# Trajectory-Level Routing for On-Policy Self-Distillation

**Anonymous ACL submission**


## Abstract

On-Policy Self-Distillation (OPSD) improves reasoning models by aligning a student to a privileged version of itself on the student's own rollouts, but standard OPSD treats all rollouts as equally useful for distillation. We argue that this is suboptimal: because teacher and student are derived from the same model lineage, corrective signal varies substantially across trajectories, making selection a central problem rather than a minor efficiency trick. We formulate this problem as **trajectory-level routing**, which asks which complete rollouts should be retained for alignment. Our analysis shows that rollout value is highly non-uniform and is associated with student uncertainty, teacher–student disagreement, and response length. Based on these findings, we propose **Selective OPSD**, a simple routing framework that supports both budget-aware batch selection and Best-of-$N$ routing before teacher forward. Experiments across multiple reasoning benchmarks show consistent improvements over vanilla OPSD, while teacher-free routing can reduce privileged teacher forward passes in batch-selection settings. These results suggest that selective alignment is a key ingredient of effective OPSD.


## 1 Introduction

Large language models (LLMs) have shown rapidly improving capabilities in mathematical reasoning, code generation, and complex decision-making (Achiam et al., 2023; Chen et al., 2021; Wei et al., 2022; Shao et al., 2024), but such capabilities often come with substantial training and deployment costs. Knowledge distillation has therefore become a central technique for model compression and capability transfer (Hinton et al., 2015; Gou et al., 2021; Sanh et al., 2019). Conventional distillation typically trains the student on teacher-provided data or distributions, which may create a mismatch between the student's training distribution and its own inference-time behavior (Kim and Rush, 2016). On-Policy Distillation (OPD) addresses this mismatch by letting the student learn from teacher feedback on the student's own generated rollouts. Within this paradigm, GKD (Agarwal et al., 2024), MiniLLM (Gu et al., 2024), DistiLLM (Ko et al., 2024), DistiLLM-2 (Ko et al.), and REOPOLD (Ko et al., 2026) have improved on-policy distillation from the perspectives of loss design, training stability, and token-level weighting.

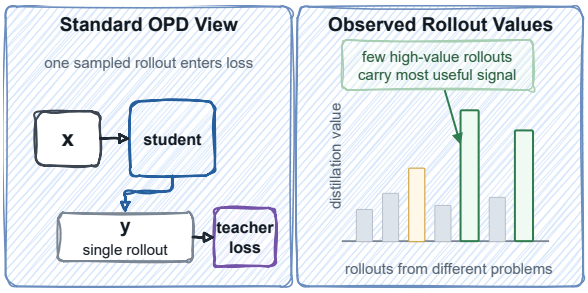


Figure 1: Non-uniform rollout value in on-policy distillation motivates selecting which student-generated trajectories should be aligned.

OPSD (Zhao et al.) pushes this idea into a self-contained setting: teacher and student share the same model lineage and base parameters, while the teacher additionally observes privileged information such as the ground-truth answer. This allows a model to distill from a stronger conditioned version of itself without relying on a separate external teacher. Recent OPSD variants have further extended the source and use of privileged context, including concise reasoning instructions, binary rewards, consensus signals, online experience, and contextual knowledge (Sang et al., 2026; He et al., 2026; Stein et al., 2026; Ye et al., 2026a,b; Wang et al., 2026). These developments make OPSD an increasingly attractive framework for improving reasoning models under limited supervision.

However, precisely because the teacher is a priv-

ileged version derived from the same base model rather than an independently trained external expert, OPSD exposes a structural tension. As a result, the corrective signal available from the teacher can vary substantially across student rollouts. On some trajectories, the student is already close to the teacher's local mode, so further alignment provides little new information; on others, teacher–student disagreement is large and contains a meaningful corrective signal. This concern is aligned with recent analyses showing that self-distillation and privileged supervision can sometimes suppress useful uncertainty expression and degrade reasoning under distribution shift (Kim et al., 2026; Kadavath et al., 2022; Lin et al.). We therefore argue that a central question in OPSD is not only *how* to align the student with the teacher, but *which rollouts should be aligned in the first place*. Figure 1 summarizes this motivation in the broader OPD setting: although standard on-policy distillation often samples one rollout per problem, the value of these rollouts is highly non-uniform across the training stream, so indiscriminate alignment may dilute useful corrective signal.

DistiLLM-2 (Ko et al.), Entropy-Aware OPD (Jin et al., 2026), TIP (Xu et al., 2026b), and FastOPD (Zhang et al., 2026) identify useful regions *within* a trajectory and adjust token-level supervision accordingly. Our setting instead concerns selection *between* complete trajectories. When multiple candidate rollouts are generated for the same problem, as in multi-sample reasoning and self-consistency (Wang et al., 2022), or when only part of a batch can be retained under a teacher-compute budget, the training procedure must make an exclusive decision: which trajectory should be kept, and which should be discarded? We call this problem **trajectory-level routing**. Because a routing error commits the full noise of an entire trajectory to training, rather than merely mis-weighting a few tokens, trajectory-level routing requires a different notion of distillation value. Figure 2 illustrates this shift from indiscriminate alignment in standard OPSD to selective alignment through trajectory-level routing.

We study trajectory value through a diagnostic-to-method pipeline. Entropy first reveals that rollout value is highly non-uniform; teacher–student divergence then adds a direct measure of corrective signal; and response length provides a practical teacher-free proxy for disagreement. These findings lead to **Selective OPSD**, which performs trajectory-level routing through batch filtering and Best-of-$N$ selection under fixed teacher-forward cost.

Our contributions are threefold:

- We formulate **trajectory-level routing** as an explicit OPSD selection problem, distinct from token-level weighting or truncation.
- Diagnostic experiments show that OPSD rollout value is highly non-uniform and can be estimated from entropy, teacher–student disagreement, and response length.
- Our Selective OPSD supports teacher-aware and teacher-free routing, improving Best-of-$N$ distillation under fixed teacher-forward cost.

## 2 Related Work

### 2.1 Distillation, OPD, and OPSD

Knowledge distillation transfers knowledge from a stronger teacher to a student (Hinton et al., 2015; Gou et al., 2021; Sanh et al., 2019; Jiao et al., 2020). In sequence generation, sequence-level distillation improves generation quality but is typically off-policy, relying on teacher-generated outputs (Kim and Rush, 2016). On-policy distillation instead aligns the student on its own rollouts, reducing train–test mismatch. Representative OPD methods include GKD, MiniLLM, DistiLLM, DistiLLM-2, and REOPOLD (Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024; Ko et al., 2026). OPSD (Zhao et al.) further turns this into privileged self-distillation, and subsequent work extends the form of privileged context or gating signal (Sang et al., 2026; He et al., 2026; Stein et al., 2026; Ye et al., 2026a,b; Wang et al., 2026; Shenfeld et al., 2026; Liu et al., 2026). Our work is complementary: rather than redesigning the distillation objective, we ask which student rollouts should enter it.

### 2.2 Selective Distillation and Routing

Recent work increasingly recognizes that distillation signals are not uniformly useful. Token-level methods such as DistiLLM-2, Entropy-Aware OPD, TIP, and FastOPD reweight or skip local regions within a trajectory (Ko et al.; Jin et al., 2026; Xu et al., 2026b; Zhang et al., 2026). Our focus is coarser: selection between complete trajectories. Related work at this level includes PACED, which selects problems near the student's competence frontier, and SCOPE, which routes trajectories by correctness (Xu et al., 2026a; Zheng et al., 2026).

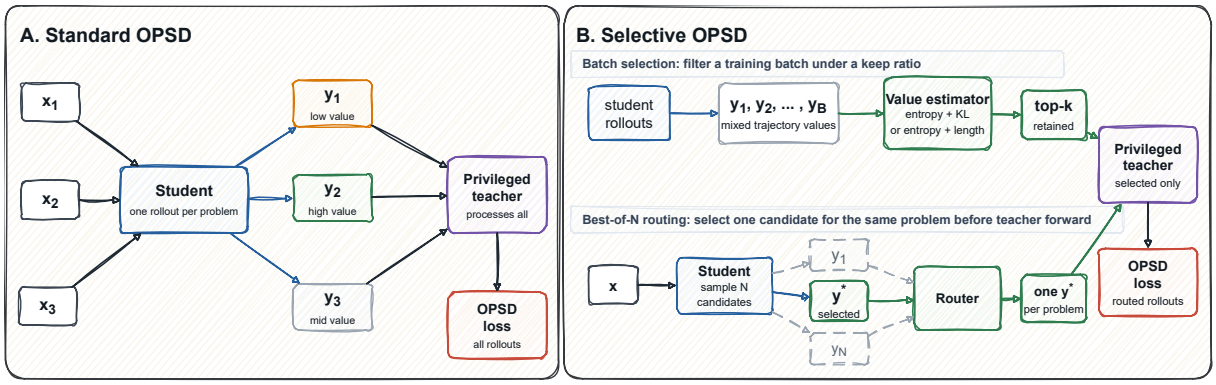


Figure 2: Overview of Selective OPSD. A trajectory-level router selects complete student rollouts either within a batch or among Best-of-$N$ candidates before teacher-side distillation. Under BoN routing, only one selected trajectory per problem is forwarded to the privileged teacher.

More broadly, RLHF, reasoning-oriented RL, and self-consistency also rely on sequence-level differences across sampled rollouts (Ouyang et al., 2022; Shao et al., 2024; Wang et al., 2022). Selective OPSD differs in two ways: it estimates trajectory value from distributional signals rather than external rewards, and in teacher-free routing it must make the decision before teacher forward.

## 3 Background and Problem Formulation

### 3.1 From OPD to OPSD

In on-policy distillation (OPD), the student first samples a rollout $y \sim \pi_\theta(\cdot \mid x)$ for a problem $x$, and the teacher provides distributional supervision on that same rollout. A generic OPD objective is

$$\mathcal{L}_{\text{OPD}} = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[ \frac{1}{|y|} \sum_{t=1}^{|y|} D\Big( \pi_T(\cdot \mid x, y_{<t}) \,\Big\|\, \pi_\theta(\cdot \mid x, y_{<t}) \Big) \right]. \quad (1)$$

OPSD specializes this setup to self-distillation. The teacher and student are derived from the same model lineage, but the teacher additionally observes privileged context $a$ (e.g., the ground-truth answer), yielding $\pi_T(\cdot \mid x, a, y_{<t})$. Thus, OPSD can be viewed as OPD with an information-asymmetric teacher rather than a separately trained external model. Selective OPSD does not modify this underlying objective; it changes which student-generated rollouts are retained for alignment.

