mid s_t)$, with $p_{(1)} \geq p_{(2)} \geq p_{(3)}$ and $p_{(i)} = q_0(v_{(i)} \mid s_t)$. Let $S_3 = \sum_{i=1}^{3} p_{(i)}$ be their total mass and define the renormalized top-three distribution

$$q_i = \frac{p_{(i)}}{S_3}, \qquad i \in \{1, 2, 3\}, \qquad \sum_{i=1}^{3} q_i = 1. \tag{17}$$

Thus, the entropy below is computed on the selected top-three set rather than on the full vocabulary. We write

$$H_3(q) = -\sum_{i=1}^{3} q_i \ln q_i, \qquad 0 \leq H_3(q) \leq \ln 3, \tag{18}$$

where the logarithm is natural. In this appendix, $H_3(q_0)$ denotes this quantity for the top-three distribution induced by q_0.

To interpret this entropy as a number of candidates, consider a uniform distribution U_m over m equiprobable candidates. Its entropy is

$$H(U_m) = -\sum_{j=1}^{m} \frac{1}{m} \ln \frac{1}{m} = \ln m. \tag{19}$$

We define the effective number of candidates as the possibly non-integer size of an equiprobable distribution with the same entropy as q. Equating its entropy with $H_3(q)$ gives

$$\ln N_{\text{eff}}(q) = H_3(q), \qquad N_{\text{eff}}(q) = \exp(H_3(q)). \tag{20}$$

For a generic finite distribution u, we use the analogous notation $N_{\text{eff}}(u) = \exp(H(u))$. For example, a distribution with $H_3(q) = \ln 2$ has the same entropy as two equally likely candidates, even though its three probabilities need not be exactly $(1/2, 1/2, 0)$.

This effective number is multiplicative under independent composition, which gives it a useful consistency property. Let $q = (q_i)_{i=1}^{m}$ and $r = (r_j)_{j=1}^{n}$ be two independent distributions, and let their product distribution be $(q \otimes r)_{ij} = q_i r_j$. Then

$$\begin{aligned} H(q \otimes r) &= -\sum_{i,j} q_i r_j \ln(q_i r_j) \\ &= -\sum_{i,j} q_i r_j \big(\ln q_i + \ln r_j\big) \\ &= -\sum_{i} q_i \ln q_i - \sum_{j} r_j \ln r_j = H(q) + H(r), \end{aligned} \tag{21}$$

$$N_{\text{eff}}(q \otimes r) = \exp(H(q \otimes r)) = \exp(H(q)) \exp(H(r)) = N_{\text{eff}}(q) N_{\text{eff}}(r). \tag{22}$$

The product identity is an interpretation under independent composition; it does not assume that successive tokens in a reasoning trajectory are independent.

The same entropy can also be related explicitly to the top-1 probability. Write $q_1 = p$, $q_2 = a$, and $q_3 = b$, so that $a + b = 1 - p$. When $p < 1$, define the conditional split within the remaining two candidates by

$$r = \frac{a}{1-p}, \qquad 1 - r = \frac{b}{1-p}, \qquad a = (1-p)r, \quad b = (1-p)(1-r). \tag{23}$$

Using the binary entropy $h_{\mathrm{b}}(u) = -u \ln u - (1-u)\ln(1-u)$, we obtain

$$\begin{aligned} H_3(q) &= -p \ln p - (1-p) r \ln\big((1-p)r\big) - (1-p)(1-r)\ln\big((1-p)(1-r)\big) \\ &= \underbrace{-p \ln p - (1-p)\ln(1-p)}_{h_{\mathrm{b}}(p)} + (1-p)\underbrace{\big[-r \ln r - (1-r)\ln(1-r)\big]}_{h_{\mathrm{b}}(r)} \\ &= h_{\mathrm{b}}(p) + (1-p) h_{\mathrm{b}}(r). \end{aligned} \tag{24}$$

The case $p = 1$ follows by continuity, with the second term equal to zero. Since $0 \le h_{\mathrm{b}}(r) \le \ln 2$, this decomposition yields the bounds

$$h_{\mathrm{b}}(p) \le H_3(q) \le h_{\mathrm{b}}(p) + (1-p)\ln 2. \tag{25}$$

Thus, top-1 concentration controls the binary part of the uncertainty, while the remaining term records how evenly the residual mass is divided between the second- and third-ranked candidates.

