CONTROLLED IN-ROLLOUT PRIVILEGED INFORMATION INTERVENTION FOR RELIABLE ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) trains a model on its own rollouts using token-level supervision from a privileged teacher instantiated from the same base model. In standard OPSD, the teacher receives the complete reference solution in its initial prompt. We study a specific failure mode of this design: full prompt-level privileged information (PI) can induce a concentrated teacher–student discrepancy near the beginning of a rollout, while its corrective influence weakens as an erroneous student prefix grows. We introduce Controlled In-Rollout Privileged-Information Intervention (CIPI), which periodically selects PI fragments by balancing their guidance benefits against the resulting teacher–student policy discrepancy, and inserts them into the teacher context along student-generated rollouts. Theoretical analysis provides conceptual motivation for balancing teacher guidance and teacher–student discrepancy in PI selection. Experiments with Qwen3-1.7B and Qwen3-4B on three competition-level mathematical reasoning benchmarks show that CIPI consistently outperforms OPSD, with macro Avg@12 gains of 3.3 and 5.1 percentage points (ppts) in non-thinking mode and 1.7 and 1.0 ppts in thinking mode, respectively. Ablation studies show that removing periodic in-rollout intervention while retaining prompt-level PI selection reduces average performance by 2.7 ppts, further confirming the effectiveness of in-rollout intervention beyond prompt-level PI selection.

1 INTRODUCTION

Post-training is central to improving the reasoning capabilities of large language models (LLMs). Existing approaches primarily include supervised fine-tuning Wei et al. (2021); Ouyang et al. (2022) (SFT), reinforcement learning with verifiable rewards Shao et al. (2024); Guo et al. (2025) (RLVR), and knowledge distillation Hinton et al. (2015); Kim and Rush (2016) (KD). Despite their success, these approaches have inherent limitations. SFT and conventional KD provide stable token-level supervision but typically rely on fixed expert- or teacher-generated trajectories, resulting in an off-policy mismatch between training and inference. RLVR instead optimizes trajectories sampled from the current policy, but its trajectory-level rewards provide sparse supervision and do not indicate which intermediate reasoning steps should be corrected.

On-policy distillation Agarwal et al. (2024) (OPD) combines the complementary advantages of these paradigms by allowing the student to generate trajectories from its current policy while receiving dense token-level supervision from a teacher on the resulting prefixes. It therefore preserves the state-distribution relevance of on-policy learning while providing substantially more informative supervision than trajectory-level outcome rewards. However, conventional OPD typically relies on a separate and stronger teacher model, introducing additional computational and model-maintenance costs. On-Policy Self-Distillation Zhao et al. (2026) (OPSD) further removes this requirement by instantiating the teacher and student from the same base model and conditioning only the teacher on privileged information (PI), such as a reference solution. OPSD thereby enables the model to learn from privileged supervision over its own on-policy behavior without requiring an external teacher.

However, eliminating the external teacher does not guarantee reliable and learnable supervision. In OPSD, PI is provided only in the initial teacher prompt. The teacher must therefore interpret

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.

an increasingly long student-generated prefix that may be inconsistent with the privileged reference. Prior studies also show that an LLM's ability to utilize relevant information may weaken as that information becomes increasingly distant in a growing context (Liu et al., 2024; Veseli et al., 2025). Consequently, along erroneous rollouts, the accumulating student prefix can progressively mislead the teacher and weaken its ability to provide corrective supervision. Meanwhile, exposing the complete reference at the beginning can make the teacher distribution locally much sharper or qualitatively different from the student distribution. The resulting signal may be correct yet difficult for the current student to absorb, a phenomenon related to capacity and distribution mismatch in knowledge distillation (Cho and Hariharan, 2019; Mirzadeh et al., 2020). As shown in Fig. 2, our diagnostic results indicate that OPSD indeed exhibits both failure patterns along erroneous student rollouts.

These observations motivate treating PI as a source of corrective guidance that should be managed throughout the rollout. A fragment useful at one prefix may provide little additional guidance at another, while repeatedly exposing the full reference can impose unnecessary policy discrepancy. Effective intervention therefore requires balancing the benefits of corrective teacher guidance against the resulting teacher–student policy discrepancy.

To improve teacher reliability while still preserving student teachability, we propose **C**ontrolled **I**n-Rollout **P**rivileged-Information **I**ntervention (CIPI). At each periodic intervention point along a student-generated rollout, CIPI selects and introduces PI fragments into the teacher context to correct supervision degraded by preceding student errors. CIPI selects each PI fragment by jointly considering its additional guidance benefit and the resulting teacher–student policy discrepancy. We provide a conceptual motivation for this trade-off through a local surrogate analysis: in a fixed reference-supervision model, the intervention-dependent part of a one-step improvement bound has the same form as our selection objective. Experiments with Qwen3-1.7B and Qwen3-4B on three competition-level mathematical reasoning benchmarks show that CIPI consistently outperforms OPSD, with macro Avg@12 gains of 3.3 and 5.1 ppts in non-thinking mode and 1.7 and 1.0 ppts in thinking mode, respectively. Ablation studies further show that removing periodic in-rollout intervention while retaining prompt-level PI selection reduces performance by 2.7 ppts on average, validating its effectiveness in maintaining reliable teacher supervision.

Our main contributions are summarized as follows:

- We identify a critical limitation of OPSD. Despite access to a correct reference solution, the teacher can be progressively misled by erroneous student-generated prefixes, leading to increasing perplexity on the correct solution and degraded corrective supervision.
- We propose **C**ontrolled **I**n-Rollout **P**rivileged Information **I**ntervention (CIPI), which periodically selects and introduces PI fragments into teacher context by balancing guidance benefits against the resulting distillation difficulty, thereby maintaining reliable teacher supervision while controlling the teacher–student policy discrepancy.
- We motivate the guidance–discrepancy trade-off through a local surrogate analysis under fixed reference supervision. Extensive experiments across model scales and mathematical reasoning benchmarks demonstrate the effectiveness of CIPI.

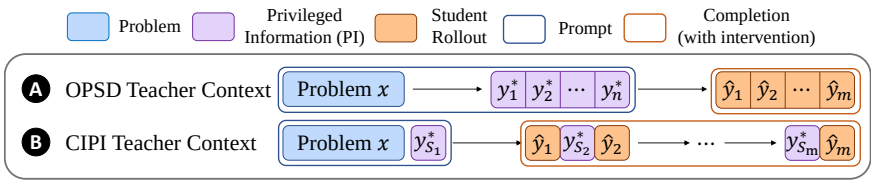

Figure 1: **Teacher context comparison between OPSD and CIPI.** OPSD places the complete PI in the prompt, whereas CIPI periodically inserts selected PI fragments into the student rollout to maintain corrective teacher guidance.

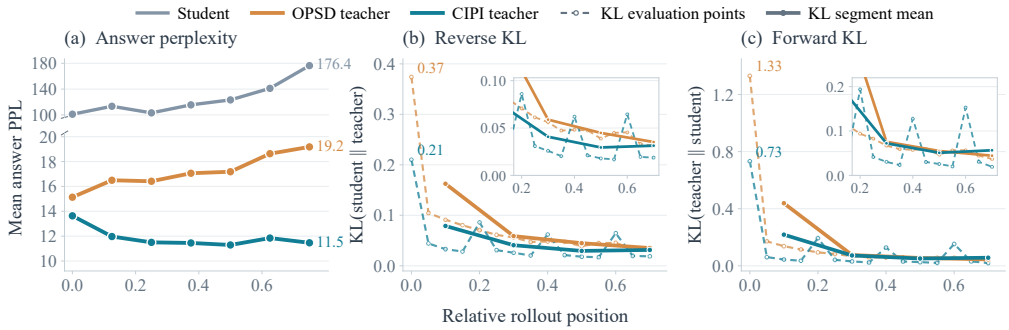

Figure 2: **Left:** Along incorrect student trajectories, the OPSD teacher exhibits progressively increasing perplexity on the correct solution, whereas CIPI maintains consistently lower teacher perplexity. **Middle and right:** Providing the complete PI in the initial OPSD prompt induces a large forward and reverse KL peak at the beginning of the rollout. By periodically introducing partial PI fragments, CIPI distributes the intervention-induced divergence across the trajectory as a sequence of smaller peaks, avoiding excessively large local discrepancies that may lead to overly aggressive policy updates. Insets magnify the later rollout positions.

2 Preliminaries

2.1 On-Policy Self-Distillation

Let $\mathcal{D} = \{(x, y^*)\}$ denote a reasoning dataset, where x is a problem and y^* is the corresponding privileged information (PI), such as the ground truth solution. Given x, the student policy generates an on-policy trajectory

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim p_S(\cdot \mid x) := p_\theta(\cdot \mid x).$$

OPSD instantiates the teacher and student policies from the same model but provides PI only to the teacher, i.e., $p_T(\cdot \mid x, y^*) := p_\theta(\cdot \mid x, y^*)$. Afterwards, the teacher provides supervision on the rollout generated by the student. The learning process is captured by the loss

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\mathbb{E}_{\hat{y}\sim p_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \ell_t^{\text{KD}}\right], \quad \ell_t^{\text{KD}} = D_{\text{KL}}(q_{\bar{\theta}}(\cdot \mid x, y^\star, \hat{y}_{<t}) \,\|\, p_\theta(\cdot \mid x, \hat{y}_{<t}))\,.$$

where D_{KL} is the KL divergence.

3 Motivation

Although OPSD provides the teacher with a ground-truth solution, the teacher can still be misled by erroneous student- generated prefixes. This section presents a controlled study to examine how teacher reliability and teacher–student policy discrepancy evolve along student rollouts. We then investigate whether intervening with partial PI fragments throughout the rollout can mitigate these problems.

Experimental setup. We sample 1,000 problems from the OpenThoughts-Math-30K Guha et al. (2025) training set and use Qwen3-1.7B to generate one rollout for each problem. Since automatic answer verification may incorrectly reject semantically equivalent answer forms, we use `gpt-3.5-turbo` to re-evaluate each rollout and retain only those confirmed to be incorrect. This produces 361 erroneous student trajectories.