### 3.2 Trajectory-Level Routing

Prior selective distillation methods usually operate within a fixed trajectory, for example by reweighting or truncating tokens after a rollout has already

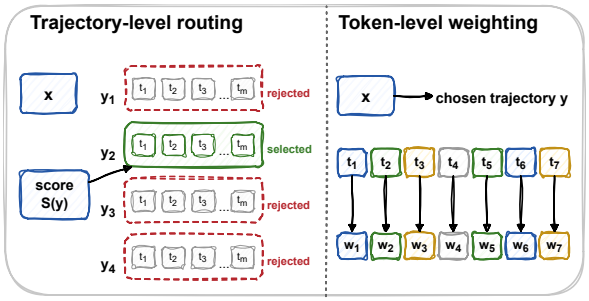


Figure 3: Trajectory-level routing versus token-level weighting. Token-level methods reweight supervision within an already selected trajectory, whereas trajectory-level routing makes an exclusive choice among complete candidate rollouts.

been chosen. Our setting is different. Given a candidate set of rollouts $\mathcal{Y} = \{y_1, \ldots, y_N\}$, the router assigns each trajectory a sequence-level score $S(y_i)$ and retains the selected trajectory or subset:

$$\mathcal{Y}^\star = \text{Select}_{y_i \in \mathcal{Y}}\, S(y_i). \quad (2)$$

For Best-of-N routing, $\mathcal{Y}^\star$ contains a single selected rollout; for batch-level selection, it contains the top-ranked fraction of the batch. Figure 3 illustrates the distinction. Token-level methods adjust supervision inside an already chosen trajectory, whereas trajectory-level routing selects among complete rollouts. This exclusive decision is less fault-tolerant: a poor trajectory choice can commit the noise of an entire rollout to training, and teacher-free routing must make this decision before teacher forward.

## 4 Diagnostic Analysis: Formulating Sequence Distillation Value

### 4.1 Entropy as a Baseline Signal

We begin with the simplest hypothesis inspired by token-level selection: rollouts with higher student

uncertainty may provide more room for teacher correction. For each rollout $y_i$, we compute its average sequence entropy:

$$H_i = \frac{1}{|y_i|} \sum_t H_{\text{student}}(y_{i,t}). \tag{3}$$

We retain the top 50%, 25%, or 12.5% highest-entropy sequences in each batch and compare against random retention at the same budget. Table 1 shows that high-entropy retention preserves much of full OPSD's benefit, while random retention is generally weaker. Entropy is therefore a useful diagnostic signal, but not a complete value estimator: high uncertainty can indicate either useful correction or unstable generation.

### 4.2 From Entropy to Teacher-Aware Sequence Value

Entropy alone measures student uncertainty, but it does not directly measure whether the privileged teacher can provide useful correction. A high-entropy rollout may indicate room for learning, but it may also reflect globally noisy generation. Conversely, a lower-entropy rollout is not necessarily uninformative if the teacher distribution still differs substantially from the student's distribution. We therefore introduce teacher–student divergence as a complementary signal for estimating sequence-level distillation value. Appendix A provides an SNR-style motivation for combining correction magnitude with uncertainty.

To incorporate this complementary signal, we use the average sequence-level teacher–student divergence:

$$K_i = \frac{1}{|y_i|} \sum_t D_{\text{KL}}\Big(\pi_T(\cdot \mid x, a, y_{<t}) \,\Big\|\, \pi_\theta(\cdot \mid x, y_{<t})\Big). \tag{4}$$

From this perspective, teacher–student divergence estimates the magnitude of the correction available from the privileged teacher, while entropy reflects the uncertainty of the student distribution along the rollout. We therefore define a teacher-aware operational score:

$$\text{Score}_{HK}(y_i) = \hat{K}_i(1 - \hat{H}_i), \tag{5}$$

where $\hat{K}_i$ and $\hat{H}_i$ are batch-normalized disagreement and entropy.

$\text{Score}_{HK}$ is an operational value estimator: $\hat{K}_i$ captures teacher correction, while $(1 - \hat{H}_i)$ discounts globally uncertain trajectories. Table 2 shows that this complements entropy-only filtering.

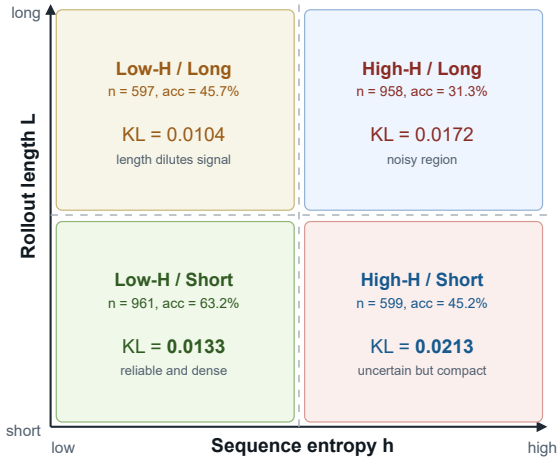


Figure 4: Quadrant analysis by entropy and response length. Shorter trajectories tend to exhibit higher teacher–student divergence within both low-entropy and high-entropy regimes.

### 4.3 Teacher-Free Proxy: Length as a Signal for Disagreement

The limitation of $\text{Score}_{HK}$ is that it requires teacher logits, and therefore cannot be used for pre-selection before teacher forward. To enable teacher-free routing, we ask whether student-side signals can provide information about the missing disagreement term.

We conduct offline quadrant analysis across two datasets (OpenThoughts (Guha et al., 2025) and DAPO (**?**)), partitioning rollouts by entropy and response length. The key empirical observation is consistent: within both high-entropy and low-entropy regimes, shorter trajectories exhibit higher average teacher–student divergence than longer trajectories. This suggests that response length provides information beyond entropy and can serve as a teacher-free proxy for disagreement.

Motivated by the observed inverse trend between response length and average divergence, we use length as a teacher-free proxy for the missing disagreement term. This yields the following ranking signal:

$$V_{\text{TF}}(\tau) \propto \frac{1}{\bar{H}_\tau \cdot L_\tau^2}. \tag{6}$$

For stable batch-wise implementation, we use normalized monotonic surrogates instead of raw reciprocals, yielding the practical teacher-free score:

$$\text{Score}_{LH}(y_i) = (1 - \hat{L}_i)(1 - \hat{H}_i). \tag{7}$$

Here $\hat{L}_i$ is the normalized response length. We do not claim that length is a perfect substitute

| Model | Benchmark | 100% (OPSD) | 50% | | 25% | | 12.5% | |
|---|---|---|---|---|---|---|---|---|
| | | | High-$H$ | Random | High-$H$ | Random | High-$H$ | Random |
| Qwen3-1.7B | AIME24 | 56.94 | 55.83 | 53.61 | 54.72 | 55.00 | 57.22 | 51.94 |
| | AIME25 | 43.05 | 43.33 | 41.94 | 41.67 | 39.72 | 42.78 | 39.72 |
| | HMMT25 | 28.61 | 28.05 | 27.22 | 27.77 | 24.44 | 27.22 | 26.67 |
| Qwen3-4B | AIME24 | 75.55 | 75.83 | 74.72 | 75.28 | 74.17 | 73.89 | 72.78 |
| | AIME25 | 69.72 | 69.44 | 68.33 | 69.72 | 69.17 | 68.33 | 66.67 |
| | HMMT25 | 44.44 | 43.06 | 42.22 | 43.33 | 41.94 | 41.67 | 40.56 |
| Qwen3-8B | AIME24 | 76.67 | 77.78 | 75.00 | 77.22 | 75.83 | 76.11 | 77.78 |
| | AIME25 | 70.56 | 70.83 | 67.22 | 69.72 | 68.61 | 70.83 | 68.33 |
| | HMMT25 | 47.22 | 46.39 | 42.22 | 46.39 | 45.56 | 44.72 | 45.83 |

Table 1: Sequence-level entropy filtering under matched retention budgets. High-$H$ keeps the highest-entropy rollouts, while Random keeps the same fraction uniformly.

| Model | Benchmark | 100% | 50% | 25% | 12.5% |
|---|---|---|---|---|---|
| Qwen3-1.7B | AIME24 | 56.94 | 57.78 | 55.83 | 55.27 |
| | AIME25 | 43.05 | 43.89 | 42.22 | 41.94 |
| | HMMT25 | 28.61 | 28.33 | 27.50 | 25.83 |
| Qwen3-4B | AIME24 | 75.55 | 76.39 | 75.00 | 72.78 |
| | AIME25 | 69.72 | 70.00 | 69.44 | 69.17 |
| | HMMT25 | 44.44 | 43.33 | 43.06 | 42.22 |
| Qwen3-8B | AIME24 | 76.67 | 75.27 | 76.67 | 75.28 |
| | AIME25 | 70.56 | 70.83 | 70.56 | 68.61 |
| | HMMT25 | 47.22 | 47.50 | 46.39 | 47.22 |

Table 2: Teacher-aware batch filtering with $\mathrm{Score}_{HK}$ under different retention budgets. The score combines normalized teacher–student divergence and entropy.

| Model | Benchmark | 100% | 50% | 25% | 12.5% |
|---|---|---|---|---|---|
| Qwen3-1.7B | AIME24 | 56.94 | 58.06 | 53.33 | 56.11 |
| | AIME25 | 43.05 | 42.22 | 40.28 | 38.61 |
| | HMMT25 | 28.61 | 28.33 | 25.56 | 26.11 |
| Qwen3-4B | AIME24 | 75.55 | 76.66 | 74.44 | 73.61 |
| | AIME25 | 69.72 | 70.28 | 68.61 | 67.50 |
| | HMMT25 | 44.44 | 44.72 | 40.83 | 44.44 |
| Qwen3-8B | AIME24 | 76.67 | 78.61 | 77.50 | 76.11 |
| | AIME25 | 70.56 | 71.11 | 68.06 | 68.06 |
| | HMMT25 | 47.22 | 48.61 | 46.67 | 45.28 |

Table 3: Teacher-free batch filtering with $\mathrm{Score}_{LH}$ under different retention budgets. The score uses only normalized response length and entropy.

for teacher disagreement; rather, it is a practical teacher-free proxy that enables routing before teacher forward.

The diagnostic results suggest a simple design principle: a useful OPSD routing score should balance correction magnitude against rollout uncertainty. Entropy provides a student-side uncertainty signal, teacher–student divergence measures the actual corrective signal, and response length offers an additional student-only proxy that correlates with disagreement.

Table 3 reports batch filtering results using $\mathrm{Score}_{LH}$ (`dynamic_lh`). The teacher-free score remains competitive across model scales and retention budgets, suggesting that sequence-level routing can reduce teacher computation while preserving much of the value of full OPSD.