The fork requirement is that there be more than two effective candidates. From Equation 20, this is

$$N_{\mathrm{eff}}(q) > 2 \quad \Longleftrightarrow \quad H_3(q) > \ln 2 \approx 0.693. \tag{26}$$

We choose $\tau_{\mathrm{fork}} = 0.75$, a fixed operating point slightly above this boundary. Every state satisfying $H_3(q_0) \ge 0.75$ then has $N_{\mathrm{eff}}(q_0) \ge \exp(0.75) \approx 2.117 > 2$.

B.2 Lock

We expand the proof sketch in Section 3.3 by connecting candidate dominance to the entropy bounds in Equation 25. Within the renormalized top-three distribution, compare the leading candidate $v_{(1)}$ with the composite alternative $\{v_{(2)}, v_{(3)}\}$. Their model probability ratio is $p/(1-p)$, giving the log-odds measure

$$W(q) = \ln \frac{p}{1-p} = \mathrm{logit}(p). \tag{27}$$

In Bayesian hypothesis testing, posterior odds equal the Bayes factor only when the two hypotheses have equal prior odds (Jeffreys, 1961; Kass & Raftery, 1995). The modified evidence scale of Kass & Raftery (1995) places the boundary for strong evidence at $2 \ln B = 6$. We use this scale to motivate the model-concentration criterion $W(q) \ge 3$, corresponding to odds of at least $e^3 \approx 20.086$, without assuming that token probabilities are calibrated Bayesian posterior probabilities. Inverting the logit function gives the critical leading probability:

$$p^* = \sigma(3.0) = \frac{1}{1+e^{-3}} = \frac{e^3}{1+e^3} \approx 0.9526, \tag{28}$$

with residual probability mass $1 - p^* = \frac{1}{1+e^3} \approx 0.0474$. The leading-token binary entropy evaluated at this critical probability is

$$h_{\mathrm{b}}(p^*) = -p^* \ln p^* - (1-p^*)\ln(1-p^*) = \ln\big(1+e^{-3}\big) + 3(1-p^*) \approx 0.1909. \tag{29}$$

For fixed p, the upper bound in Equation 25 is attained when the two residual probabilities are equal. Let $f(p) = h_{\mathrm{b}}(p) + (1-p)\ln 2$. Its derivative is $f'(p) = \ln[(1-p)/(2p)] < 0$ for $p > 1/3$, so the local entropy under the dominance condition $p \ge p^*$ satisfies

$$H_3(q) \le h_{\mathrm{b}}(p^*) + (1-p^*)\ln 2 \approx 0.1909 + 0.0474 \ln 2 \approx 0.2237. \tag{30}$$

This derivation supplies an entropy-scale reference of 0.2237, close to the operating threshold $\tau_{\mathrm{lock}} = 0.22$. The implication is one-way: $p \ge p^*$ implies the entropy upper bound, but $H_3(q_0) \le 0.22$ does not by itself guarantee odds exceeding 20:1, because the residual mass can be distributed unequally. The selected log-odds criterion therefore motivates the threshold's scale rather than uniquely determining an optimal gate. We set the operating value using the empirical change-point minimum $\tau = 0.22$ of $S(\tau)$ in Equation 12 across all three rollout pools in Figure 3(a).

C Local Properties of Target Reshaping

The distinction between retained support and relative weights within that support follows the analysis of SSD by Zhang et al. (2026, Appendix B.2–B.4). Here we specialize the temperature calculation to the selected FS-OPSD source and derive the corresponding local tail-suppression gradient for symmetric JSD.