For each trajectory, we partition the student rollout into M approximately equal-length segments and the PI into N approximately equal-length fragments:

$$\hat{y} = \hat{y}_1 \oplus \cdots \oplus \hat{y}_M, \qquad y^* = y_1^* \oplus \cdots \oplus y_N^*.$$

The token-level partition points are locally adjusted to nearby sentence boundaries to preserve semantic completeness. After the i-th rollout segment, the student and OPSD teacher contexts are respectively

$$c_i^S = x \oplus \hat{y}_{1:i}, \qquad c_i^{\mathrm{OPSD}} = x \oplus y^* \oplus \hat{y}_{1:i},$$

where $\hat{y}_{1:i} = \hat{y}_1 \oplus \cdots \oplus \hat{y}_i$.

CIPI selects PI fragments before each rollout segment and retains the selected fragments in all subsequent teacher contexts. Let S_i denote the set of PI fragments selected at the i-th intervention point, and define

$$y^*_{S_i} := \bigoplus_{j \in S_i} y^*_j, \qquad S_i \subseteq \{1, \ldots, N\}.$$

The CIPI teacher context after the i-th rollout segment is then

$$c_i^{\mathrm{CIPI}} = x \oplus y^*_{S_1} \oplus \hat{y}_1 \oplus \cdots \oplus y^*_{S_i} \oplus \hat{y}_i.$$

Figure 1 illustrates the difference between the teacher contexts constructed by OPSD and CIPI, and the principle of PI selection is detailed in Section 4.1.

We evaluate each context using answer perplexity and teacher–student KL divergence. After each rollout segment, we append the template-formatted ground-truth answer a^* and compute

$$\mathrm{PPL}_i^X = \mathrm{PPL}\left(a^* \mid c_i^X\right), \qquad X \in \{S, \mathrm{OPSD}, \mathrm{CIPI}\},$$

where only the answer tokens are included in the perplexity calculation. For each teacher method $X \in \{\mathrm{OPSD}, \mathrm{CIPI}\}$, we also compute the average teacher–student divergence over the i-th rollout segment:

$$\mathrm{KL}_i^X = \operatorname*{Avg}_{n \in \hat{y}_i} D_{\mathrm{KL}}\left(p_T^X(\cdot \mid c_{i,n}^X) \,\middle\|\, p_S(\cdot \mid c_{i,n}^S)\right).$$

We report both forward and reverse KL by exchanging the order of the teacher and student distributions.

Observation 1: Teacher supervision degrades along erroneous student rollouts. As shown in the left panel of Figure 2, the answer PPL of the OPSD teacher exhibits a clear upward trend as it observes more student-generated segments. This indicates that the teacher is progressively misled by the student's output and becomes less capable of providing reliable corrective supervision. In contrast, CIPI maintains substantially lower and more stable answer PPL by periodically introducing selected PI fragments, thereby preserving the corrective influence of PI throughout the rollout.

Observation 2: Periodic PI intervention reduces excessive local policy shifts. As shown in the middle and right panels of Figure 2, providing the complete PI in the initial OPSD prompt induces an extremely large forward and reverse KL peak at the beginning of the rollout. Such a concentrated teacher–student discrepancy may dominate the distillation objective at early positions and lead to overly aggressive policy updates, undermining training stability. CIPI instead periodically releases selected PI fragments throughout the rollout. Each intervention produces a local increase in KL, but its magnitude is substantially smaller than the initial peak induced by OPSD. Consequently, CIPI replaces a single extreme policy shift with a sequence of moderate shifts distributed across the trajectory, providing a smoother and more controlled distillation signal. This behavior suggests that CIPI's advantage comes not merely from stronger supervision, but from regulating when and how that supervision is delivered throughout the rollout.

4 METHOD

4.1 CONCEPTUAL MOTIVATION FOR PI SELECTION

The observations in Section 3 suggest balancing corrective teacher guidance against the discrepancy of the supervision the student must absorb. We first give a conceptual motivation for this trade-off through a local model of distillation with imperfect teacher supervision. Let $S = (S_1, \ldots, S_M)$ denote an intervention scheme, where S_i is the PI fragment set inserted before the i-th student rollout segment. Fix the current student's normalized rollout-prefix distribution $\mu = d^{p_\theta}$ and a ideal reference supervision distribution $r(\cdot \mid h)$ shared across intervention schemes. Let q^S be the

teacher induced by scheme S, and let q^0 be a fixed baseline teacher. In this model, teacher error and teacher–student discrepancy are measured by

$$E_{\mathrm{teach}}(S) = \mathbb{E}_{h\sim\mu} D_{\mathrm{KL}}\big(r(\cdot \mid h) \,\|\, q^S(\cdot \mid h)\big)\,, C_{\mathrm{distill}}(S) = \mathbb{E}_{h\sim\mu} D_{\mathrm{KL}}\big(q^S(\cdot \mid h) \,\|\, p_\theta(\cdot \mid h)\big)\,.$$

The guidance gain $G_{\mathrm{teach}}(S) = E_{\mathrm{teach}}(0) - E_{\mathrm{teach}}(S)$ measures the reduction in reference supervision error. To examine how this gain might benefit the student, consider its agreement with the same reference:

$$J_{\mathrm{sur}}(\vartheta) = \mathbb{E}_{h\sim\mu} \sum_a r(a \mid h) \log p_\vartheta(a \mid h).$$

All prefix and supervision distributions are held fixed during a single forward-KL gradient step with learning rate $\eta > 0$:

$$\theta_S^+ = \theta - \eta g_S, \qquad g_S = \nabla_\vartheta \mathcal{L}_{\mathrm{KD}}(q^S, p_\vartheta)\big|_{\vartheta=\theta}\,,$$

where $\mathcal{L}_{\mathrm{KD}}(q^S, p_\vartheta) = \mathbb{E}_{h\sim\mu} D_{\mathrm{KL}}(q^S(\cdot \mid h) \| p_\vartheta(\cdot \mid h))$.

Theorem 1 *Suppose J_{sur} is L-smooth on a neighborhood containing the one-step update paths, the above expectations and divergences are finite, differentiation under the fixed expectations is valid, and the student log-probability gradient satisfies*

$$\|\nabla_\theta \log p_\theta(a \mid h)\|_2 \le B$$

for some $B > 0$ and all relevant prefixes and tokens. Then the student's one-step surrogate improvement satisfies

$$J_{\mathrm{sur}}(\theta_S^+) - J_{\mathrm{sur}}(\theta) \ge \eta b_\theta + \eta B^2 G_{\mathrm{teach}}(S) - L\eta^2 B^2 C_{\mathrm{distill}}(S),$$

where

$$b_\theta = \frac{1}{2}\,\|\nabla J_{\mathrm{sur}}(\theta)\|_2^2 - B^2 E_{\mathrm{teach}}(0)$$

is independent of the intervention scheme S.

Teacher error controls the deviation from the reference-supervision gradient, while teacher–student KL bounds the update magnitude entering the smoothness remainder. The proof is given in Appendix C.1. Since b_θ is independent of S, the bound yields the following selection principle.

Corollary 1 (PI selection in the local surrogate model) *Under the conditions of Theorem 1, let $\lambda_\eta = L\eta$. Then*

$$S^* \in \arg\max_S \left[G_{\mathrm{teach}}(S) - \lambda_\eta C_{\mathrm{distill}}(S)\right]$$

maximizes the derived lower bound on one-step improvement in J_{sur}.

Although directly obtaining S^* remains challenging, Corollary 1 provides a conceptual basis for the objective: balance the improvement in teacher guidance against the induced teacher–student policy discrepancy. This trade-off guides the practical PI-selection objective developed next.

4.2 PI Selection and Distillation

Guided by Corollary 1, we formulate CIPI as selecting an intervention scheme that balances teacher guidance gain against teacher–student discrepancy:

$$\max_S \; G_{\mathrm{teach}}(S) - \lambda C_{\mathrm{distill}}(S). \tag{1}$$

This objective favors PI that brings the teacher closer to the ideal reference supervision while controlling the teacher–student policy discrepancy. Its guidance term G_{teach}, however, depends on the reference distribution r, which is unavailable during training. We therefore use the increase in the teacher's correct-answer log-likelihood, evaluated after the same student rollout segment, as a practical proxy for guidance gain. The intuition is that PI that helps the teacher retain the correct conclusion despite a misleading student prefix may also improve its supervision along that rollout. Appendix C.2 gives a sufficient condition for this relationship under a simple mixture model of reference and degraded reasoning modes. The discrepancy term C_{distill} is computed using the teacher–student KL defined above. Like the reference-dependent guidance gain, the theoretical coefficient

	Method	Qwen3-1.7B				Qwen3-4B			
		AIME24	AIME25	HMMT25	Overall	AIME24	AIME25	HMMT25	Overall
Nonthinking	Base	11.9	9.2	5.0	8.7	23.1	21.4	10.8	18.4
	SFT	13.3	9.7	6.9	10.0	25.6	23.6	12.8	20.7
	GRPO	15.3	11.7	7.3	11.4	25.0	21.3	13.1	19.8
	OPSD	15.1±0.1	10.2±0.6	7.5±0.3	10.9±0.1	29.6±2.8	22.0±0.2	15.1±1.6	22.2±1.3
	CIPI	**19.2±0.6**	**15.2±0.3**	**8.2±0.4**	**14.2±0.2**	**35.8±0.6**	**28.2±0.9**	**18.0±0.2**	**27.3±0.4**
Thinking	Base	51.5	36.7	23.1	37.1	74.9	66.4	42.2	61.2
	SFT	48.4	36.3	22.7	35.8	70.2	62.3	43.4	58.6
	GRPO	51.1	38.3	23.7	37.7	75.6	68.1	44.4	62.7
	OPSD	56.0±1.3	42.7±1.1	27.5±1.6	42.0±1.3	75.9±0.4	68.3±0.3	45.1±1.2	63.1±0.5
	CIPI	**58.4±1.4**	**43.8±0.4**	**28.8±0.6**	**43.7±0.5**	**76.6±0.9**	**69.1±0.4**	**46.5±0.5**	**64.1±0.2**

Table 1: Performance on competition-level mathematical reasoning benchmarks, measured by average accuracy (Avg@12, %). "Overall" denotes the macro-average across the three benchmarks. OPSD and CIPI report mean ± sample standard deviation over three runs.

$\lambda_\eta = L\eta$ is difficult to determine in practice, since the local smoothness constant L is unknown. We therefore select λ based on validation-set performance to calibrate the trade-off between the answer-likelihood proxy and policy discrepancy.