### 4.4 Empirical Validation of Score Dynamics

To confirm that our proposed sequence-value scores meaningfully distinguish trajectories in practice, we visualize the actual routing scores of selected versus discarded rollouts throughout the training process. Figure 5 illustrates this dynamic for the Qwen3-1.7B model at a 50% retention budget.

The visualization reveals a clear, persistent margin between the retained and discarded trajectories across all training steps. This demonstrates that our scoring functions—whether teacher-aware ($\mathrm{Score}_{HK}$) or teacher-free ($\mathrm{Score}_{LH}$)—do not merely make arbitrary splits. Instead, they consistently isolate trajectories with distinct distributional properties (e.g., higher corrective potential and appropriate uncertainty) from less informative ones. For a comprehensive analysis of these score dynamics across all model scales (1.7B, 4B, and 8B), please refer to Appendix C.

## 5 Our New Method: Selective OPSD

Selective OPSD keeps the OPSD loss unchanged and modifies only the pre-loss data path. Given student-generated rollouts, a router scores complete trajectories and retains those estimated to provide stronger distillation signal.

We instantiate this idea in two settings. Batch-level selection filters independently generated rollouts under a keep ratio and can be teacher-aware or teacher-free. Best-of-$N$ routing samples multiple candidates for the same problem and selects one before teacher forward, keeping the number of priv-

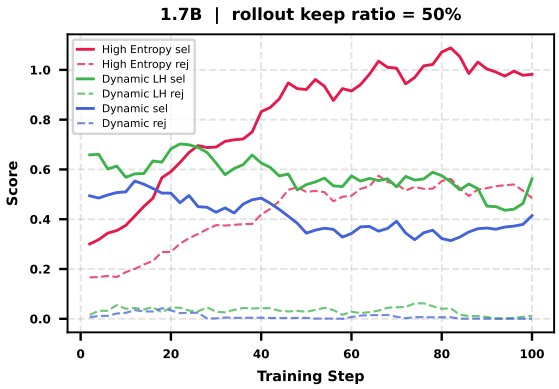


Figure 5: Routing-score dynamics for Qwen3-1.7B at 50% retention. Selected and discarded trajectories remain separated during training.

ileged teacher forward passes fixed with respect to $N$.

### 5.1 Batch-Level Selection

For batch-level selection, each rollout $y_i$ receives a sequence-value score $S(y_i)$, and only the top-ranked samples are retained for the loss. When teacher logits are available, we use the disagreement-aware score introduced in Section 4:

$$S_i^{\text{dyn}} = \text{Score}_{HK}(y_i) = \hat{K}_i(1 - \hat{H}_i). \quad (8)$$

This teacher-aware mode uses actual teacher–student divergence and therefore provides the most direct estimate of corrective value. Since teacher forward is performed before filtering, it improves the quality of samples entering the loss but does not reduce teacher-forward cost.

When routing must happen before teacher forward, we use the teacher-free proxy:

$$S_i^{\text{dyn-lh}} = \text{Score}_{LH}(y_i) = (1 - \hat{L}_i)(1 - \hat{H}_i). \quad (9)$$

This mode, denoted `dynamic_lh`, depends only on student-side statistics: response length and sequence entropy. The implementation first scores all student rollouts and forwards only the selected top-$k$ samples through the teacher, so teacher computation can be reduced when the keep ratio is below one. We also use two controls: `high_entropy`, which keeps the largest-entropy rollouts, and `random`, which uniformly retains the same number of samples.

### 5.2 Best-of-$N$ Routing

BoN routing is the most direct realization of trajectory-level routing. In standard OPSD, each problem produces one student rollout, which is then aligned with the privileged teacher. In BoN routing, by contrast, the student first samples $N$ candidate trajectories $\{y_1, \ldots, y_N\}$ for the same problem. The router selects one candidate before teacher forward, and only this selected trajectory is used for distillation.

For each candidate $y_i$, let $L_i$ denote its effective response length and $H_i$ its average sequence entropy. We normalize both signals within the candidate group:

$$\hat{L}_i = \frac{L_i - \min_j L_j}{\max_j L_j - \min_j L_j + \epsilon}, \quad (10)$$

$$\hat{H}_i = \frac{H_i - \min_j H_j}{\max_j H_j - \min_j H_j + \epsilon}. \quad (11)$$

The BoN routing score is

$$S_i^{\text{BoN}} = (1 - \hat{L}_i)(1 - \hat{H}_i), \quad (12)$$

and the selected trajectory is

$$y^\star = \arg\max_i S_i^{\text{BoN}}. \quad (13)$$

This score applies $\text{Score}_{LH}$ within each problem-specific candidate group. It favors short, low-entropy candidates, which our diagnostics associate with denser corrective signal. Although BoN increases student-side sampling, only the selected trajectory is forwarded to the privileged teacher, so the number of privileged teacher forward passes remains fixed with respect to $N$.

## 6 Experiments

### 6.1 Experimental Setup

We evaluate Selective OPSD on Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Yang et al., 2025), trained on OpenThoughts Math 30k (Guha et al., 2025). We report results on AIME24, AIME25, and HMMT25, following competition-level evaluation practice in reasoning distillation (Ko et al., 2026). All experiments are conducted on 8 NVIDIA A100 GPUs. Full training hyperparameters, LoRA configuration, thinking-mode settings, and evaluation prompt format are provided in Appendix D.

### 6.2 Main Results

We compare Selective OPSD against Base, SFT, GRPO, and Full OPSD baselines. Our main variant uses BoN-4 routing, sampling four student rollouts per problem but distilling only the selected one. Table 4 shows BoN-4 consistently improves over Full OPSD on AIME24, AIME25, and HMMT25

| Model | Method | AIME24 | AIME25 | HMMT25 |
|---|---|---|---|---|
| Qwen3-1.7B | Base | 51.38 | 36.66 | 23.05 |
| | SFT | 48.33 | 36.38 | 22.77 |
| | GRPO | 51.11 | 38.33 | 23.61 |
| | Full OPSD | 56.94 | 43.05 | 28.61 |
| | BoN-4 (Ours) | **59.45** | **45.00** | **30.56** |
| Qwen3-4B | Base | 74.72 | 66.38 | 42.22 |
| | SFT | 70.28 | 62.22 | 43.33 |
| | GRPO | 75.55 | 68.05 | 44.44 |
| | Full OPSD | 75.55 | 69.72 | 44.44 |
| | BoN-4 (Ours) | **78.61** | **72.50** | **47.22** |
| Qwen3-8B | Base | 75.56 | 65.55 | 43.88 |
| | SFT | 72.22 | 64.16 | 42.77 |
| | GRPO | 76.38 | 68.88 | 46.66 |
| | Full OPSD | 76.67 | 70.56 | 47.22 |
| | BoN-4 (Ours) | **79.17** | **73.61** | **50.28** |

Table 4: Main results comparing the base models, full OPSD, and Selective OPSD with BoN-4 routing. All numbers report Avg@12 accuracy.

| Model | Method | AIME24 | AIME25 | HMMT25 |
|---|---|---|---|---|
| Qwen3-1.7B | Base | 11.9 | 9.2 | 5.0 |
| | OPSD | 15.0 | 8.3 | 7.2 |
| | BoN-4 (Ours) | **22.0** | **15.8** | **8.6** |

Table 5: Results under the non-thinking mode (Qwen3-1.7B, Avg@12). BoN-4 improves over both the base model and full OPSD.

across 1.7B, 4B, and 8B models. This indicates that trajectory-level selection enhances distillation quality without increasing teacher-forward counts. We additionally verify in Appendix F that these gains stem strictly from selective routing, as naively training on all candidates or scaling training steps degrades performance.

Furthermore, to confirm that the benefits of our method extend beyond standard long reasoning traces, we additionally evaluate the performance under a non-thinking mode. As shown in Table 5 for the Qwen3-1.7B model, BoN-4 routing continues to provide substantial improvements over both the Base model and Full OPSD, validating the robustness of trajectory-level routing across different generation modes.

### 6.3 BoN Routing Rules

Best-of-$N$ (BoN) is the most distinctive setting for trajectory-level routing because the decision is exclusive: for each problem, only one of the $N$ candidate trajectories can be retained. The key claim to establish here is not only that BoN improves performance, but that the routing rule affects how effectively extra samples are converted into useful distillation signal. Table 6 compares different routing strategies under the same $N = 4$ candidate

| Model | Rule | AIME24 | AIME25 | HMMT25 |
|---|---|---|---|---|
| Qwen3-1.7B | Random-$N$ | 56.39 | 42.77 | 28.33 |
| | Max-$H$ | 57.22 | 43.33 | 28.05 |
| | Min-$H$ | 51.66 | 36.94 | 23.61 |
| | $\mathrm{Score}_{LH}$ (Ours) | **59.45** | **45.00** | **30.56** |
| Qwen3-4B | Random-$N$ | 74.44 | 68.89 | 43.61 |
| | Max-$H$ | 75.83 | 69.72 | 43.33 |
| | Min-$H$ | 75.00 | 66.94 | 42.78 |
| | $\mathrm{Score}_{LH}$ (Ours) | **78.61** | **72.50** | **47.22** |
| Qwen3-8B | Random-$N$ | 76.11 | 71.11 | 46.94 |
| | Max-$H$ | 77.78 | 70.83 | 46.39 |
| | Min-$H$ | 75.28 | 66.67 | 44.44 |
| | $\mathrm{Score}_{LH}$ (Ours) | **79.17** | **73.61** | **50.28** |

Table 6: BoN routing-rule ablation with $N = 4$. All rules sample four candidates but differ in which trajectory is distilled.

generation budget. As predicted by our sequence-value formulation, relying purely on high entropy introduces excessive gradient noise, while relying purely on low entropy often selects uninformative trajectories. Our score formula, $\mathrm{Score}_{LH}$, successfully balances the risk-reward trade-off, consistently yielding the best downstream capabilities.

### 6.4 Marginal Utility of Increasing $N$

We next examine the benefit of increasing the BoN candidate count. Figure 6 shows normalized improvements for Qwen3-1.7B as $N$ increases from 2 to 8. The gains exhibit diminishing returns: $N = 2$ or 4 captures most of the benefit, while $N = 8$ adds only modest improvement. We therefore use BoN-4 as the default setting in the main experiments, balancing routing quality and student-side sampling cost. Thus, Selective OPSD exploits student-side diversity while keeping privileged teacher forward passes fixed, at the cost of student-side sampling that grows with $N$. Full results across model scales are provided in Appendix E.