Temperature within the retained set. Fix a student-visited prefix and write $q = q_t^{\mathrm{src}}$, with $q(v) > 0$ for all vocabulary tokens. For any $\tau > 0$, the map $q(v) \mapsto q(v)^{1/\tau}$ preserves rankings, so $\mathcal{K} = \mathrm{TopK}(q, k)$ is independent of temperature when ties are broken by a fixed rule. Let $Z(\tau) = \sum_{u \in \mathcal{K}} q(u)^{1/\tau}$ and define $q_\tau(v) = q(v)^{1/\tau}/Z(\tau)$ on $\mathcal{K}$, with zero probability outside it. For $u, v \in \mathcal{K}$, the log-probability ratio is $\log[q_\tau(u)/q_\tau(v)] = \tau^{-1} \log[q(u)/q(v)]$. Thus, increasing temperature reduces preference gaps within the same retained set. Its entropy satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\tau} H(q_\tau) = \frac{\mathrm{Var}_{v \sim q_\tau}[\log q(v)]}{\tau^3} \geq 0. \tag{31}$$

To obtain this identity, write $L(v) = \log q(v)$. Differentiation gives $\partial_\tau q_\tau(v) = q_\tau(v)(\mathbb{E}_{q_\tau}[L] - L(v))/\tau^2$; substituting this into $H(q_\tau) = \log Z(\tau) - \mathbb{E}_{q_\tau}[L]/\tau$ yields Equation 31. In particular, $H(q_{T_f}) \geq H(q_1)$ for $T_f > 1$, with strict inequality unless q is uniform on $\mathcal{K}$. Fork softening therefore increases entropy relative to the unit-temperature target on the retained set while keeping the excluded tail at zero target mass.

Tail suppression under JSD. At the same fixed prefix, write the student distribution as $p = \mathrm{softmax}(\ell)$, the stop-gradient target as $Q = q_t^\star$, and their mixture as $m = (p + Q)/2$. Treat ℓ as a vector of independent local logits. Differentiating the symmetric JSD gives $\partial\, \mathrm{JSD}(Q\|p)/\partial p(v) = \frac{1}{2} \log[p(v)/m(v)]$. The softmax chain rule subtracts its p-weighted mean, $\frac{1}{2}\, \mathrm{KL}(p\|m)$. For an excluded token $v \notin \mathcal{K}$, we have $Q(v) = 0$ and $p(v)/m(v) = 2$, giving

$$\frac{\partial\, \mathrm{JSD}(Q\|p)}{\partial \ell(v)} = \frac{p(v)}{2} \left[\log 2 - \mathrm{KL}(p\|m)\right] > 0, \qquad v \notin \mathcal{K}. \tag{32}$$

For finite student logits, $p(v) > 0$ everywhere. Since $m(v) \geq p(v)/2$, with strict inequality on the nonempty retained set, $\mathrm{KL}(p\|m) < \log 2$. Equation 32 therefore shows that a gradient-descent step in the local logits lowers every excluded-token logit. Together, these properties explain how the FS-OPSD target combines more balanced fork candidates with local suppression of the discarded tail.

D Additional Experimental Details

Training data and information access. All distillation methods use the same fixed 15,000-example subset of `Openthoughts_math_filtered_30K`, consisting of mathematics problems paired with worked solutions. We take the first 15,000 entries in the source data order and shuffle this subset with seed 42 before training. At each update, the current student generates a fresh reasoning trajectory conditioned on the problem. The self-teacher is initialized from the same checkpoint as the student and remains frozen throughout training. For FS-OPSD, the two teacher conditions score the same student-generated prefixes: q_0 receives the problem and prefix, while q_z additionally receives the worked solution. The Base configuration is evaluated without additional training.

Training settings. Table 6 summarizes the shared training configuration. OPSD and FS-OPSD use symmetric JSD with equal mixture weights $\alpha = 0.5$; teacher targets are stop-gradient, and only the student is updated.

Table 6: **Training settings.** The learning rate for Qwen3-8B is the stated exception to the default.