Using these practical scores, we optimize a local objective at each intervention point:

$$F_i(S_i) = \ell_i(S_i; c_{i-1}^{\mathrm{CIPI}}) - \lambda d_i(S_i; c_{i-1}^{\mathrm{CIPI}}).$$

We hold the preceding context fixed and optimize F_i greedily. This is a myopic choice: inserted fragments persist and affect later teacher contexts, but the local score evaluates only the current segment. Appendix A.2 analyzes greedy approximation to the best subset for this fixed local objective under explicit assumptions; it does not bound the loss from replacing the trajectory-level objective with local selection. At the i-th intervention point, initializing S_i from an empty set, the solver repeatedly selects the fragment with the largest positive marginal score:

$$\begin{aligned} j^* &= \arg\max_{j \notin S_i} \{\Delta\ell_i(j \mid S_i) - \lambda \Delta d_i(j \mid S_i)\}, \quad \text{where} \\ \Delta\ell_i(j \mid S_i) &= \ell_i(S_i \cup \{j\}; c_{i-1}^{\mathrm{CIPI}}) - \ell_i(S_i; c_{i-1}^{\mathrm{CIPI}}), \\ \Delta d_i(j \mid S_i) &= d_i(S_i \cup \{j\}; c_{i-1}^{\mathrm{CIPI}}) - d_i(S_i; c_{i-1}^{\mathrm{CIPI}}), \end{aligned}$$

and updates $S_i \leftarrow S_i \cup \{j^*\}$ only if the corresponding score is positive. Here we use $\ell_i(S_i; c_{i-1}^{\mathrm{CIPI}})$ to denote the teacher's average log-likelihood of the ground-truth answer after observing the current rollout segment $\hat{y}_i$, with PI fragments $y_{S_i}^*$ inserted before $\hat{y}_i$, and use $d_i(S_i; c_{i-1}^{\mathrm{CIPI}})$ to denote the average teacher–student divergence on the current rollout segment $\hat{y}_i$ under the same teacher context.

With the constructed CIPI teacher context, the student is trained using the same token-level on-policy distillation objective as OPSD:

$$\mathcal{L}_{\mathrm{CIPI}}(\theta) = \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \mathbb{E}_{\hat{y}\sim p_S(\cdot|x)} \sum_{i=1}^{M} \sum_{n\in\hat{y}_i} D\left(p_T^{\mathrm{CIPI}}\left(\cdot \mid c_{i,n}^T\right) \| p_S\left(\cdot \mid c_{i,n}^S\right)\right),$$

where $c_{i,n}^T$ and $c_{i,n}^S$ are the teacher and student contexts at token position n in segment $\hat{y}_i$.

5 Experiments

5.1 Experimental Setup

Models and datasets. We evaluate Qwen3-1.7B and Qwen3-4B Team (2025) under thinking and non-thinking configurations. Following OPSD, the *thinking* configuration uses a thinking-mode teacher, non-thinking student rollouts, and thinking-mode evaluation. A matched *non-thinking*

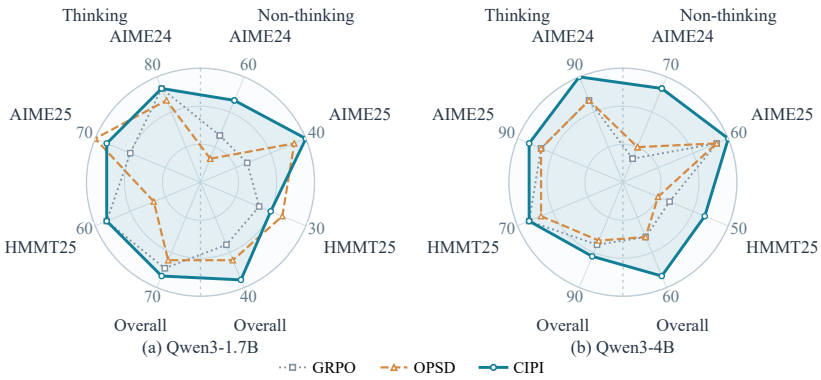

Figure 3: Pass@12 (%) of Qwen3-1.7B (left) and Qwen3-4B (right) in thinking and non-thinking modes. Overall averages the three benchmarks. Axis maxima are labeled; centers are 30 percentage points lower, with 10 points per ring.

configuration, with non-thinking mode used for the teacher, student rollouts, and evaluation, tests whether CIPI's benefits extend beyond this mode asymmetry. Following prior work Zhao et al. (2026); Han et al. (2026); Tan and Hong (2026), we train on the mathematical reasoning subset of OpenThoughts Guha et al. (2025) and evaluate on three competition-level mathematics benchmarks: AIME 2024, AIME 2025, and HMMT 2025.

Baselines. Under the same training settings, we compare CIPI with **SFT** on expert trajectories, **GRPO** Shao et al. (2024) with verified binary outcome rewards, and **OPSD** Zhao et al. (2026), which provides PI only in the initial teacher prompt. Ablations additionally compare with **PAINT** Tan and Hong (2026) and **ATESD** Han et al. (2026), which adapt prompt-level PI to reduce the teacher–student policy gap without in-rollout intervention.

Implementation details. Following OPSD Zhao et al. (2026), we fix the teacher to the initial policy and use full-vocabulary logit distillation. We train with LoRA Hu et al. (2021) on two NVIDIA H20 GPUs; hyperparameters are provided in Appendix B.1.

5.2 Overall Performance

CIPI consistently achieves the strongest overall performance. Table 1 shows that CIPI achieves the highest macro Avg@12 in all four model–mode settings and outperforms OPSD in all twelve model–mode–benchmark comparisons, supporting its effectiveness across model scales and evaluation modes.

CIPI improves exploration of successful reasoning paths. Figure 3 shows that CIPI achieves the highest macro Pass@12 in all four settings. Relative to OPSD, the gains are 5.6 and 11.1 percentage points on Qwen3-1.7B and Qwen3-4B in non-thinking mode, and 4.5 points on both models in thinking mode. With the same budget of 12 samples, CIPI thus finds at least one correct solution for more problems. A plausible explanation, consistent with Section 3, is that in-rollout PI intervention preserves the teacher's corrective guidance on erroneous student prefixes, while controlling teacher–student discrepancy keeps this guidance learnable. Distilling these token-level signals may help the student move away from failing reasoning paths and explore successful continuations during subsequent sampling, increasing the chance of finding a correct solution.

The gains are larger in non-thinking mode. CIPI improves macro Avg@12 over OPSD by 3.3 and 5.1 percentage points on Qwen3-1.7B and Qwen3-4B in non-thinking mode, compared with 1.7 and 1.0 points in thinking mode. Together with the larger Pass@12 gains, this suggests that corrective PI guidance is especially valuable without explicit thinking-mode reasoning.

5.3 Ablation: In-Rollout Intervention and Its Control

Effectiveness of In-rollout Intervention. To isolate in-rollout intervention (IRI), we retain controlled PI selection only in the initial prompt. In Figure 4(a,b), CIPI without IRI achieves macro

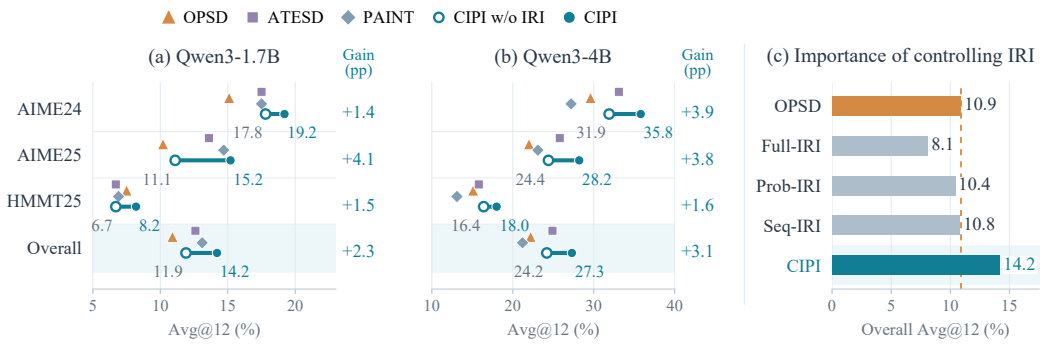

Figure 4: Ablations in non-thinking mode, measured by Avg@12 (%). **(a,b)** Adaptive-prompt comparisons on Qwen3-1.7B and Qwen3-4B. Open and filled teal circles denote CIPI without and with in-rollout intervention (IRI); connectors link their scores and "Gain" reports differences in percentage points. Baseline markers are staggered vertically. **(c)** Alternative IRI strategies on Qwen3-1.7B; the dashed line marks OPSD. "Overall" averages the three benchmarks.

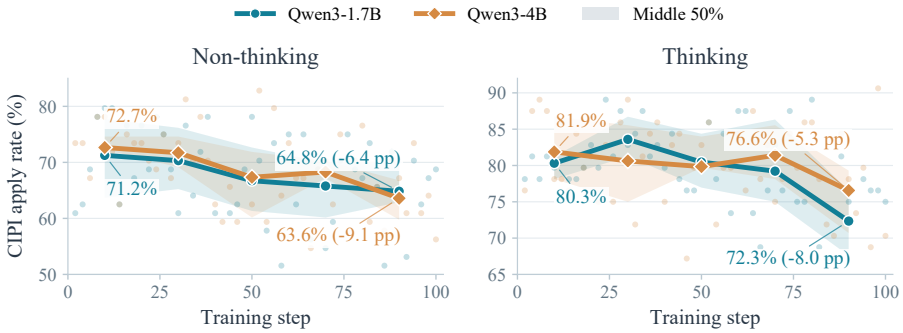

Figure 5: CIPI apply rates over the first 100 training steps. Dots show logged values; lines and shaded bands show means and interquartile ranges within non-overlapping 20-step windows. Endpoint labels report the first and last window means, with their changes in percentage points in parentheses.

Avg@12 scores of 11.9 and 24.2 on Qwen3-1.7B and Qwen3-4B, outperforming OPSD but showing no consistent advantage over ATESD and PAINT. Adding IRI raises these scores to 14.2 and 27.3, respectively, demonstrating the benefit of intervention within student rollouts.