### 6.5 Pseudo-Target Routing without Ground-Truth Context

As an additional variant of our method, we test whether the teacher-free routing score can provide useful pseudo-targets when ground-truth teacher context is unavailable. We denote this setting as Pseudo-Target Routing. Instead of conditioning the privileged teacher on the ground-truth answer $y^*$, we use the highest-$\mathrm{Score}_{LH}$ rollout in the batch as pseudo-context $y^+$ and distill on a low-scoring rollout. This preserves the routing principle, but uses the selected trajectory as teacher-side context rather than as the direct distillation input.

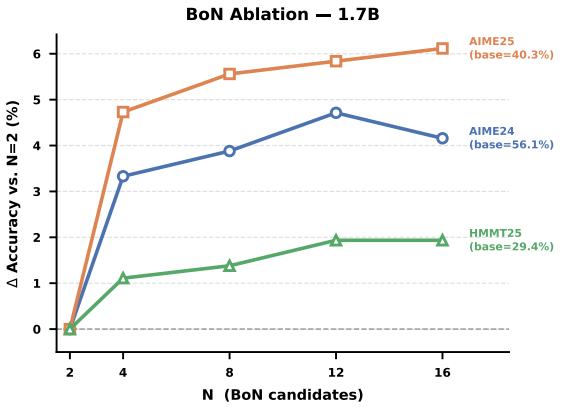


Figure 6: BoN candidate-count ablation. This figure quantifies the marginal utility of extra student-side diversity under a fixed teacher-forward count.

| Model | Method | AIME24 | AIME25 | HMMT25 |
|---|---|---|---|---|
| Qwen3-1.7B | Base | 51.38 | 36.66 | 23.05 |
| | OPSD | 56.94 | **43.05** | **28.61** |
| | Pseudo-Target | **57.22** | 42.22 | 27.78 |
| Qwen3-4B | Base | 74.72 | 66.38 | 42.22 |
| | OPSD | **75.55** | **69.72** | 44.44 |
| | Pseudo-Target | 75.28 | 67.78 | **44.72** |
| Qwen3-8B | Base | 75.56 | 65.55 | 43.88 |
| | OPSD | 76.67 | **70.56** | 47.22 |
| | Pseudo-Target | **76.94** | 69.72 | **47.78** |

Table 7: Pseudo-Target Routing without ground-truth teacher context. The highest-$\text{Score}_{LH}$ rollout is used as teacher-side context, and a low-score rollout is used as the distillation input.

Table 7 shows that this unsupervised variant remains competitive in several settings, suggesting that $\text{Score}_{LH}$ may capture trajectory structure useful for distillation even without ground-truth teacher context. For detailed prompt examples and case studies of this process, please refer to Appendix G.

### 6.6 Qualitative Case Study

To intuitively understand why trajectory-level routing improves distillation, we examine a representative problem from the OpenThoughts training distribution, contrasting a high-value trajectory selected by $\text{Score}_{LH}$ with a discarded low-value trajectory.

**Problem**

How many plums will balance one pear, given that 3 apples and one pear weigh as much as 10 plums, and one apple and 6 plums balance one pear? Assume that fruits of the same kind have the same weight.

**Retained Trajectory (High $\text{Score}_{LH}$):**

**Selected Rollout ($y^+$)**

Let $A$, $P$, and $L$ be the weights of an apple, a pear, and a plum, respectively.
From the first condition, we have:
$3A + P = 10L$
From the second condition, we have:
$A + 6L = P$
We want to find $P$ in terms of $L$. Substitute $P = A + 6L$ into the first equation:
$3A + (A + 6L) = 10L \implies 4A + 6L = 10L \implies 4A = 4L \implies A = L$.
Now substitute $A = L$ back into the second equation:
$P = A + 6L = L + 6L = 7L$.
So, one pear balances 7 plums.
\boxed{7}

*Metrics:* Length: Short ($\hat{L} \approx 0.15$), Entropy: Low ($\hat{H} \approx 0.18$), $\text{Score}_{LH}$: **High**.

**Discarded Trajectory (Low $\text{Score}_{LH}$):**

**Discarded Rollout ($y^-$)**

Let's see. 3 apples + 1 pear = 10 plums.
And 1 apple + 6 plums = 1 pear.
Maybe I can guess some numbers. Let 1 plum = 1 unit of weight.
Then 3 apples + 1 pear = 10.
And 1 apple + 6 = 1 pear.
If 1 apple is 1, then pear = 7. Let's check: 3(1) + 7 = 10. This works!
Wait, what if 1 apple is 2? Then pear = 2 + 6 = 8.
Then 3(2) + 8 = 14, not 10. So apple must be 1.
If apple is 1 and plum is 1, then pear is 7.
So 7 plums balance one pear.
Let me double check this logic. Is it possible there are other solutions?
No, the equations are linear so the ratio is fixed.
The answer is 7.
\boxed{7}

*Metrics:* Length: Long ($\hat{L} \approx 0.75$), Entropy: High ($\hat{H} \approx 0.68$), $\text{Score}_{LH}$: **Low**.

Overall, this example shows that trajectory-level routing is not just filtering incorrect answers. It also distinguishes correct rollouts by directness, confidence, and suitability as alignment targets.

## 7 Conclusion

This paper frames OPSD as a problem of *selective alignment*: not all on-policy rollouts are equally worth distilling. We introduce trajectory-level routing and show that rollout value can be estimated from entropy, teacher–student disagreement, and response length. Selective OPSD uses these signals for batch filtering and BoN routing, improving distillation under a fixed teacher-forward count.

## Limitations

This work focuses on mathematical reasoning benchmarks and Qwen3 model scales, so the effectiveness of trajectory-level routing on broader domains, larger models, and more diverse generation settings remains to be tested. Our teacher-free routing relies on response length as a proxy for teacher–student disagreement; although supported by our diagnostics, this relationship may not hold uniformly across tasks or model families. In addition, BoN routing keeps the number of privileged teacher forward passes fixed but increases student-side generation cost, which may matter in deployment settings where sampling is expensive.

## Ethical Considerations

Selective OPSD is a training method for improving reasoning models and does not introduce new data collection or human annotation procedures. However, stronger reasoning models may inherit or amplify biases, unsafe behaviors, or memorized content from their base models and training data. Reducing teacher-side computation may also make advanced model adaptation more accessible, which has both beneficial and potentially harmful uses. We therefore recommend applying standard safety evaluation and deployment safeguards when using Selective OPSD in downstream applications.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

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*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

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

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. 2026. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

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, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. 2026. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. Distillm-2: A contrastive approach boosts the distillation of llms, 2025. *URL https://arxiv.org/abs/2503.07067*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words, 2022. *URL https://arxiv. org/abs/2205.14334*.

Yihong Liu, Raoyuan Zhao, Michael A Hedderich, and Hinrich Schütze. 2026. Crosslingual on-policy self-distillation for multilingual reasoning. *arXiv preprint arXiv:2605.09548*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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*.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.

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

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and 1 others. 2026. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. 2026a. Paced: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*.

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

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

Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026a. Online experiential learning for language models. *arXiv preprint arXiv:2603.16856*.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026b. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E Tillman. 2026. Fast and effective on-policy distillation from reasoning prefixes. *arXiv preprint arXiv:2602.15260*.

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, 2026. *URL https://arxiv. org/abs/2601.18734*.

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

## A Complete Derivation of the Sequence Distillation Value Score

This appendix provides the motivation for the $\mathrm{Score}_{LH}$ score introduced in Section 4. We proceed in four steps: (1) we establish a token-level signal-to-noise baseline; (2) we show why sequence-level routing requires a different noise model; (3) we derive a teacher-aware Oracle Score $V_{\text{oracle}}$; and (4) we explain how the teacher-free motivation is converted into the bounded practical score $\mathrm{Score}_{LH} = (1 - \hat{H})(1 - \hat{L})$.

### A.1 A Token-Level SNR Baseline from OPSD's Own Formulation

Before addressing sequences, it is instructive to characterize the efficiency of a single token's gradient contribution within the OPSD framework. We use OPSD's policy-gradient interpretation (Zhao et al.), in which the training objective can be written as

$$\mathcal{L}(\theta) \approx -\mathbb{E}_{\hat{y}\sim\pi_S}\left[\frac{1}{|\hat{y}|}\sum_{t=1}^{|\hat{y}|} A_t \log \pi_S(\hat{y}_t \mid x, \hat{y}_{<t})\right]. \tag{14}$$

where the per-token advantage is

$$\begin{aligned} A_t = &\log \pi_T(\hat{y}_t \mid x, y^*, \hat{y}_{<t}) \\ &- \log \pi_S(\hat{y}_t \mid x, \hat{y}_{<t}). \end{aligned} \tag{15}$$

Equation (14) treats each token's update as a policy-gradient step with a dense log-ratio reward, a perspective already established in the OPSD paper.

**Signal.** By the log-derivative identity, the expected gradient contribution of token $t$ satisfies

$$\begin{aligned} &\mathbb{E}[A_t \nabla_\theta \log \pi_S(\hat{y}_t|\cdot)] \\ &\quad = \nabla_\theta \mathbb{E}_{\hat{y}_t\sim\pi_S}[A_t] \\ &\quad = -\nabla_\theta D_{KL}(\pi_S\|\pi_T)\big|_t, \end{aligned} \tag{16}$$

so the expected signal magnitude is proportional to the teacher–student divergence $\delta_t = D_{KL}(\pi_T\|\pi_S)|_t$.

**Noise.** The gradient variance at position $t$ is dominated by the second moment of the score function (the REINFORCE variance (Williams, 1992)):

$$\begin{aligned} M_t &= \mathbb{E}_{\hat{y}_t\sim\pi_S}\left[\|\nabla_\theta \log \pi_S(\hat{y}_t|\cdot)\|^2\right] \\ &= \mathrm{tr}(F_t), \end{aligned} \tag{17}$$

where $F_t$ is the Fisher information matrix of $\pi_S$ at position $t$. For a softmax policy $\pi_S(v) \propto e^{z_v}$, the score function is $\nabla_z \log \pi_S(v) = e_v - \pi_S$, giving

$$\begin{aligned} \mathrm{tr}(F_t) &= \mathbb{E}_{v\sim\pi_S}\left[\|e_v - \pi_S\|^2\right] \\ &= 1 - \sum_v \pi_S(v)^2 \\ &\le H_t, \end{aligned} \tag{18}$$

where the inequality follows from $-\log p \ge 1 - p$ applied entry-wise. Both $\mathrm{tr}(F_t)$ and $H_t$ vanish for a near-deterministic student and approach their maxima for a near-uniform student, making entropy a tight and interpretable proxy for the Fisher trace. We write $M_t \propto H_t$.