Setting	Value
Training examples	15,000
Training epochs	1
Optimizer updates (four main models)	234
Per-device batch size	1
Gradient accumulation steps	16
Effective batch size	$4 \times 1 \times 16 = 64$
Maximum training prompt length	25,000 tokens
Maximum training completion length	4,096 tokens
Default learning rate	5×10^{-6}
Learning rate for Qwen3-8B	2×10^{-6}
Optimizer	Fused AdamW
Adam betas / epsilon	$(0.9, 0.999)$ / 10^{-8}
Weight decay	0
Learning-rate schedule	Cosine
Warmup ratio	0.03
Maximum gradient norm	1.0
Rollouts per problem per update	1
Rollout temperature / top-k / top-p	1.0 / 50 / 1.0
Training seed	42
Precision	bfloat16
Gradient checkpointing	Enabled
Hardware	Four NVIDIA H100 GPUs

FS-OPSD targets. Table 7 collects entropy range, target sources, and target temperatures. The regime statistic $H_3(q_0)$ is computed after renormalizing the three leading q_0 probabilities, using natural logarithms. After applying the listed target temperature, FS-OPSD retains the top $k = 20$ tokens and renormalizes in every regime. These temperatures reshape the teacher distribution during distillation; evaluation sampling uses the settings given below.

Table 7: **FS-OPSD target settings.** The entropy range, target temperatures are fixed across models.

Regime	Entropy range	Source	Temperature
Fork	$H_3(q_0) \geq 0.75$	q_0	1.5
Neutral	$0.22 < H_3(q_0) < 0.75$	q_0	1
Lock	$H_3(q_0) \leq 0.22$	q_z	1

Routing comparisons. Vanilla OPSD uses unreshaped q_z throughout. All-q_0 retains FS-OPSD's partition, temperatures, top-k truncation, and JSD, but uses q_0 in every regime. Fork-Adaptive uses q_0 at forks and q_z elsewhere. Random Routing permutes treatment labels across valid positions within each sequence, preserving their counts, sources, and reshaping assignments.

Evaluation and aggregation. All models are evaluated in thinking mode on AIME 2024, AIME 2025, and HMMT 2025, with 30 problems per benchmark. We generate 16 responses per problem with vLLM (Kwon et al., 2023), using temperature 0.6, top-$p = 0.95$, and a maximum generation length of 38,912 tokens. Evaluation uses four single-GPU shards with engine seed 0 and top-k filtering disabled. Let $c_i \in \{0, \ldots, 16\}$ be the number of correct responses for problem i. The reported percentage-valued avg@16 is

$$\text{avg@16} = \frac{100}{30} \sum_{i=1}^{30} \frac{c_i}{16}.$$

Each benchmark therefore contains 480 evaluated responses per model–method configuration. The *Average* column in the result tables is the unweighted mean of the three benchmark scores.

Training prompts and context construction. Each training example supplies a problem and a worked solution. The problem text ends with `Return your final response within \boxed{}`. The student and q_0 receive this problem text as a single user message. For q_z, the user message is constructed with the following template, where `{problem}` includes that instruction and `{solution}` is the supplied worked solution:

```
{problem}

This is an example for a response to the question:
{solution}

Now answer with a response of your own, including the thinking process.
```

Both conditions use the checkpoint's native chat template with an assistant-generation prompt. We add no system message or extra thinking prefix. Each formatted training prompt is left-truncated to at most 25,000 tokens before the student-generated completion tokens are appended. The worked solution is subject to this combined prompt limit, without a separate solution-length cap. The two teacher conditions receive exactly the same completion tokens, so their scores correspond to the same student prefix despite their different prompt lengths.