Importance of controlling PI selection. We compare three simple IRI strategies on Qwen3-1.7B with eight PI fragments and four insertion positions, including the initial prompt. FULL-IRI inserts all fragments at every position; PROB-IRI independently inserts each fragment at each position with probability 0.25, matching one complete reference solution in expectation; SEQ-IRI inserts two fragments per position in their original order. Their macro Avg@12 scores are 8.1, 10.4, and 10.8, respectively, versus 10.9 for OPSD and 14.2 for CIPI (Figure 4(c)). These results highlight the need to select PI that balances corrective guidance against teacher–student discrepancy. Benchmark-level results are in Appendix B.4.

5.4 ON-DEMAND INTERVENTION DURING TRAINING

CIPI activates IRI only when the teacher's log-likelihood of the correct answer shows a decreasing trend along a student rollout, indicating degraded teacher guidance; otherwise, it retains standard OPSD supervision (Appendix B.2).

Figure 5 shows that the apply rate, the fraction of rollouts activating CIPI, declines during training. Between steps 1–20 and 81–100, its mean drops by 6.4 and 9.1 percentage points for Qwen3-1.7B and Qwen3-4B in non-thinking mode, and by 8.0 and 5.3 points in thinking mode, respectively.

This is consistent with improving student capability: more reliable rollouts are less likely to mislead the teacher, reducing intervention frequency while retaining corrective guidance where needed. Appendix B.5 ablates this reliability gate and examines its effect on downstream accuracy.

5.5 Sensitivity to the Trade-off Coefficient

Corollary 1 motivates $\lambda = L\eta$ in Eq. (1). Since the local smoothness constant L depends on the unknown reference distribution r, we tune λ for Qwen3-1.7B in non-thinking mode on 100 held-out OpenThoughts problems, evaluating checkpoints after 50 training steps using Avg@6 (six solutions per problem). In Figure 6, the horizontal lines mark the base model (53.8%) and OPSD baseline (55.5%). Without KL control ($\lambda = 0$), accuracy drops to 53.4%, below base; a weak penalty ($\lambda = 0.1$) yields 55.1%, below OPSD. These results reinforce Section 5.3: IRI alone is insufficient, and controlling teacher–student discrepancy is important for its benefits. By contrast, $\lambda = 0.7$ achieves the best score of 58.0%, while $\lambda \in [0.5, 0.9]$ maintains high accuracy (57.3%–58.0%), suggesting robustness to the choice of λ within this range. We fix $\lambda = 0.7$ for the main experiments. These selection runs are separate from the independent training runs in Table 1.

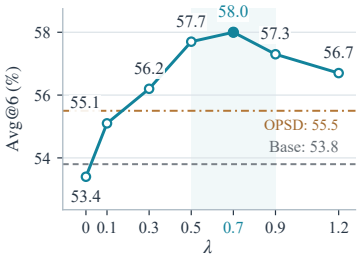

Figure 6: OpenThoughts validation Avg@6 (%) after 50 training steps, with base and OPSD reference lines.

6 Related Work

Training on fixed teacher-generated trajectories can introduce a mismatch with student-generated prefixes at inference time. Generalized Knowledge Distillation Agarwal et al. (2024) reduces this mismatch by supervising the student on its own generated sequences. Recent OPD studies further show that on-policy token-level supervision can provide denser and more stable learning signals than sparse outcome rewards, but also reveal that teacher feedback may become unreliable when student-generated prefixes deviate from the teacher's familiar distribution Fu et al. (2026); Li et al. (2026); Liu et al. (2026a;b).

On-Policy Self-Distillation (OPSD) Zhao et al. (2026) extends this paradigm by removing the need for an external teacher. It uses the same model as both student and teacher, where the student observes only the problem while the teacher additionally receives privileged information such as the reference solution. However, existing OPSD methods provide PI only in the initial teacher prompt. As the student rollout grows, erroneous prefixes can weaken the corrective influence of the initial PI and mislead the teacher. CIPI selects and inserts PI fragments during the rollout to preserve corrective guidance while controlling teacher–student discrepancy. Closely related adaptive-prompt methods, such as ATESD Han et al. (2026) and PAINT Tan and Hong (2026), adjust how much PI is exposed in the teacher prompt, whereas our CIPI further studies how PI should be controlled and refreshed within the rollout.

7 Conclusion

We present CIPI, a controlled in-rollout privileged information intervention framework for improving On-Policy Self-Distillation. We show that the OPSD teacher can be progressively misled by erroneous student-generated prefixes, even when privileged information is provided in the initial prompt. CIPI addresses this issue by periodically selecting and inserting PI fragments during the rollout, balancing improved teacher guidance against the resulting teacher–student policy discrepancy. A local surrogate analysis provides conceptual motivation for the guidance–discrepancy trade-off in this objective. Empirically, CIPI consistently outperforms OPSD on competition-level mathematical reasoning benchmarks, while ablation and adaptive-triggering studies confirm the effectiveness and flexibility of controlled in-rollout intervention.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims and assist in the writing of proofs, specifically to refine theorem statements and proof arguments under theoretical directions specified by us. We also used gpt-3.5-turbo to re-evaluate model-generated rollouts in the diagnostic analysis in Section 3. Additionally, we used generative AI tools to create or modify scientific figures or images and edit a research paper to improve readability, specifically to improve figure presentation and polish the manuscript for clarity and grammar. We have reviewed all AI-assisted work. In particular, we checked the mathematical claims and proofs for correctness, verified the accuracy of the figures, and reviewed textual revisions to ensure that they preserve the intended scientific meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, 2022.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327, 2016.

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*, 2024.

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.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. Positional biases shift as inputs approach context window limits. *arXiv preprint arXiv:2508.07479*, 2025.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

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

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zihao Han, Tiangang Zhang, Huaibin Wang, and Yilun Sun. Adaptive teacher exposure for self-distillation in llm reasoning. *arXiv preprint arXiv:2605.11458*, 2026.

Zhiquan Tan and Yinrong Hong. Paint: Partial-solution adaptive interpolated training for self-distilled reasoners. *arXiv preprint arXiv:2604.26573*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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.

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

Yanjiang Liu, Jie Lou, Xinyan Guan, Yuqiu Ji, Hongyu Lin, Ben He, Xianpei Han, Le Sun, Xing Yu, and Yaojie Lu. Your teacher can't help you here: Combating supervision fidelity decay in on-policy distillation. *arXiv preprint arXiv:2605.30833*, 2026a.

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

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

A Methods

A.1 Detailed Computation of PI Selection

This section provides the detailed computation of the practical selection objective introduced in Section 4.2 and summarizes the corresponding greedy selection procedure.

At the i-th intervention point, the previous CIPI context c_{i-1}^{CIPI} already contains the problem, all preceding student rollout segments, and the PI fragments introduced at earlier intervention points. Given a candidate fragment set $S_i \subseteq \{1, \ldots, N\}$, we temporarily insert the corresponding PI fragments before the current rollout segment $\hat{y}_i$ and evaluate the resulting teacher context from two complementary perspectives: whether it improves the teacher's guidance and whether the induced teacher distribution remains sufficiently close to the current student distribution.

Answer log-likelihood. We approximate teacher guidance quality using the teacher's log-likelihood of the ground-truth answer. Specifically, after the teacher observes the current rollout segment under the candidate intervention, we autoregressively evaluate the ground-truth answer and average its token-level log-probabilities:

$$\ell_i(S_i; c_{i-1}^{\mathrm{CIPI}}) = \frac{1}{|\mathrm{ans}|} \sum_{m=1}^{|\mathrm{ans}|} \log p_T\Big(\mathrm{ans}_m \,\Big|\, c_{i-1}^{\mathrm{CIPI}} \oplus y^*_{S_i} \oplus \hat{y}_i \oplus \mathrm{ans}_{<m}\Big). \tag{2}$$

Here, $\mathrm{ans} = (\mathrm{ans}_1, \ldots, \mathrm{ans}_{|\mathrm{ans}|})$ denotes the template-formatted ground-truth answer. Only the answer tokens are included in the calculation. Consequently, a larger ℓ_i indicates that the intervened

Algorithm 1 Greedy PI Selection at Intervention Point i

Require: Previous CIPI context c_{i-1}^{CIPI}, current rollout segment $\hat{y}_i$, PI fragments $\{y_1^*, \ldots, y_N^*\}$, trade-off coefficient λ, maximum number of selected fragments K

$A \leftarrow \varnothing$
while $|A| < K$ **do**
  for each $j \notin A$ **do**
    Compute answer log-likelihood gain $\Delta\ell_i(j \mid A)$
    Compute KL divergence increase $\Delta d_i(j \mid A)$ on $\hat{y}_i$
    $g_i(j \mid A) \leftarrow \Delta\ell_i(j \mid A) - \lambda \Delta d_i(j \mid A)$
  end for
  $j^\star \leftarrow \arg\max_{j \notin A} g_i(j \mid A)$
  if $g_i(j^\star \mid A) \leq 0$ **then**
    break
  end if
  $A \leftarrow A \cup \{j^\star\}$
end while
return $S_i = A$

teacher assigns higher probability to the correct answer after observing the same student-generated segment. Normalizing by answer length gives a per-token score. Appendix C.2 provides a sufficient condition under which an increase in this score also reduces the teacher's reference supervision error on rollout prefixes.

Teacher–student divergence. To quantify the distillation difficulty introduced by the same candidate intervention, we compute the full-vocabulary KL divergence between the teacher and student predictive distributions at every token position of $\hat{y}_i$:

$$d_i(S_i; c_{i-1}^{\mathrm{CIPI}}) = \frac{1}{|\hat{y}_i|} \sum_{n \in \hat{y}_i} D_{\mathrm{KL}}\left(p_T\Big(\cdot \;\Big|\; c_{i-1}^{\mathrm{CIPI}} \oplus y_{S_i}^* \oplus \hat{y}_{i,<n} \Big) \;||\; p_S\Big(\cdot \;\Big|\; c_{i-1}^{\mathrm{S}} \oplus \hat{y}_{i,<n} \Big) \right). \quad (3)$$

The teacher distribution is conditioned on the CIPI context containing the candidate PI fragments, whereas the student distribution is conditioned only on the original student-visible context. Both distributions are evaluated at the same rollout token position. We average the divergence over all tokens in the current segment so that d_i measures the local policy discrepancy induced by the intervention independently of the segment length. A larger value therefore indicates that the resulting teacher supervision may be more difficult for the current student to absorb.