**Token-level SNR.** Combining Signal and Noise:

$$\mathrm{SNR}_t = \frac{\mathrm{Signal}_t}{\mathrm{Noise}_t} \propto \frac{\delta_t}{H_t}. \tag{19}$$

This result follows entirely from OPSD's policy-gradient structure and standard information geometry; no external framework is required. The ratio $\delta_t/H_t$ is high precisely for tokens that the teacher strongly disagrees with ($\delta_t$ large) while the student is confidently wrong ($H_t$ small)—the combination most worth correcting at the token level.

### A.2 Why Sequence-Level Routing Requires a Different Noise Model

Token-level analysis treats each position independently, so noise at position $s$ does not affect position $t \neq s$. At the sequence level, however, autoregressive generation creates strong coupling: position $t$ is conditioned on $\tau_{<t} = (\tau_1, \ldots, \tau_{t-1})$, which is itself sampled from the student. Any uncertainty at position $s < t$ distorts the conditioning context for all subsequent positions—this is **exposure bias** (Bengio et al., 2015).

Let $\epsilon_t$ denote the deviation of the student's state at step $t$ from the oracle state (i.e., the state that would be reached if all prior tokens had been chosen deterministically optimally). Under a first-order error-propagation model, the per-step deviation grows as a random walk:

$$\begin{aligned} \epsilon_t &= \epsilon_{t-1} + \xi_t, \\ \mathbb{E}[\xi_t] &= 0, \\ \mathbb{E}[\xi_t^2] &\propto H_t, \end{aligned} \tag{20}$$

where $\xi_t$ is an independent increment whose variance is proportional to the student entropy $H_t$: a more uncertain model makes larger random deviations. Summing over all $L_\tau$ positions gives:

$$\begin{aligned} \mathrm{Var}[\epsilon_{L_\tau}] &= \sum_{t=1}^{L_\tau} \mathbb{E}[\xi_t^2] \\ &\propto \bar{H}_\tau \cdot L_\tau. \end{aligned} \tag{21}$$

This stands in stark contrast to the token-independence model, which yields variance $\propto \bar{H}_\tau/L_\tau$ (decreasing with length). Under exposure bias, variance *increases linearly* with length because errors accumulate additively rather than averaging out. We define the **sequence-level noise** accordingly:

$$\mathrm{Noise}(\tau) \propto \bar{H}_\tau \cdot L_\tau. \tag{22}$$

### A.3 Signal: Expected Corrective Benefit of a Trajectory

The signal from trajectory $\tau$ is the total corrective information the teacher can deliver. Because OPSD minimizes the mean token-level KL divergence along the rollout, the expected gradient magnitude from $\tau$ is proportional to:

$$\text{Signal}(\tau) = \bar{K}_\tau = \frac{1}{L_\tau} \sum_{t=1}^{L_\tau} D\Big(\pi_T(\cdot \mid x, y^*, \tau_{<t}) \,\Big\|\, \pi_S(\cdot \mid x, \tau_{<t})\Big). \tag{23}$$

A large $\bar{K}_\tau$ means the teacher and student disagree substantially on average, providing an informative corrective gradient. A trajectory on which $\bar{K}_\tau \approx 0$ produces near-zero gradient in expectation: there is nothing for the teacher to correct.

### A.4 The Oracle Score

Combining Signal (23) and Noise (22) as a signal-to-noise ratio gives the theoretical **Oracle Score**:

$$V_{\text{oracle}}(\tau) = \frac{\text{Signal}(\tau)}{\text{Noise}(\tau)} = \frac{\bar{K}_\tau}{\bar{H}_\tau \cdot L_\tau}. \tag{24}$$

The formula has a transparent three-way interpretation. (i) The numerator $\bar{K}_\tau$ rewards trajectories on which the teacher's correction is substantive. (ii) The entropy factor $\bar{H}_\tau$ in the denominator penalizes global student confusion, which injects random-walk noise into the parameter update. (iii) The length factor $L_\tau$ penalizes long trajectories, which accumulate exposure-bias errors over more steps. Under this score, the most valuable trajectory is one that is short, confident, and strongly disagreed upon by the teacher—precisely the Q3 quadrant (low entropy, high KL, short length) identified in our diagnostic analysis.

Computing $V_{\text{oracle}}$ requires the teacher for $\bar{K}_\tau$, making it a teacher-aware estimator analogous to the $\text{Score}_{HK}$ (`dynamic` mode) in our batch-selection experiments.

### A.5 Teacher-Free Substitution via the Length–KL Relationship

The offline analysis in Section 4.3 and Appendix B consistently reveals an inverse relationship between sequence length and average teacher–student divergence across two datasets and all four quadrant subsets:

$$\bar{K}_\tau \approx \frac{c}{L_\tau} \tag{25}$$

for a positive constant $c$. The intuition is that a model condensing its reasoning into fewer tokens necessarily encounters a higher density of correction opportunities per token, while a longer rambling trajectory dilutes the same total disagreement over more positions. Substituting (25) into (24):

$$V_{\text{Teacher-Free}}(\tau) \propto \frac{c/L_\tau}{\bar{H}_\tau \cdot L_\tau} = \frac{c}{\bar{H}_\tau \cdot L_\tau^2}. \tag{26}$$

Equation (26) depends only on the student's own statistics—length and entropy—and requires no teacher forward pass. The quadratic dependence on $L_\tau^2$ reflects the two independent roles of length: once as a direct amplifier of exposure-bias noise (Eq. 22), and once as a proxy for the missing teacher signal (Eq. 25).

### A.6 From Teacher-Free Motivation to Practical $\text{Score}_{LH}$

Equation (26) defines a preference ordering over trajectories, but its absolute values vary across batches. For numerical stability, we apply within-batch min-max normalization:

$$\hat{H}_i = \frac{H_i - \min_j H_j}{\max_j H_j - \min_j H_j + \varepsilon}, \quad \hat{L}_i = \frac{L_i - \min_j L_j}{\max_j L_j - \min_j L_j + \varepsilon}. \tag{27}$$

so that $\hat{H}_i, \hat{L}_i \in [0, 1]$ within every batch.

We then use the following practical $\text{Score}_{LH}$ score:

$$\text{Score}_{LH}(y_i) = (1 - \hat{H}_i)(1 - \hat{L}_i). \tag{28}$$

**Rank-Preserving Surrogate via First-Order Approximation.** While Equation (26) provides the theoretical motivation, applying raw reciprocals in practice suffers from severe numerical instability when $\hat{H}_i$ or $\hat{L}_i$ approaches zero. However, since trajectory-level routing only relies on the *relative preference ordering* (i.e., ranking) within a candidate set, we can substitute the unstable reciprocal formulation with a bounded, order-preserving surrogate.

Specifically, considering the Taylor expansion of $f(x) = 1/x$ around $x = 1$, we have $f(x) \approx 1 - (x - 1) = 2 - x$. After min-max normalization

where variables are squashed to $\hat{x} \in [0, 1]$, both $1/\hat{x}$ and $1 - \hat{x}$ serve as strictly monotonically decreasing functions. Therefore, we substitute the inverse terms with their linear complements:

$$\frac{1}{\hat{H}_i} \to (1 - \hat{H}_i), \quad \frac{1}{\hat{L}_i^2} \to (1 - \hat{L}_i). \tag{29}$$

Note that we relax the quadratic term $\hat{L}_i^2$ to a linear penalty $1 - \hat{L}_i$ to prevent overly aggressive penalization of length in bounded space. The resulting multiplicative form $\mathrm{Score}_{LH} = (1 - \hat{H}_i)(1 - \hat{L}_i)$ rigorously preserves the joint monotonic preference: it strictly decreases with respect to both entropy and length, while elegantly bounding the output space to $[0, 1]$. This transforms the unbounded reciprocal heuristic into a stable and theoretically sound routing metric.

**Practical advantages.** The $\mathrm{Score}_{LH}$ form $(1 - \hat{H})(1 - \hat{L})$ has three practical benefits for routing. First, its range is bounded to $[0, 1]$, preventing numerical overflow in large batches. Second, the multiplicative form is *jointly strict*: a trajectory must be simultaneously short *and* confident to receive a high score. Table 8 summarizes this intended heuristic behavior across the four diagnostic quadrants.

| Quadrant | Entropy | Length | Implicit KL | $\mathrm{Score}_{LH}$ |
|---|---|---|---|---|
| Q1 (High-$H$, Short) | high | short | high | medium |
| Q2 (High-$H$, Long) | high | long | low | $\approx 0$ |
| Q3 (Low-$H$, Short) | low | short | high | $\approx 1$ |
| Q4 (Low-$H$, Long) | low | long | low | med–low |

Table 8: Heuristic $\mathrm{Score}_{LH}$ profile across the four diagnostic quadrants. The practical score favors low-entropy short trajectories, while assigning lower scores to cases where either uncertainty or length makes the teacher-free evidence ambiguous.

**Summary of the derivation chain.**

$$\underbrace{V_{\text{oracle}} = \frac{\bar{K}_\tau}{\bar{H}_\tau L_\tau}}_{\text{SNR + exposure bias}} \xrightarrow{\bar{K}_\tau \approx c/L_\tau} \underbrace{V_{\text{TF}} \propto \frac{1}{\bar{H}_\tau L_\tau^2}}_{\text{teacher-free}}$$
$$\xrightarrow{\text{first-order surrogate}} \underbrace{\mathrm{Score}_{LH} = (1 - \hat{H}_\tau)(1 - \hat{L}_\tau)}_{\text{practical score}}. \tag{30}$$

The first arrow is motivated by the exposure-bias gradient model and the empirical length–KL relationship. The final arrow applies a first-order monotonic approximation, converting the unbounded reciprocal motivation into a stable, rank-preserving surrogate for practical routing.

# B Detailed Quadrant Analysis

This appendix provides the detailed numerical results for the offline quadrant analysis referenced in Section 4.3. To understand the relationship between sequence properties and distillation value, we partition the student-generated trajectories into four quadrants based on their average sequence entropy and sequence length, split at their respective medians. The analysis is presented for all samples, as well as split by correctness (whether the student rollout's final answer is correct), allowing us to control for the confounding effect of reasoning success.

Table 9 shows the comprehensive statistics for the OpenThoughts dataset. Several key patterns emerge from this quadrant breakdown:

First, within any given entropy level and across all subsets (All Samples, Correct, and Wrong), trajectories with shorter lengths consistently exhibit higher KL divergence. For example, in the "All Samples" split under High Entropy, short trajectories yield a KL of $0.0213$ compared to $0.0172$ for long trajectories. This holds true even when isolating only correct or only wrong responses, confirming that the length–KL inverse relationship is an intrinsic property of the token-matching process rather than a mere artifact of incorrect answers tending to be longer.