Training implementation. We fine-tune all student parameters using bfloat16 computation, fully sharded data parallelism with full sharding and automatic module wrapping, and gradient checkpointing. The teacher remains frozen, with synchronization and exponential-moving-average updates disabled; no auxiliary KL-to-reference penalty is applied. Distillation divergences are evaluated over the full vocabulary, using token chunks to reduce memory consumption. Padding and the first three completion tokens are excluded from the distillation loss. Completions that reach the training length limit remain in the loss on their otherwise valid positions. For the forward- and reverse-KL ablations, we smooth the final target as $q^{\star}_{t,\epsilon}(v) = (1-\epsilon)q^{\star}_{t}(v) + \epsilon/|\mathcal{V}|$ with $\epsilon = 10^{-4}$, ensuring finite reverse KL after top-k truncation, whereas JSD uses no uniform target smoothing.

Rollout generation and importance correction. Training uses colocated vLLM engines with one engine per GPU, tensor parallel size 1, and engine seeds 0, 1, 2, and 3. We enable vLLM importance-sampling correction with an upper cap of 2.0. For each generated token, we compute the ratio of its probability under the trainer's pre-update student to its probability reported by the vLLM sampler and clip this ratio at 2.0. The clipped ratios are averaged over the completion positions included in the loss, and this sequence-level weight multiplies the sequence's mean token-wise distillation loss.

Evaluation prompts. Each benchmark problem is supplied as a single user message using the evaluated checkpoint's native chat template and assistant-generation prompt. For AIME 2024 and AIME 2025, the stored problem text includes the instruction `Please reason step by step, and put your final answer within \boxed{}.` HMMT 2025 uses the stored problem statement without this appended instruction. Evaluation enables thinking for Qwen3. The DeepSeek-R1-Distill chat templates supply their native `<think>` opening; we do not append an additional reasoning marker. Generation uses vLLM with the sampling settings reported above.

Answer extraction and scoring. We use a rule-based mathematical answer extractor and equivalence checker adapted from RLVR-Decomposed.[1] The extractor first takes the last boxed answer in the generated response. If no boxed answer is present, it looks for the text after the last occurrence of `the answer is`, falling back to the last numeric string when that phrase is absent. The extracted text is restricted to its first line and normalized for whitespace and LaTeX formatting. Correctness is determined by normalized string matching, numerical comparison with relative tolerance 10^{-4} and the checker's percentage-equivalence handling, or symbolic equivalence using SymPy. The final symbolic-equivalence check has a one-second timeout; unsuccessful extraction or equivalence checks are scored as incorrect. For HMMT, reference answers are normalized directly from their supplied LaTeX, and comma-separated multiple answers are compared by one-to-one matching without regard to order.

[1] `https://github.com/TianHongZXY/RLVR-Decomposed`

E Additional Results

Table 8 reports additional results. All settings use the same evaluation protocol as Table 1.

Table 8: **Additional results (avg@16, %).** Thinking mode is enabled for all models. Average is the unweighted mean across benchmarks, and bolded scores indicate the best-performing method for each model and metric.

Models	Method	AIME 2024	AIME 2025	HMMT 2025	Average
Qwen3-0.6B	Base	9.17	**15.83**	**10.21**	11.74
	OPSD	10.62	13.54	9.17	11.11
	FS-OPSD (Ours)	**11.46**	15.63	8.75	**11.94**
Qwen3-1.7B	Base	45.62	**36.25**	**23.95**	35.27
	OPSD	41.67	30.00	19.58	30.42
	FS-OPSD (Ours)	**49.38**	35.00	23.54	**35.97**
DS-R1-Distill-Qwen-1.5B	Base	**30.63**	21.67	14.17	**22.15**
	OPSD	29.58	20.83	12.92	21.11
	FS-OPSD (Ours)	26.88	**24.17**	**14.79**	21.94

Small-scale behavior. FS-OPSD improves average accuracy over Base by 0.20 and 0.70 points on Qwen3-0.6B and Qwen3-1.7B, respectively, but the gains are concentrated on AIME 2024 and do not transfer uniformly to the other benchmarks. On DeepSeek-R1-Distill-Qwen-1.5B, FS-OPSD improves AIME 2025 and HMMT 2025 but decreases AIME 2024, leaving average accuracy 0.21 points below Base.