For a candidate fragment $j \notin S_i$, its marginal guidance gain and marginal divergence cost are obtained by comparing the above quantities before and after adding j: $\Delta\ell_i(j \mid S_i)$ measures the increase in answer log-likelihood, while $\Delta d_i(j \mid S_i)$ measures the accompanying increase in teacher–student divergence. Importantly, these marginal quantities are recomputed after every selected fragment. The selection procedure can therefore account for redundancy and interaction among PI fragments rather than scoring each fragment independently.

Greedy selection. Jointly enumerating all PI-fragment subsets at every intervention point would be prohibitively expensive during on-policy training. We instead use the greedy procedure summarized in Algorithm 1.

The algorithm initializes the selected set as empty and evaluates every remaining PI fragment according to its marginal selection score, which balances the improvement in answer log-likelihood against the additional teacher–student discrepancy. It then adds the fragment with the largest score and recomputes all marginal scores conditioned on the updated selected set. The procedure terminates when the best remaining fragment has a non-positive score or when the maximum number of selected fragments K is reached. In all our experiments, we set K equal to the total number of PI fragments N, so the number of inserted fragments is not explicitly constrained by an artificial selection budget.

The resulting set is used as S_i and inserted into the teacher context before $\hat{y}_i$. The constructed context, including the newly selected fragments, is subsequently retained when processing later rollout segments. In this way, the local greedy solver progressively constructs the complete intervention scheme $\mathcal{S} = (S_1, \ldots, S_M)$ along the student trajectory. At each intervention point, it performs at most K selection rounds and evaluates at most N candidate fragments per round, avoiding enumeration over the 2^N possible fragment subsets.

A.2 Greedy Approximation of the Local Proxy Objective

We analyze the approximation error of the greedy PI-selection procedure for the answer-likelihood–divergence proxy at a fixed intervention point. Let $\mathcal{V} = \{1, \ldots, N\}$ denote the set of candidate PI fragments. For a fixed intervention point i and its preceding CIPI context c_{i-1}^{CIPI}, define the local selection objective as

$$F_i(S) = \ell_i(S; c_{i-1}^{\mathrm{CIPI}}) - \lambda d_i(S; c_{i-1}^{\mathrm{CIPI}}), \qquad S \subseteq \mathcal{V}.$$

For clarity, we suppress the fixed context c_{i-1}^{CIPI} in the following analysis.

We define the normalized objective

$$f_i(S) = F_i(S) - F_i(\varnothing),$$

such that $f_i(\varnothing) = 0$. This normalization does not affect the greedy selection because f_i and F_i have identical marginal gains:

$$\Delta_i(j \mid S) := f_i(S \cup \{j\}) - f_i(S) = F_i(S \cup \{j\}) - F_i(S).$$

Accordingly,

$$\Delta_i(j \mid S) = \Delta\ell_i(j \mid S) - \lambda \Delta d_i(j \mid S),$$

which is exactly the marginal score used in Algorithm 1.

Assumption 1 (Local weak submodularity) *For each fixed intervention point i and its preceding context c_{i-1}^{CIPI}, the normalized local objective f_i is γ_i-weakly submodular for some $\gamma_i \in (0, 1]$. Specifically, for any $A \subseteq \mathcal{V}$ and $B \subseteq \mathcal{V} \setminus A$ with $|A| \leq K$ and $|B| \leq K$,*

$$\sum_{j \in B} \Delta_i(j \mid A) \geq \gamma_i \left[f_i(A \cup B) - f_i(A) \right].$$

The parameter γ_i is the local submodularity ratio. When $\gamma_i = 1$, the condition recovers the standard submodular case. A smaller γ_i allows complementarity among PI fragments, while requiring that the joint benefit of a fragment set remains sufficiently reflected by its individual marginal gains.

Assumption 2 (Local approximate monotonicity) *For each fixed intervention point i, there exists $\delta_i \geq 0$ such that*

$$f_i(B) \geq f_i(A) - \delta_i$$

for any $A \subseteq B \subseteq \mathcal{V}$ with $|B| \leq 2K$.

Assumption 2 allows the addition of PI fragments to decrease the local likelihood–divergence objective, but requires the total decrease between any two relevant nested sets to be bounded by δ_i. The standard monotonicity assumption is recovered when $\delta_i = 0$.

Let

$$S_i^\star \in \mathrm{argmax}_{S \subseteq \mathcal{V},\, |S| \leq K} F_i(S)$$

denote the optimal PI subset at intervention point i, and let S_i^{g} denote the subset returned by Algorithm 1.

Theorem 2 (Greedy approximation analysis) *Under Assumptions 1 and 2, the greedy solution satisfies*

$$F_i(S_i^{\mathrm{g}}) - F_i(\varnothing) \geq \left[1 - \left(1 - \frac{\gamma_i}{K} \right)^K \right] \left[F_i(S_i^\star) - F_i(\varnothing) - \delta_i \right].$$

Consequently,

$$F_i(S_i^{\mathrm{g}}) - F_i(\varnothing) \geq \left(1 - e^{-\gamma_i} \right) \left[F_i(S_i^\star) - F_i(\varnothing) - \delta_i \right].$$

Position	Retained Gain	Jaccard	Exact Opt.
S_1	$98.92\% \pm 8.35\%$	0.971 ± 0.133	95.00%
S_2	$95.06\% \pm 18.83\%$	0.941 ± 0.212	92.50%
S_3	$95.57\% \pm 19.13\%$	0.958 ± 0.186	95.00%

Table 2: Empirical approximation quality of greedy PI selection over 200 trajectories. S_1 is the initial prompt position; S_2 and S_3 are rollout positions.

Theorem 2 bounds the fraction of local proxy improvement retained by greedy selection relative to exhaustive subset optimization at the same fixed context. Its scope is separate from Theorem 1: it does not control the answer-likelihood proxy error, the effect of current selections on future segments, or downstream student accuracy.

Empirical approximation quality. We evaluate greedy selection against exhaustive subset search on 200 problems randomly sampled from the OpenThoughts mathematical reasoning training set with seed 42. Using Qwen3-1.7B without a trained adapter, we generate one student rollout per problem and divide each rollout into 3 segments. We use the PI-selection method in Section 4.2, with the same configuration as in the main experiments. We set $K = N = 8$, so all subsets are feasible and the selection budget does not cause early stopping. Let $S_i^\star$ maximize F_i and S_i^{g} denote the greedy solution. We measure the fraction of optimal local objective improvement retained by greedy selection:

$$\rho_i = \frac{F_i(S_i^{\mathrm{g}}) - F_i(\varnothing)}{F_i(S_i^\star) - F_i(\varnothing)}.$$

Retained Gain is computed only for contexts with a nonempty optimal selection and positive optimal gain (65.7% of all contexts). When the empty set is optimal, the denominator is zero and the context is excluded from Retained Gain. We also report Jaccard similarity and the proportion of contexts in which greedy attains the optimal objective (Exact Opt.), both computed over all contexts. The Jaccard similarity of two empty sets is defined as one. For Exact Opt., two empty sets constitute an optimal-objective match and contribute one.

As shown in Table 2, at the initial prompt position S_1, greedy selection retains 98.92% of the optimal local objective improvement on average, achieves a mean Jaccard similarity of 0.971, and matches the optimal objective in 95.00% of cases. At rollout position S_2, mean Retained Gain is 95.06%, mean Jaccard similarity is 0.941, and the Exact Opt. rate is 92.50%. At S_3, the corresponding values are 95.57%, 0.958, and 95.00%, respectively. These results show that greedy selection closely approximates exhaustive subset search for the local proxy at both the initial prompt and rollout positions. They assess local optimization quality, rather than the validity of the proxy or optimality of the complete intervention scheme.

B Experiments

B.1 Detailed Experimental Setup

We train on the mathematical reasoning subset of OpenThoughts for one epoch, with performance typically converging within 75–150 gradient-update steps. The effective batch size is set to 32 for all experiments. Following the OPSD study, which finds only limited performance gains from further increasing the student rollout length, we cap each student rollout at 1,024 tokens. We also follow OPSD in fixing the teacher to the initial model throughout training, so that only the student parameters are updated. All models are optimized with a learning rate of 5×10^{-6} using LoRA. We set the LoRA rank to 64 and the LoRA scaling parameter to 128. LoRA adapters are applied to the attention projection modules `q_proj`, `k_proj`, `v_proj`, and `o_proj`, as well as the feed-forward modules `gate_proj`, `up_proj`, and `down_proj`. We hold out 100 problems from the OpenThoughts dataset as the validation set, none of which are used for training. Checkpoints are saved every 25 gradient-update steps, and the checkpoint achieving the best validation performance is selected for final evaluation. For the λ selection in Section 5.5, we compare candidates using validation Avg@6 at a fixed budget of 50 training steps.

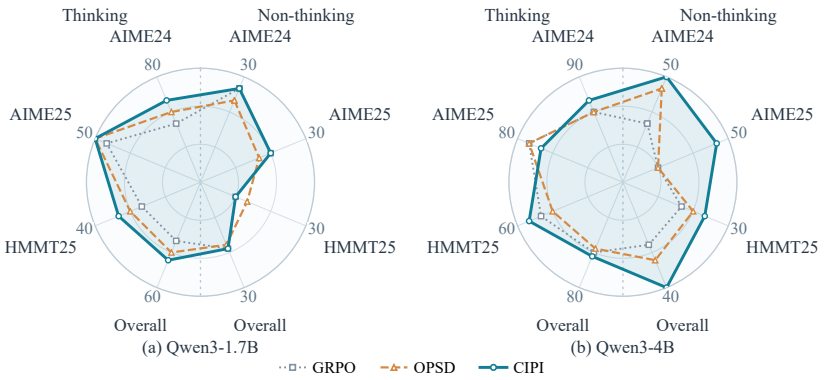

Figure 7: Self-consistency accuracy (SC@12, %) using majority voting over 12 sampled solutions for Qwen3-1.7B (left) and Qwen3-4B (right) in thinking and non-thinking modes. Overall averages the three benchmarks. Axis maxima are labeled; centers are 30 percentage points lower, with 10 points per ring.

B.2 CIPI Implementation Details

Before inserting PI into the teacher context, we remove the sentence containing the final answer marked by `\boxed{}`, so that the inserted PI does not directly reveal the answer.