Second, entropy acts as a strong independent indicator of divergence. Across both length groups and all correctness splits, high-entropy trajectories consistently show significantly higher KL divergence than low-entropy trajectories. For instance, among short and wrong rollouts, high-entropy trajectories yield a KL of $0.0222$, whereas low-entropy trajectories yield only $0.0138$. This confirms that sequence entropy directly captures teacher–student disagreement, justifying its role in our scoring functions.

Finally, we observe the interaction between correctness and length. In the "All Samples" split, short trajectories are associated with higher accuracy (63.2% vs. 45.7% in Low $H$; 45.2% vs. 31.3% in High $H$). This suggests that shorter paths to the solution are generally more reliable. By preferring shorter lengths, our proxy metric $\mathrm{Score}_{LH}$ implicitly upweights correct and efficient reasoning paths, which are highly valuable for distillation.

Table 10 provides the corresponding statistics for the DAPO dataset (**?**). The trends are strikingly similar: the inverse relationship between se-

quence length and teacher–student divergence remains strictly consistent across all conditions, and high entropy remains robustly associated with high KL. This cross-dataset validation strengthens the empirical foundation of using Length and Entropy as teacher-free proxies for estimating trajectory-level corrective signals.

## C Comprehensive Score Dynamics Across Model Scales

Expanding on the analysis in Section 4.4, this appendix provides the complete training dynamics of the routing scores across different model scales. Figures 7, 8, and 9 illustrate the score separation between retained (top-50%) and discarded (bottom-50%) sequences for the Qwen3-1.7B, 4B, and 8B models, respectively.

We observe several critical phenomena across these training trajectories:

First, the **consistency across model scales** is striking. Regardless of whether the student is a relatively small 1.7B model or a more capable 8B model, the routing scores consistently partition the student rollouts into two distinct populations. This suggests that the non-uniformity of rollout distillation value is not merely an artifact of an under-trained small model, but rather a fundamental property of the on-policy generation space. Even as base model capabilities increase, there remains significant variance in trajectory-level teacher–student disagreement that Selective OPSD can exploit.

Second, the **stability of the score gap over time** highlights the robustness of our metrics. In standard reinforcement learning or score-based filtering, metrics often collapse or suffer from over-optimization as the policy updates. However, throughout the entirety of the training steps across all three model sizes, we observe a robust and persistent gap between the chosen and discarded populations. The scores do not collapse; instead, the separation remains stable, ensuring that the router continues to provide a meaningful filtering signal even late in the training process when the student is already highly aligned.

Third, the **alignment between** $\mathrm{Score}_{HK}$ **and its proxy** $\mathrm{Score}_{LH}$ is preserved at scale. Both the teacher-aware oracle score and the teacher-free proxy maintain similar discriminative dynamics. The fact that $\mathrm{Score}_{LH}$ smoothly tracks the separation without requiring expensive teacher logits validates its use in Best-of-$N$ routing across different model sizes.

Collectively, this empirical evidence validates two key points: first, that rollout value remains highly non-uniform throughout the entire on-policy training lifecycle; and second, that our proposed sequence-value estimators maintain their discriminative power without collapsing as the student policy updates, providing a reliable foundation for trajectory-level routing at any model scale.

## D Experimental Setup Details

### Models and Training Framework

All experiments use a LoRA-based OPSD setting (Zhao et al.): the student is updated through LoRA adapters, while teacher forward passes are computed by disabling the adapters and conditioning the corresponding base model on privileged context. When reference refresh is used, the teacher is updated from a student-derived checkpoint rather than treated as a separately trained external model. Experiments are run on 8 NVIDIA A100 (40GB) GPUs using DeepSpeed ZeRO-2 with the Accelerate framework.

### Training Hyperparameters

Table 11 lists the full set of training hyperparameters used across all model scales.

### LoRA Configuration

LoRA adapters are applied to the following linear modules: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`. Rank and scaling are fixed to $r = 64$, $\alpha = 128$ across all scales. Teacher forward passes disable the active student adapters and use the corresponding base/reference weights with privileged conditioning; no separately trained external teacher model is maintained.

### Thinking-Mode Configuration

Qwen3 supports an explicit thinking mode that prepends a `<think>` block to the model's generation. All main experiments use the following configuration:

- **Student rollout** (`student_thinking=False`): the student generates without thinking tokens. This matches the standard OPSD setup and keeps rollout length comparable across methods.
- **Teacher forward** (`teacher_thinking=True`): the privileged teacher prompt uses thinking mode.

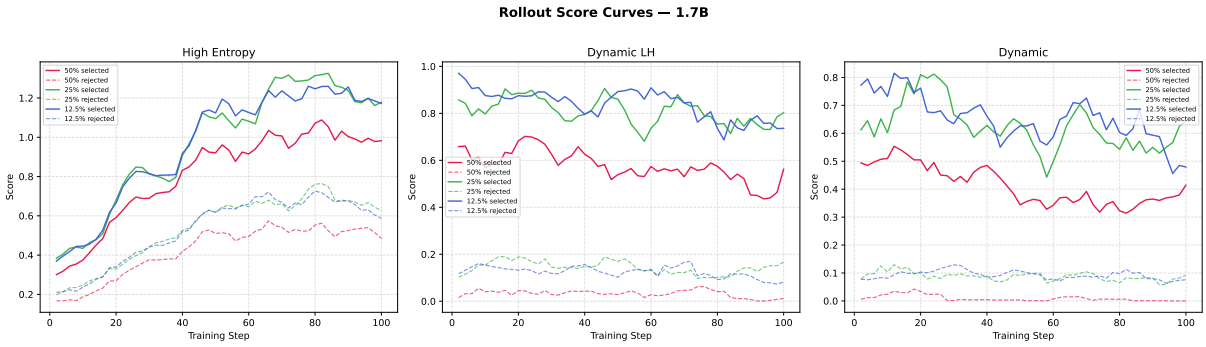


Figure 7: Score dynamics between selected and discarded trajectories during the training of Qwen3-1.7B.

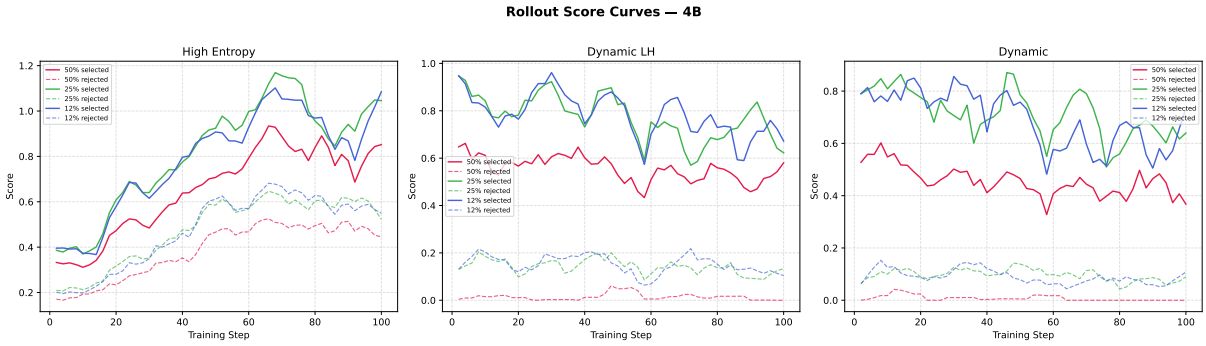


Figure 8: Score dynamics between selected and discarded trajectories during the training of Qwen3-4B.

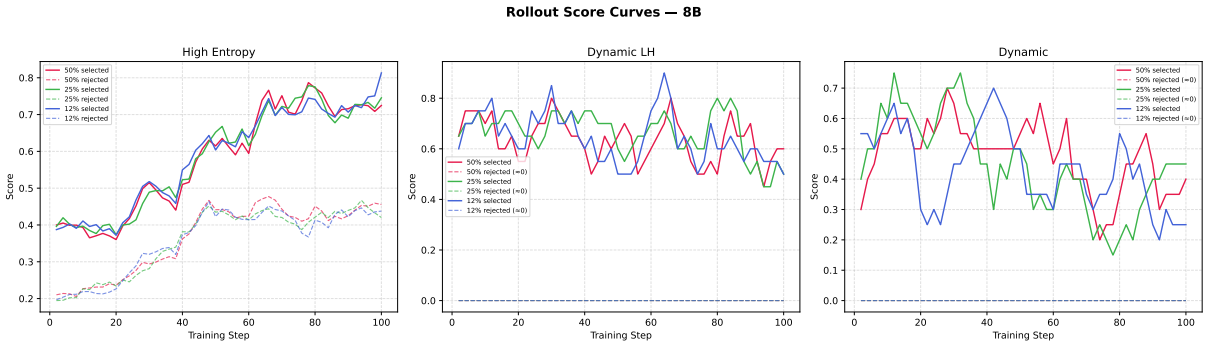


Figure 9: Score dynamics between selected and discarded trajectories during the training of Qwen3-8B.

| Subset | Entropy Level | Short Length | Long Length |
|---|---|---|---|
| **All Samples** | Low $H$ | $n = 961$, acc=63.2%, len=572, kl=0.0133 | $n = 597$, acc=45.7%, len=1594, kl=0.0104 |
| | High $H$ | $n = 599$, acc=45.2%, len=665, kl=0.0213 | $n = 958$, acc=31.3%, len=1795, kl=0.0172 |
| **Correct** | Low $H$ | $n = 607$, acc=100.0%, len=552, kl=0.0130 | $n = 273$, acc=100.0%, len=1441, kl=0.0101 |
| | High $H$ | $n = 270$, acc=100.0%, len=656, kl=0.0203 | $n = 299$, acc=100.0%, len=1636, kl=0.0165 |
| **Wrong** | Low $H$ | $n = 354$, acc=0.0%, len=607, kl=0.0138 | $n = 324$, acc=0.0%, len=1722, kl=0.0106 |
| | High $H$ | $n = 328$, acc=0.0%, len=673, kl=0.0222 | $n = 657$, acc=0.0%, len=1868, kl=0.0174 |

Table 9: Detailed quadrant statistics for the OpenThoughts dataset in a $2 \times 2$ format. Within every group (All, Correct, Wrong), short trajectories consistently manifest a higher KL divergence than long trajectories for both low-entropy and high-entropy regimes.

| Subset | Entropy Level | Short Length | Long Length |
|---|---|---|---|
| **All Samples** | Low $H$ | $n = 780$, acc=54.1%, len=982, kl=0.0109 | $n = 691$, acc=28.9%, len=2380, kl=0.0091 |
| | High $H$ | $n = 692$, acc=21.0%, len=1075, kl=0.0186 | $n = 779$, acc=12.5%, len=2422, kl=0.0149 |
| **Correct** | Low $H$ | $n = 422$, acc=100.0%, len=946, kl=0.0104 | $n = 200$, acc=100.0%, len=2169, kl=0.0087 |
| | High $H$ | $n = 145$, acc=100.0%, len=1021, kl=0.0184 | $n = 97$, acc=100.0%, len=2266, kl=0.0143 |
| **Wrong** | Low $H$ | $n = 358$, acc=0.0%, len=1025, kl=0.0114 | $n = 491$, acc=0.0%, len=2466, kl=0.0093 |
| | High $H$ | $n = 547$, acc=0.0%, len=1089, kl=0.0186 | $n = 682$, acc=0.0%, len=2445, kl=0.0150 |

Table 10: Detailed quadrant statistics for the DAPO dataset in a $2 \times 2$ format. The inverse relationship between sequence length and teacher-student divergence remains strictly consistent.

| Hyperparameter | 1.7B / 4B | 8B |
|---|---|---|
| Training steps | 100 | |
| Learning rate | $5 \times 10^{-6}$ | |
| Gradient clipping | 0.1 | |
| Per-device batch size | 4 | |
| Gradient accumulation | 2 | |
| Effective batch size | 32 | |
| Max completion length | 2048 | |
| Max sequence length | 20000 | |
| JSD $\beta$ (forward KL) | 0 | |
| JSD token clip | 0.05 | 0.06 |
| Generation temperature | 1.1 | |
| Top-$p$ | 0.95 | |
| Top-$k$ | 20 | |
| Generation backend | vLLM colocate | |
| vLLM GPU memory util. | 0.60 | |

Table 11: Training hyperparameters. All models use the same settings except for the JSD token clip threshold.

| SFT Hyperparameter | 1.7B / 4B | 8B |
|---|---|---|
| Training steps | 100 | |
| Learning rate | $5 \times 10^{-6}$ | |
| Per-device batch size | 4 | |
| Gradient accumulation | 4 | |
| lora-r | 64 | |
| lora target modules | all | |
| Max sequence length | 16000 | |

Table 12: Training hyperparameters for SFT.

| Hyperparameter | 1.7B / 4B | 8B |
|---|---|---|
| Training steps | 100 | |
| Learning rate | $5 \times 10^{-6}$ | |
| Gradient clipping | 0.1 | |
| Per-device batch size | 1 | |
| Gradient accumulation | 4 | |
| Num generations | 8 | |
| Max completion length | 2048 | |
| Max sequence length | 16000 | |
| Generation temperature | 1.2 | |

Table 13: Training hyperparameters for GRPO.

This gives the teacher access to chain-of-thought reasoning when conditioning on the reference solution, producing a richer corrective signal.

- **Evaluation**: inference is run with thinking mode enabled (`enable_thinking=True`) unless otherwise noted. The non-thinking evaluation variant (`-no_thinking`) uses top-$p = 0.8$ instead of 0.95, consistent with Qwen3 recommendations.

The `student_thinking` and `teacher_thinking` flags are passed directly to `tokenizer.apply_chat_template(..., enable_thinking=...)` when constructing training prompts in the data collator.

### Evaluation Protocol

Evaluation uses vLLM with tensor parallelism across 4 GPUs. For each checkpoint, we generate $n = 12$ solutions per problem and report Avg@12 accuracy. Generation parameters during evalua-

tion are: temperature $= 1.0$, thinking-mode top-$p = 0.95$, `max_new_tokens`$= 38912$.

### Prompt Format

The student prompt follows the standard instruction format:

**Student Prompt**

```
Problem: {problem}

Please reason step by step, and put
your final answer within \boxed{}.
```

The teacher prompt additionally includes the reference solution wrapped between delimiter markers, followed by a transition instruction that asks the model to understand the solution and then independently derive the same answer:

**Teacher Prompt (Privileged OPSD)**

```
Problem: {problem}

Here is a reference solution to this
problem:
=== Reference Solution Begin ===
{solution}
=== Reference Solution End ===

After reading the reference solution
above, make sure you truly understand
the reasoning behind each step-do not
copy or paraphrase it. Now,
using your own words and independent
reasoning, derive the same final
answer to the problem above. Think step by
step, explore different
approaches, and don't be afraid to
backtrack or reconsider if something
doesn't work out:

Please reason step by step, and put
your final answer within \boxed{}.
```

The evaluation prompt uses the same student format above, with `tokenizer.apply_chat_template` called with `add_generation_prompt=True` and `enable_thinking` set according to the evaluation mode.

## E Complete BoN Candidate-Count Ablation

Expanding on the analysis in Section 6.4, this appendix presents the full BoN candidate-count ablation across different model scales. Figure 10 shows the marginal utility of increasing the number of candidates $N$ under a fixed teacher-forward count for the Qwen3-1.7B, 4B, and 8B models. The results consistently demonstrate that increasing $N$ yields diminishing returns across all model sizes, confirming that our method efficiently leverages student-side diversity without requiring an overly large candidate pool.

## F Ablation on Generation Budget and Multi-Rollout Distillation

In Selective OPSD with Best-of-4 (BoN-4) routing, the student generates four candidate rollouts per problem, but only the selected one is forwarded to the teacher and used for distillation. A natural question is whether the performance gains stem from the routing mechanism itself, or simply from the increased student-side generation budget. To isolate the effect of selective routing, we compare BoN-4 against three alternative ways to utilize the expanded generation budget:

- **4x-Steps:** Instead of generating multiple rollouts per problem, we scale up the training by feeding $4\times$ more unique problems (or equivalently, extending the training steps proportionally, `step=400`).
- **All-R:** We retain all 4 rollouts ($N = 4$) generated per problem and perform distillation on every single one of them independently.
- **All-R-Mean:** We process all 4 rollouts ($N = 4$) per problem, but average their token-level OPSD losses before the backward pass to reduce the dominance of long, noisy trajectories.

Table 14 reports the results of these ablations across all three Qwen3 model scales (1.7B, 4B, and 8B).

We observe a striking and consistent phenomenon across all model sizes: naively extending the training budget by $4\times$ (**4x-Steps**) severely degrades performance, causing it to collapse back to roughly the Base Model level. This suggests that prolonged exposure to uninformative or noisy on-policy teacher signals without routing leads to harmful over-optimization. Furthermore, when training on all 4 rollouts per problem—whether independently (**All-R**) or by averaging the loss (**All-R-Mean**)—the performance lands squarely between the Base Model and Full OPSD. While it surpasses the base model, it clearly underperforms standard single-rollout OPSD, as the influx

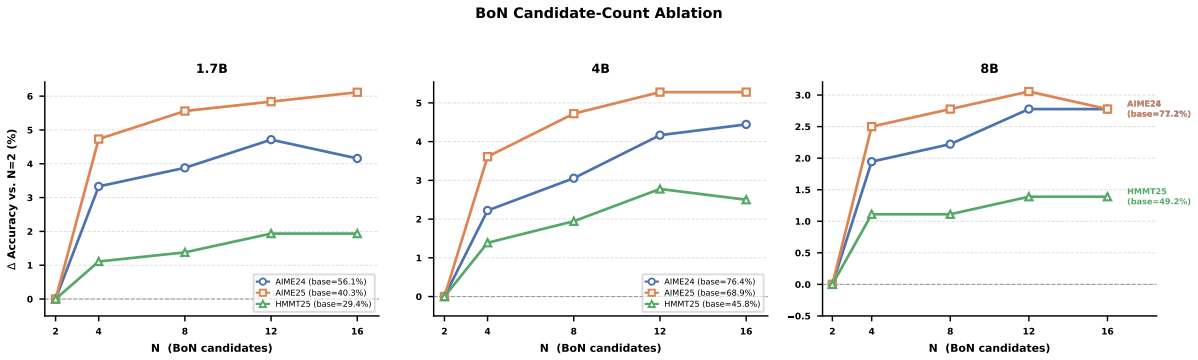


Figure 10: Complete BoN candidate-count ablation across different model scales (1.7B, 4B, and 8B).

| Model | Method | AIME24 | AIME25 | HMMT25 |
|---|---|---|---|---|
| Qwen3-1.7B | Base | 51.38 | 36.66 | 23.05 |
| | OPSD | 56.94 | 43.05 | 28.61 |
| | 4x-Steps | 51.11 | 36.94 | 22.77 |
| | All-R | 54.16 | 40.00 | 25.55 |
| | All-R-Mean | 53.88 | 40.27 | 25.27 |
| | **BoN-4 (Ours)** | **59.45** | **45.00** | **30.56** |
| Qwen3-4B | Base | 74.72 | 66.38 | 42.22 |
| | OPSD | 75.55 | 69.72 | 44.44 |
| | 4x-Steps | 74.44 | 66.11 | 41.94 |
| | All-R | 75.00 | 68.33 | 43.33 |
| | All-R-Mean | 75.27 | 68.05 | 43.61 |
| | **BoN-4 (Ours)** | **78.61** | **72.50** | **47.22** |
| Qwen3-8B | Base | 75.56 | 65.55 | 43.88 |
| | OPSD | 76.67 | 70.56 | 47.22 |
| | 4x-Steps | 75.27 | 65.83 | 44.16 |
| | All-R | 76.11 | 68.61 | 45.83 |
| | All-R-Mean | 75.83 | 68.33 | 45.55 |
| | **BoN-4 (Ours)** | **79.17** | **73.61** | **50.28** |

Table 14: Comparison of different strategies for utilizing a $4\times$ generation budget across all model scales. Indiscriminate training on all rollouts or simply scaling up the steps consistently leads to sub-optimal performance or collapse compared to selective routing.

of low-quality, high-noise trajectories overwrites the useful learning signal. These results confirm that *selective alignment*—deliberately isolating the most valuable trajectory while discarding the rest—is the true driver of the performance gains across all parameter scales.

## G Unsupervised Distillation via Score-Based Pseudo-Targets

In Section 6.5, we introduce Pseudo-Target Routing, which removes the reliance on ground-truth answers during the on-policy distillation phase. Instead of providing the privileged teacher with a human-annotated reference solution, this variant uses the student's own highest-scoring candidate ($y^+$) under our teacher-free proxy $\text{Score}_{LH}$ as the pseudo-target. The distillation loss is then computed on the lowest-scoring candidate ($y^-$).

To illustrate this process, we provide a representative example from the OpenThoughts training distribution.

### Example Rollouts and Scoring

**Problem**

Given that $1 \leq x, y, z \leq 6$, how many cases are there in which the product of natural numbers $x, y, z$ is divisible by 10?