We partition the PI into 8 approximately equal-length fragments by token count. As with the student rollout segmentation, each partition boundary is adjusted to a nearby sentence ending whenever possible to preserve semantic completeness. We divide each student rollout into 3 segments under the thinking setting and 4 segments under the non-thinking setting. Once a PI fragment is inserted, it remains in the teacher context for all subsequent rollout segments. A PI fragment may be selected again at a later intervention position S_j, but duplicate occurrences of the same fragment are not allowed within a single set S_i. When multiple fragments are selected at the same intervention position, they are arranged according to their original order in the reference solution.

Based on the validation study presented in Section 5.5, we use $\lambda = 0.7$ for both settings. Following OPSD, we use the full-vocabulary forward KL divergence from the teacher distribution to the student distribution to compute the distillation loss.

Before performing PI selection, we apply a trajectory-level reliability gate to determine whether CIPI is needed for the current student rollout. Specifically, we evaluate the teacher's average log-likelihood of the ground-truth answer at intermediate rollout positions and fit a linear trend over these measurements. CIPI is activated only when the estimated log-likelihood slope is below 0.0, indicating that the teacher is being misled by the student rollout. Otherwise, the trajectory falls back to standard OPSD without in-rollout intervention.

B.3 Self-Consistency Performance

Figure 7 shows that CIPI achieves the highest overall SC@12 in all four settings, tying with GRPO for Qwen3-1.7B in non-thinking mode. Relative to OPSD, the gains are 1.1 and 7.8 percentage points for Qwen3-1.7B and Qwen3-4B in non-thinking mode, and 2.2 points for both models in thinking mode. The Pass@12 gains in Figure 3 show that CIPI increases the probability of finding at least one correct solution; the SC@12 gains further suggest that successful exploration is not merely occasional, as correct answers recur across sampled reasoning paths.

B.4 Benchmark-Level Results for the IRI Ablation

Figure 8 provides the per-benchmark scores underlying the overall results in Figure 4(c), using the same intervention strategies described in Section 5.3. All IRI variants (Full-IRI, Prob-IRI, and Seq-IRI), as well as CIPI, use the same trajectory-level reliability gate described in Section 5.4.

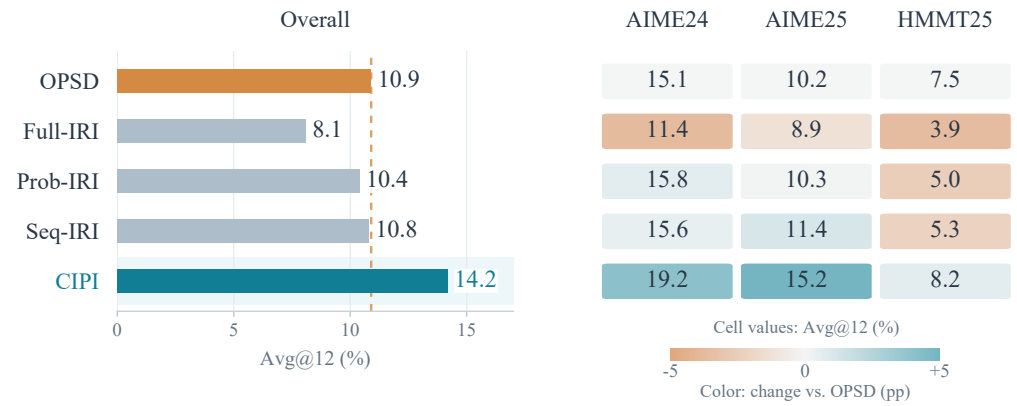

Figure 8: Supplementary results for the IRI ablation in Figure 4(c) on Qwen3-1.7B in non-thinking mode (Avg@12, %). **Left:** Overall scores, repeated for reference; the dashed line marks OPSD. **Right:** Scores on each benchmark, with colors indicating changes relative to OPSD in percentage points.

CIPI outperforms OPSD on all three benchmarks, with gains of 4.1, 5.0, and 0.7 percentage points on AIME 2024, AIME 2025, and HMMT 2025, respectively. All three uncontrolled variants remain below OPSD in overall accuracy. Their benchmark-level results are mixed: PROB-IRI and SEQ-IRI provide small gains on AIME 2024, but all three variants reduce accuracy on HMMT 2025 relative to OPSD.

B.5 ABLATION: TRAJECTORY-LEVEL RELIABILITY GATING

We ablate the trajectory-level reliability gate underlying the apply rates in Figure 5. On Qwen3-1.7B, *w/o gating* applies CIPI's PI-selection procedure to every rollout, retaining the same selection objective and intervention positions, instead of falling back to OPSD when the teacher's correct-answer log-likelihood does not decline.

Removing the gate lowers accuracy on all three benchmarks in both modes (Figure 9). Overall Avg@12 falls from 14.2 to 13.1 in non-thinking mode and from 43.7 to 42.7 in thinking mode, corresponding to drops of 1.1 and 1.0 percentage points, respectively. This is consistent with Theorem 1, which motivates balancing G_{teach} against $L\eta C_{\text{distill}}$. When answer perplexity is already stable or decreasing, additional PI may yield insufficient guidance gain to offset its distillation cost. Gating thus implements our intent to introduce in-rollout intervention only when the teacher needs correction. The declining apply rates in Figure 5 further suggest that the need for such intervention gradually diminishes as the student policy improves.

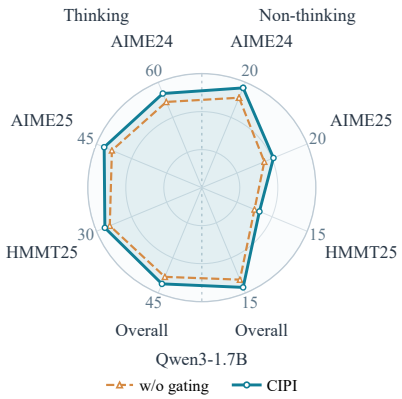

Figure 9: Gating ablation on Qwen3-1.7B (Avg@12, %). Axis maxima are labeled; centers are 15 points lower, with 5 points per ring.

B.6 EXTENSION: ADAPTIVE INTERVENTION TIMING

Beyond deciding whether a rollout requires intervention (Section 5.4), we examine when to intervene within that rollout. Standard CIPI uses uniformly spaced intervention points. Jointly optimizing these positions and the selected PI fragments in Equation 1 would substantially enlarge the

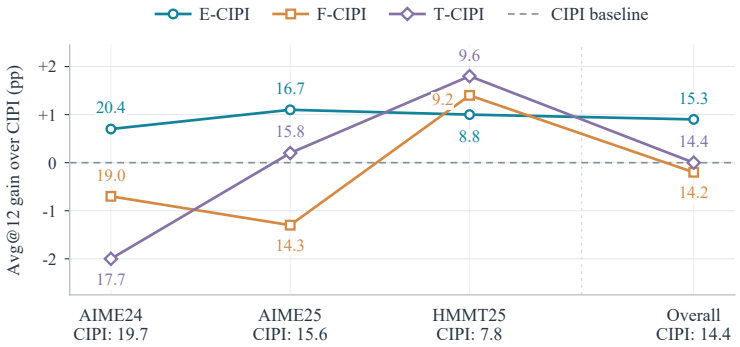

Figure 10: Adaptive intervention timing on Qwen3-1.7B in non-thinking mode. Vertical positions show changes in Avg@12 relative to standard CIPI (percentage points); the dashed zero line denotes uniformly spaced intervention. Point labels report absolute Avg@12 scores (%), with standard CIPI scores listed along the bottom.

combinatorial search space. We therefore consider a simpler extension that uses window-based teacher-reliability signals to trigger intervention while retaining the same PI selection objective.

We evaluate three triggers using Qwen3-1.7B in non-thinking mode. **E**ntropy-CIPI (E-CIPI) intervenes when teacher entropy is high Jin et al. (2026); **F**idelity-CIPI (F-CIPI) uses low teacher confidence on student-generated tokens Liu et al. (2026a); and **T**eachability-CIPI (T-CIPI) uses low teacher–student top-k overlap Liu et al. (2026b).

Figure 10 shows that E-CIPI improves all three benchmarks, raising overall Avg@12 from 14.4 to 15.3. F-CIPI and T-CIPI achieve larger gains on HMMT 2025 (+1.4 and +1.8 percentage points, respectively), but weaker AIME performance leaves their overall scores at 14.2 and 14.4.

Since adaptive triggers introduce additional tuning choices and signal-monitoring overhead, we retain uniformly spaced intervention as the default. The E-CIPI results nevertheless show that adaptive timing can complement CIPI's PI selection, allowing improved teacher-reliability signals to be incorporated without changing the selection objective.

B.7 Amount of Selected Privileged Information

Table 3: Mean PI-selection ratios over the first 100 training steps (%). The final column is the total insertion ratio.

Model	Mode	Prompt ratio	Rollout ratio	Sum
Qwen3-1.7B	Non-thinking	18.21	35.11	53.32
Qwen3-4B	Non-thinking	20.31	35.03	55.34
Qwen3-1.7B	Thinking	12.08	10.26	22.34
Qwen3-4B	Thinking	12.18	9.42	21.60

Table 3 summarizes the mean prompt and rollout PI-selection ratios over the first 100 training steps, together with the total insertion ratio. In thinking mode, the prompt ratio averages about 12%, while the rollout ratio is about 9%–10%; in non-thinking mode, the corresponding averages are about 18%–20% and 35%. The total insertion ratios remain well below the full-PI reference level of 100%. For comparison, OPSD uses one full PI in its initial prompt, while Prob-IRI (in expectation) and Seq-IRI in Section 5.3 also insert one full PI across the intervention positions. Together with Table 1, these statistics suggest that CIPI's best overall performance is driven by controlled PI selection rather than simply inserting more PI.