Suppose the student generates multiple candidates. The router evaluates their sequence length and entropy to estimate their distillation value.

**Candidate $y^+$ (High $\text{Score}_{LH}$, Selected as Pseudo-Target):**

**Selected Rollout ($y^+$)**

The total number of cases for $(x, y, z)$ is $6 \times 6 \times 6 = 216$.
For $xyz$ to be divisible by 10, it must be divisible by 2 and by 5.
Let's find the complement: cases where $xyz$ is NOT divisible by 10.
This happens if $xyz$ is not divisible by 2 OR not divisible by 5.
Let $A$ be the set of cases where $xyz$ is not divisible by 2 (all must be odd: 1, 3, 5). $|A| = 3^3 = 27$.
Let $B$ be the set of cases where $xyz$ is not divisible by 5 (none can be 5: 1, 2, 3, 4, 6). $|B| = 5^3 = 125$.
The intersection $A \cap B$ is cases where numbers are odd and not 5 (must be 1, 3). $|A \cap B| = 2^3 = 8$.
By Principle of Inclusion-Exclusion, $|A \cup B| = 27 + 125 - 8 = 144$.
The number of valid cases is $216 - 144 = 72$.
\boxed{72}

*Analysis:* This trajectory is concise and exhibits low entropy (high confidence), resulting in a high

$\text{Score}_{LH}$. It serves as a high-quality pseudo-target.

**Candidate $y^-$ (Low $\text{Score}_{LH}$, Selected for Distillation):**

**Discarded Rollout ($y^-$)**

Let's think about this. The product $xyz$ must be a multiple of 10.
This means at least one of them must be 5, and at least one must be even (2, 4, or 6).
What if we have exactly one 5? There are 3 positions for the 5.
Then the other two numbers can be anything, as long as at least one is even.
Wait, this is getting complicated. Let's list the possibilities for the number of 5s.
Case 1: Exactly three 5s. Product is 125, not divisible by 2. So 0 cases.
Case 2: Exactly two 5s. The third number must be even. Choices for the third number: 2, 4, 6.
There are 3 choices, and 3 positions for the third number. So $3 \times 3 = 9$ cases.
Case 3: Exactly one 5. The other two numbers cannot be 5, and at least one must be even.
Total ways to pick two numbers not 5 is $5 \times 5 = 25$.
Ways to pick two numbers not 5 and both odd (1, 3) is $2 \times 2 = 4$.
So ways to pick two numbers not 5 with at least one even is $25 - 4 = 21$.
Since the 5 can be in 3 positions, $3 \times 21 = 63$ cases.
Total cases = $0 + 9 + 63 = 72$.
\boxed{72}

*Analysis:* This trajectory is rambling and exhibits hesitation (high entropy), even though it eventually reaches the correct answer. It receives a low $\text{Score}_{LH}$, making it an ideal candidate to receive corrective signal.

**Pseudo-Target Teacher Prompt**

In pseudo-target routing, we construct the teacher prompt by inserting $y^+$ into the reference solution block. The teacher model is conditioned on this prompt to generate the distillation targets for $y^-$.

**Pseudo-Target Teacher Prompt**

Problem: Given that $1 \leq x, y, z \leq 6$, how many cases are there in which the product of natural numbers $x, y, z$ is divisible by 10?

Here is a reference solution to this problem:
=== Reference Solution Begin ===
The total number of cases for $(x, y, z)$ is $6 \times 6 \times 6 = 216$.
For $xyz$ to be divisible by 10, it must be divisible by 2 and by 5.
Let's find the complement: cases where $xyz$ is NOT divisible by 10.
This happens if $xyz$ is not divisible by 2 OR not divisible by 5.
Let $A$ be the set of cases where $xyz$ is not divisible by 2 (all must be odd: 1, 3, 5). $|A| = 3^3 = 27$.
Let $B$ be the set of cases where $xyz$ is not divisible by 5 (none can be 5: 1, 2, 3, 4, 6). $|B| = 5^3 = 125$.
The intersection $A \cap B$ is cases where numbers are odd and not 5 (must be 1, 3). $|A \cap B| = 2^3 = 8$.
By Principle of Inclusion-Exclusion, $|A \cup B| = 27 + 125 - 8 = 144$.
The number of valid cases is $216 - 144 = 72$.
\boxed{72}
=== Reference Solution End ===

After reading the reference solution above, make sure you truly understand the reasoning behind each step-do not copy or paraphrase it. Now, using your own words and independent reasoning, derive the same final answer to the problem above. Think step by step, explore different approaches, and don't be afraid to backtrack or reconsider if something doesn't work out:

Please reason step by step, and put your final answer within \boxed{}.

## H Additional Qualitative Case Studies

This section provides two additional case studies to deepen the understanding of trajectory-level routing dynamics: one demonstrating how Best-of-$N$ routing escapes local optima, and another analyzing a failure mode of our teacher-free proxy score.

### H.1 Case A: Escaping Local Optima via BoN Routing

A key advantage of Best-of-$N$ routing is its ability to bypass the base policy's default, yet flawed, reasoning biases.

**Problem**

A sequence is defined by $a_1 = 2$ and $a_{n+1} = \frac{a_n}{1+a_n}$. Find $a_{50}$.

**Base Policy Default (Greedy/High Probability, Often Flawed):**

**Base Default Rollout**

The base model frequently attempts to compute terms directly: $a_2 = \frac{2}{3}$, $a_3 = \frac{2/3}{5/3} = \frac{2}{5}$, $a_4 = \frac{2/5}{7/5} = \frac{2}{7}$. It then struggles to generalize the pattern formally, often hallucinating a closed form like $a_n = \frac{2}{2n-1}$ (which is correct) but making arithmetic errors when substituting $n = 50$, outputting \boxed{2/99} or getting lost in the fractions. This path typically results in high entropy towards the end.

**Selected Candidate via BoN (High $\mathrm{Score}_{LH}$):** Among $N = 8$ candidates, the router selects a trajectory that takes the reciprocal:

**Selected Rollout ($y^+$)**

We have $a_{n+1} = \frac{a_n}{1+a_n}$. Let's consider the reciprocal:
$\frac{1}{a_{n+1}} = \frac{1+a_n}{a_n} = \frac{1}{a_n} + 1$.
This means the sequence $b_n = \frac{1}{a_n}$ is an arithmetic progression with common difference 1.
Since $b_1 = \frac{1}{2}$, we have $b_n = b_1 + (n-1)(1) = \frac{1}{2} + n - 1 = n - \frac{1}{2} = \frac{2n-1}{2}$.
Therefore, $a_n = \frac{2}{2n-1}$.
For $n = 50$, $a_{50} = \frac{2}{100-1} = \frac{2}{99}$.
\boxed{2/99}

*Analysis:* This trajectory recognizes the mathematical trick (reciprocal sequence) early on. Because the resulting steps are deterministic and algebraically simple, the model maintains very low entropy and reaches the answer quickly. BoN routing successfully surfaces this optimal path for distillation, effectively transferring an advanced reasoning skill into the model's primary mode.

### H.2 Case B: Failure Mode – The "Confident Hallucination"

While $\mathrm{Score}_{LH}$ serves as a reliable proxy for teacher–student disagreement in the vast majority of cases, it exhibits a known edge case: confident hallucinations.

**Problem**

Is 51 a prime number?

**Confident Hallucination (High $\mathrm{Score}_{LH}$, but High Oracle $\mathrm{Score}_{HK}$):**

**Hallucinated Rollout**

To determine if 51 is prime, we check for divisibility by small primes.
51 ends in 1, so it is not divisible by 2.
The sum of its digits is $5 + 1 = 6$, which is divisible by 3.
Wait, $51/3 = 17$.
Actually, I miscalculated. 51 is not divisible by 3 because 51 is a known prime.
Therefore, 51 is prime.
\boxed{Yes}

*Analysis:* In this rare scenario, the model makes a glaring error but rapidly converges on a confident (though false) conclusion. Because the response is short and the token-level probability distribution at the end is heavily skewed towards "Yes" (low entropy), $\mathrm{Score}_{LH}$ assigns it a high sequence value. However, an Oracle teacher given the ground truth ("No, 51=3×17") would register a massive KL divergence ($\mathrm{Score}_{HK}$) on this trajectory. This highlights the fundamental limitation of teacher-free proxies: they rely on the student's internal calibration, which can occasionally fail when the student is confidently wrong. In the pseudo-target routing variant, such instances are mitigated by the sheer volume of correct routing decisions across the training batch, but they underscore why Oracle-guided OPSD remains the upper bound for performance.

**Algorithm 1** Selective On-Policy Self-Distillation (Selective OPSD)

**Require:** Reasoning dataset $\mathcal{S} = \{(x_i, y_i^\star)\}_{i=1}^{M}$;
language model $p_\theta$;
candidate count $N$; divergence $D$ (e.g., $\mathrm{JSD}_\beta$)

Let $p_S(\cdot \mid x)$ be the adapter-augmented student and $p_T(\cdot \mid x, y^\star)$ be the corresponding refreshed teacher with adapters disabled and privileged conditioning.
**while** not converged **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{S}$
  **for all** $(x, y^\star) \in \mathcal{B}$ **do**
    Sample $N$ candidate responses $\{\hat{y}^{(1)}, \ldots, \hat{y}^{(N)}\} \sim p_S(\cdot \mid x)$
    Compute length $L_j$ and entropy $H_j$ for each candidate $\hat{y}^{(j)}$
    Normalize lengths to $\hat{L}_j$ and entropies to $\hat{H}_j$ within the candidate group
    Compute routing score $S_j = (1-\hat{L}_j)(1-\hat{H}_j)$ for each candidate
    Select the optimal trajectory $\hat{y}^{\text{sel}} = \arg\max_j S_j$
    Compute the token-wise divergence along the selected rollout:

$$\begin{aligned}\ell(x, y^\star) \leftarrow & D(p_T \,\|\, p_S)(\hat{y}^{\text{sel}} \mid x) \\ = & \frac{1}{|\hat{y}^{\text{sel}}|} \sum_{n=1}^{|\hat{y}^{\text{sel}}|} D\big(p_T(\cdot \mid \hat{y}^{\text{sel}}_{<n}, x, y^\star) \\ & \big\|\, p_S(\cdot \mid \hat{y}^{\text{sel}}_{<n}, x)\big)\end{aligned}$$

  **end for**
  Calculate loss $\mathcal{L}_{\text{Selective OPSD}}(\theta) \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(x,y^\star)\in\mathcal{B}} \ell(x, y^\star)$ and update $\theta$
**end while**