C PROOFS

C.1 PROOF OF THEOREM 1

Proof 1 *Throughout the proof, the normalized prefix distribution $\mu = d^{p_\theta}$, the reference supervision r, and the teacher distributions are fixed at the current student parameters. Write*

$$s_\theta(h,a) = \nabla_\theta \log p_\theta(a \mid h), \qquad u = \nabla J_{\mathrm{sur}}(\theta).$$

By the definitions of the surrogate and the forward-KL distillation loss,

$$u = \mathbb{E}_{h\sim\mu} \sum_a r(a \mid h) s_\theta(h,a),$$
$$g_S = -\mathbb{E}_{h\sim\mu} \sum_a q^S(a \mid h) s_\theta(h,a).$$

For any two conditional distributions v and w with finite expected KL divergence, the triangle inequality, Jensen's inequality, and $\|s_\theta(h,a)\|_2 \le B$ imply

$$\begin{aligned}\left\| \mathbb{E}_{h\sim\mu} \sum_a \big(v(a \mid h) - w(a \mid h)\big) s_\theta(h,a) \right\|_2^2 &\le B^2 \mathbb{E}_{h\sim\mu} \left[\|v(\cdot \mid h) - w(\cdot \mid h)\|_1^2 \right] \\ &\le 2B^2 \mathbb{E}_{h\sim\mu} D_{\mathrm{KL}}(v(\cdot \mid h) \,\|\, w(\cdot \mid h)),\end{aligned}$$

where the last step uses Pinsker's inequality. Applying this bound with $v = r$ and $w = q^S$ gives

$$\|g_S + u\|_2^2 = \left\| \mathbb{E}_{h\sim\mu} \sum_a \big(r(a \mid h) - q^S(a \mid h)\big) s_\theta(h,a) \right\|_2^2 \le 2B^2 E_{\mathrm{teach}}(S).$$

Thus, the teacher's supervision error controls the deviation of the distillation update direction from the surrogate ascent direction.

We also bound the magnitude of the distillation gradient. For every prefix h, the student score has zero expectation under its own distribution:

$$\sum_a p_\theta(a \mid h) s_\theta(h,a) = \nabla_\theta \sum_a p_\theta(a \mid h) = 0.$$

Consequently,

$$g_S = -\mathbb{E}_{h\sim\mu} \sum_a \big(q^S(a \mid h) - p_\theta(a \mid h)\big) s_\theta(h,a).$$

Applying the same distributional bound with $v = q^S$ and $w = p_\theta$ yields

$$\|g_S\|_2^2 \le 2B^2 C_{\mathrm{distill}}(S).$$

The first-order improvement term satisfies

$$-\langle u, g_S \rangle = \frac{1}{2}\|u\|_2^2 + \frac{1}{2}\|g_S\|_2^2 - \frac{1}{2}\|u + g_S\|_2^2 \ge \frac{1}{2}\|u\|_2^2 - B^2 E_{\mathrm{teach}}(S).$$

By the local smoothness of J_{sur} along the update path and $\theta_S^+ - \theta = -\eta g_S$, we obtain

$$\begin{aligned}J_{\mathrm{sur}}(\theta_S^+) - J_{\mathrm{sur}}(\theta) &\ge -\eta \langle u, g_S \rangle - \frac{L\eta^2}{2}\|g_S\|_2^2 \\ &\ge \frac{\eta}{2}\|u\|_2^2 - \eta B^2 E_{\mathrm{teach}}(S) - L\eta^2 B^2 C_{\mathrm{distill}}(S).\end{aligned}$$

Since both teacher quality and supervision error are evaluated against the same fixed r and μ,

$$E_{\mathrm{teach}}(0) - E_{\mathrm{teach}}(S) = \mathbb{E}_{h\sim\mu} \sum_a r(a \mid h) \log \frac{q^S(a \mid h)}{q^0(a \mid h)} = G_{\mathrm{teach}}(S).$$

Substituting $E_{\text{teach}}(S) = E_{\text{teach}}(0) - G_{\text{teach}}(S)$ into the preceding bound gives

$$J_{\text{sur}}(\theta_S^+) - J_{\text{sur}}(\theta) \geq \eta b_\theta + \eta B^2 G_{\text{teach}}(S) - L\eta^2 B^2 C_{\text{distill}}(S),$$
$$b_\theta = \frac{1}{2}\|u\|_2^2 - B^2 E_{\text{teach}}(0).$$

This proves the theorem. Since b_θ is independent of S, maximizing this lower bound over intervention schemes is equivalent to maximizing $G_{\text{teach}}(S) - L\eta C_{\text{distill}}(S)$, which establishes Corollary 1.

On the trade-off coefficient. Retaining the nonnegative term $\frac{1}{2}\|g_S\|_2^2$ in the first-order improvement bound, rather than dropping it as above, gives

$$J_{\text{sur}}(\theta_S^+) - J_{\text{sur}}(\theta) \geq \eta b_\theta + \eta B^2 G_{\text{teach}}(S) - \frac{\eta}{2}(L\eta - 1)\|g_S\|_2^2.$$

To obtain a tighter KL-based lower bound by substituting $\|g_S\|_2^2 \leq 2B^2 C_{\text{distill}}(S)$ into this expression, we require $L\eta \geq 1$, so that the resulting coefficient $\lambda = L\eta - 1$ is nonnegative. When $L\eta < 1$, this substitution does not preserve the lower bound and therefore does not justify a negative KL weight. Alternatively, dropping the nonnegative first-order contribution as in the proof gives the looser coefficient $\lambda = L\eta \geq 0$, without requiring $L\eta \geq 1$. Thus, both KL-based bounds support a nonnegative discrepancy penalty. We use the latter relaxation to keep the theorem concise and avoid an additional condition on the unknown local smoothness constant L. This changes the theoretical coefficient while preserving the objective's form, $G_{\text{teach}} - \lambda C_{\text{distill}}$.

Empirically, Figure 6 further supports using a strictly positive penalty: reducing λ below the best-performing value of 0.7 sharply degrades validation accuracy, with Avg@6 falling from 58.0% to 55.1% at $\lambda = 0.1$ and to 53.4% at $\lambda = 0$, below the base model's 53.8%. We therefore tune λ over nonnegative candidates and select a positive value based on validation performance. This choice is also consistent with standard knowledge distillation, which minimizes teacher–student KL divergence (or equivalently, cross-entropy for a fixed teacher) as a discrepancy penalty Hinton et al. (2015); Kim and Rush (2016).

C.2 A Sufficient Condition for the Answer-Likelihood Proxy

Fix a problem, a student rollout segment, and a normalized distribution μ over its token prefixes. Suppress the common problem conditioning. Consider two fixed autoregressive distributions over generated sequences, P_{ref} and P_{deg}, representing a reference reasoning mode and a degraded reasoning mode. Suppose the effective teacher distribution under intervention S is

$$P_{\pi_S} = \pi_S P_{\text{ref}} + (1 - \pi_S) P_{\text{deg}}, \qquad 0 < \pi_S < 1. \tag{4}$$

Thus, the candidate PI changes the mixing weight while the two component distributions remain fixed. Its effect is represented by π_S; the student prefixes used for comparison do not include candidate-specific PI. Write $P_k(h)$ for the probability of prefix h under component k, and set

$$q^S(v \mid h) = P_{\pi_S}(v \mid h), \qquad r(v \mid h) = P_{\text{ref}}(v \mid h), \qquad b(v \mid h) = P_{\text{deg}}(v \mid h).$$

Let h_A denote the fixed answer-scoring prefix, including the student segment and the answer prompt, and let $a^* = (a_1^*, \ldots, a_m^*)$ be the correct answer. The score used in Section 4.2 is

$$\ell(S) = \frac{1}{m} \log P_{\pi_S}(a^* \mid h_A) = \frac{1}{m} \sum_{j=1}^{m} \log P_{\pi_S}(a_j^* \mid h_A, a_{<j}^*).$$

Proposition 1 (Answer likelihood and reference supervision) *Suppose both components assign positive probability to h_A and to μ-almost every rollout prefix, and the expected supervision KL divergences are finite. Assume that the reference mode assigns the correct answer a higher conditional probability:*

$$P_{\text{ref}}(a^* \mid h_A) > P_{\text{deg}}(a^* \mid h_A).$$

Under Eq. (4), for any two interventions S_1, S_2 evaluated on the same fixed prefixes,

$$\ell(S_2) > \ell(S_1) \quad \Longrightarrow \quad E_{\text{teach}}(S_2) \leq E_{\text{teach}}(S_1),$$

where $E_{\text{teach}}(S) = \mathbb{E}_{h\sim\mu} D_{\text{KL}}(r(\cdot \mid h) \| q^S(\cdot \mid h))$. The inequality is strict if $r(\cdot \mid h) \neq b(\cdot \mid h)$ on a set of positive μ-measure. In particular, if the baseline teacher q^0 belongs to the same family, $\ell(S) > \ell(0)$ implies $G_{\text{teach}}(S) \geq 0$.

Proof 2 *Conditioning the sequence mixture on any fixed prefix h gives*

$$q^S(\cdot \mid h) = w_h(\pi_S) r(\cdot \mid h) + [1 - w_h(\pi_S)] b(\cdot \mid h),$$
$$w_h(\pi) = \frac{\pi P_{\mathrm{ref}}(h)}{\pi P_{\mathrm{ref}}(h) + (1-\pi) P_{\mathrm{deg}}(h)}. \tag{5}$$

Positive prefix probabilities imply that $w_h(\pi)$ is strictly increasing in π. Applying the same conditioning to the entire answer sequence,

$$P_\pi(a^* \mid h_A) = w_{h_A}(\pi) P_{\mathrm{ref}}(a^* \mid h_A) + [1 - w_{h_A}(\pi)] P_{\mathrm{deg}}(a^* \mid h_A).$$

The assumed ordering of the answer probabilities makes ℓ strictly increasing in π. Hence $\ell(S_2) > \ell(S_1)$ implies $\pi_{S_2} > \pi_{S_1}$.

For each relevant rollout prefix, let $w_k = w_h(\pi_{S_k})$. Since $0 < w_1 < w_2 < 1$, Eq. (5) gives

$$q^{S_2}(\cdot \mid h) = \alpha_h r(\cdot \mid h) + (1 - \alpha_h) q^{S_1}(\cdot \mid h), \qquad \alpha_h = \frac{w_2 - w_1}{1 - w_1} \in (0, 1).$$

By convexity of KL divergence in its second argument,

$$\begin{aligned} D_{\mathrm{KL}}(r \| q^{S_2}) &\le \alpha_h D_{\mathrm{KL}}(r \| r) + (1 - \alpha_h) D_{\mathrm{KL}}(r \| q^{S_1}) \\ &= (1 - \alpha_h) D_{\mathrm{KL}}(r \| q^{S_1}) \le D_{\mathrm{KL}}(r \| q^{S_1}), \end{aligned}$$

where all distributions are conditioned on the same h. If $r(\cdot \mid h) \neq b(\cdot \mid h)$, then $D_{\mathrm{KL}}(r \| q^{S_1}) > 0$, so the final comparison is strict. Averaging over the unchanged μ proves the result.

The proposition describes a possible mechanism: PI increases the weight of a reference reasoning mode, improving both answer prediction and supervision on the student's existing prefixes. The posterior weight w_h may differ across prefixes, so the argument also accommodates multi-token answers. The fixed-component assumption is a sufficient condition for this explanation, not an assumption of Theorem 1 or a claim that every PI intervention follows this model. Accordingly, CIPI uses answer likelihood as a practical proxy for guidance gain. This local ordering result does not equate their magnitudes or imply identical optimizers after adding the KL penalty; downstream accuracy remains an empirical question.

C.3 Proof of Theorem 2

Proof 3 *Let $S_i^{(t)}$ denote the fragment set after t greedy iterations, where $S_i^{(0)} = \varnothing$, and define the remaining optimality gap as*

$$R_t = f_i(S_i^\star) - f_i(S_i^{(t)}).$$

We first consider an iteration at which the greedy procedure selects another fragment. By weak submodularity,

$$\sum_{j \in S_i^\star \setminus S_i^{(t)}} \Delta_i(j \mid S_i^{(t)}) \ge \gamma_i \left[f_i(S_i^{(t)} \cup S_i^\star) - f_i(S_i^{(t)}) \right].$$

Since both $S_i^{(t)}$ and $S_i^\star$ contain at most K fragments, their union contains at most $2K$ fragments. Approximate monotonicity therefore gives

$$f_i(S_i^{(t)} \cup S_i^\star) \ge f_i(S_i^\star) - \delta_i.$$

Combining the two inequalities yields

$$\sum_{j \in S_i^\star \setminus S_i^{(t)}} \Delta_i(j \mid S_i^{(t)}) \ge \gamma_i (R_t - \delta_i).$$

If $R_t > \delta_i$, the right-hand side is positive. Since $|S_i^\star \setminus S_i^{(t)}| \le K$, at least one remaining fragment satisfies

$$\max_{j \notin S_i^{(t)}} \Delta_i(j \mid S_i^{(t)}) \ge \frac{\gamma_i}{K} (R_t - \delta_i).$$

If $R_t \le \delta_i$, the right-hand side is non-positive, while the greedy procedure only performs another iteration when its selected marginal gain is positive. Thus, the same inequality also holds in this case.

Because the greedy procedure selects the fragment with the largest positive marginal gain,

$$f_i(S_i^{(t+1)}) - f_i(S_i^{(t)}) \ge \frac{\gamma_i}{K}(R_t - \delta_i).$$

It follows that

$$\begin{aligned} R_{t+1} &= R_t - \left[f_i(S_i^{(t+1)}) - f_i(S_i^{(t)})\right] \\ &\le \left(1 - \frac{\gamma_i}{K}\right) R_t + \frac{\gamma_i}{K}\delta_i. \end{aligned}$$

Let

$$a_i = 1 - \frac{\gamma_i}{K}.$$

Applying the recurrence for K iterations gives

$$R_K \le a_i^K R_0 + (1 - a_i^K)\delta_i.$$

Since $f_i(\varnothing) = 0$, we have $R_0 = f_i(S_i^\star)$. Therefore,

$$f_i(S_i^{(K)}) \ge (1 - a_i^K)\left[f_i(S_i^\star) - \delta_i\right].$$

Substituting $f_i(S) = F_i(S) - F_i(\varnothing)$ proves the first inequality. The second follows from

$$\left(1 - \frac{\gamma_i}{K}\right)^K \le e^{-\gamma_i}.$$

D Prompt and Formatting Details

CIPI uses different formatting for PI fragments selected for the initial teacher prompt and those inserted within the student rollout. In both cases, the selected fragments are sorted according to their original indices in the reference solution before being placed in the teacher context. The resulting user message is then processed using the model's standard chat template, with the teacher's configured thinking mode enabled or disabled accordingly.

Initial teacher prompt. At the first intervention position, the selected PI fragments are included in the teacher's initial prompt as a partial reference solution. Given a problem x and the selected set S_1, the teacher user message follows the template below:

```
Problem: {problem}
Here is a reference solution to this problem:
=== Reference Solution Begin ===
Reference Solution Fragments:
=== Reference Solution Fragment {index} Begin ===
{fragment content}
=== Reference Solution Fragment {index} End ===

  ...
=== Reference Solution End ===
{transition prompt}
Please reason step by step, and put your final answer
within \boxed{}.
```

Only fragments selected into S_1 appear between the reference-solution delimiters. When multiple fragments are selected, each fragment is enclosed by its own begin and end markers and arranged according to its position in the original reference solution.

Table 4: Training efficiency of OPSD and CIPI under the same experimental setting.

Method	Samples / second	Steps / second
OPSD	1.203	0.038
CIPI	0.913	0.029

In-rollout PI intervention. For an intervention position $i > 1$, the selected fragments are inserted immediately before the current student rollout segment $\hat{y}_i$. The inserted block follows the template:

```
[Teacher-only reference reminder]
The following reference solution fragments are
privileged information for the teacher.  Use them
only to stay calibrated while evaluating the student's
current reasoning prefix.
=== Reference Solution Fragment {index} Begin ===
{fragment content}
=== Reference Solution Fragment {index} End ===

  ...
Continue from the student's current prefix.
```

The reminder explicitly identifies the inserted material as teacher-only privileged information and instructs the teacher to use it for calibrating its evaluation of the current student prefix rather than treating it as student-generated reasoning. All fragments selected at the same intervention position are placed within a single reminder block and ordered by their original reference indices.

We also experimented with a more natural in-rollout insertion format, for example, prepending the selected PI fragments with "Let me work through this part more carefully." However, this design produced a noticeably larger teacher–student KL divergence. We believe that such reasoning-style continuation templates cause the inserted PI to be interpreted as part of the ongoing reasoning trajectory, thereby inducing a stronger shift in the teacher's reasoning state when the reference fragments are not fully aligned with the current student prefix. We therefore adopt the structured teacher-only reference template described above, which explicitly separates the inserted PI from the student-generated reasoning and yields a more controlled policy shift.

E Computational Cost and Efficiency

CIPI introduces additional computation for evaluating candidate PI fragments at each intervention position. To reduce this overhead, candidate teacher contexts are grouped into batches, and their answer likelihoods and teacher distributions are evaluated in parallel rather than through separate sequential forward passes. The throughput is measured using Qwen3-1.7B under the thinking setting on two NVIDIA H20 GPUs. We use an effective batch size of 32, a candidate batch size of 8, and 3 intervention positions per rollout. The reported numbers reflect end-to-end training throughput, including student rollout generation, online PI selection, teacher forward passes, and parameter optimization.

Table 4 compares the training throughput of CIPI and OPSD under the same experimental setting. CIPI processes 0.913 samples per second and 0.029 optimization steps per second, compared with 1.203 samples per second and 0.038 steps per second for OPSD. This corresponds to an approximately 24.1% reduction in throughput, or equivalently an approximately 31.8% increase in training time.

Performance under comparable training budgets. To examine whether CIPI's gains can be explained by its additional training time, we evaluate Qwen3-1.7B in thinking mode at checkpoints from 25 to 150 optimization steps (Figure 11). Under the same step budget, OPSD reaches its highest observed overall Avg@12 of 41.8% at step 75; extending training to 150 steps does not improve this

score, which instead decreases to 40.5%. CIPI continues to improve over the evaluated checkpoints, reaching 43.8% at step 150. Thus, in this experiment, OPSD saturates within the first 100 steps and does not benefit from the additional steps evaluated here.

For a time-based comparison, we estimate relative wall-clock budgets using the measured 31.8% training-time overhead, assuming a constant average time per step for each method. With the time required for one OPSD step as the unit, checkpoint s is placed at s for OPSD and $1.318s$ for CIPI. CIPI at step 75 therefore lies at 98.85, slightly before OPSD at step 100, and achieves 41.7% versus 40.9%. Similarly, CIPI at step 100 reaches 42.0% at a relative time of 131.8, exceeding OPSD's highest observed score of 41.8% even when OPSD is allowed to train through step 150. These comparisons suggest that additional training time alone does not explain CIPI's gains over OPSD in this setting.

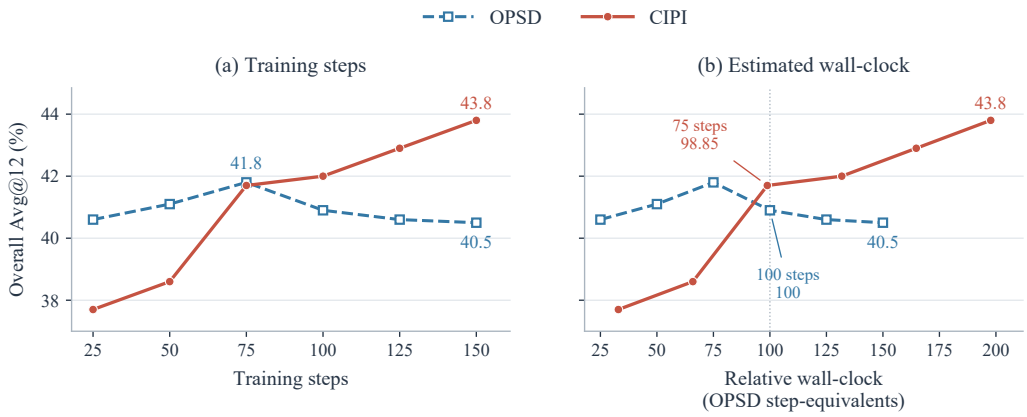

Figure 11: Performance under different training budgets for Qwen3-1.7B in thinking mode. Overall Avg@12 is the macro-average across the three benchmarks. (a) Performance at the same training checkpoints. (b) The same scores plotted against estimated relative wall-clock time, in OPSD step-equivalents: $t_{\text{OPSD}}(s) = s$ and $t_{\text{CIPI}}(s) = 1.318s$, using the measured 31.8% training-time overhead. Markers denote evaluated checkpoints and lines connect adjacent observations. The exact scaled positions are retained, including 98.85 for CIPI at step 75 and 100 for OPSD at step 100.