CoTrek: Toward Scalable On-Policy Distillation for Long Chain-of-Thought Reasoning

Heng Zhang[1], Chengyu Zhou[2], Jiajun Wu[3], Xinyuan Liu[4]
Liheng Zhang[1], Yueqi Guo[1], Rui Liu[1], JiaHao Hong[1]
Xuanxun Lian[1], Jinpeng Lu[1], Jin Huang[1]*
[1]South China Normal University [2]LexisNexis
[3]Central South University [4]Northwestern University
2024025450@m.scnu.edu.cn, czhou24190@gmail.com
8210231904@csu.edu.cn, xinyuanliu2024@U.NORTHWESTERN.EDU
huangjin@m.scnu.edu.cn

Abstract

On-policy distillation has emerged as an efficient paradigm for improving long chain-of-thought reasoning in large language models, where the student learns from its own rollouts while receiving dense teacher feedback on the states it actually visits. Prior work advances *continuation learning* through better teacher scoring, more stable optimization, and more learnable traces. Long-horizon OPD exhibits a recovery challenge when student reasoning drifts at deep prefixes, requiring targeted supervision for path re-entry. To address this challenge, we propose CoTrek, a recovery-centric framework for on-policy distillation in long chain-of-thought reasoning. CoTrek operates on student-generated deep prefixes that have drifted off track and admit successful teacher recovery. From each such prefix, it samples multiple teacher recoveries, extracts the short *repair segment* they share, and trains the student to produce this repair before continuing the remaining reasoning on policy. This design uses focused path re-entry as the distillation target. To further support long-horizon learning, CoTrek follows a depth curriculum that expands training depth according to the observed recovery gap, allowing the student to progressively extend its effective reasoning horizon. Extensive experiments on long chain-of-thought benchmarks demonstrate that CoTrek delivers (I) **substantial performance gains**, improving Vanilla OPD by up to 6.84 percentage points in final accuracy and 9.27 points in recovery success; and (II) **strong long-horizon robustness**, preserving gains on deep prefixes, with accuracy improvements over Vanilla OPD growing from 2.1 to 8.9 percentage points across increasing generation-length regimes.

1 Introduction

Large language models have advanced rapidly on reasoning tasks, with long chain-of-thought generation emerging as a primary driver of these gains [1, 2, 3, 4, 5, 6]. A key enabler is on-policy distillation: the student generates its own reasoning rollouts, receives dense teacher feedback on the states it actually visits, and improves at a cost far below reinforcement learning [7, 8, 9]. This paradigm has spawned a growing family of methods that differ in how the teacher signal is constructed and consumed [10, 11, 12, 13, 14, 15].

Early approaches train the student to imitate fixed teacher traces [16, 17]. Subsequent work shifts the locus of supervision to student-generated prefixes, letting the teacher score or complete states

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

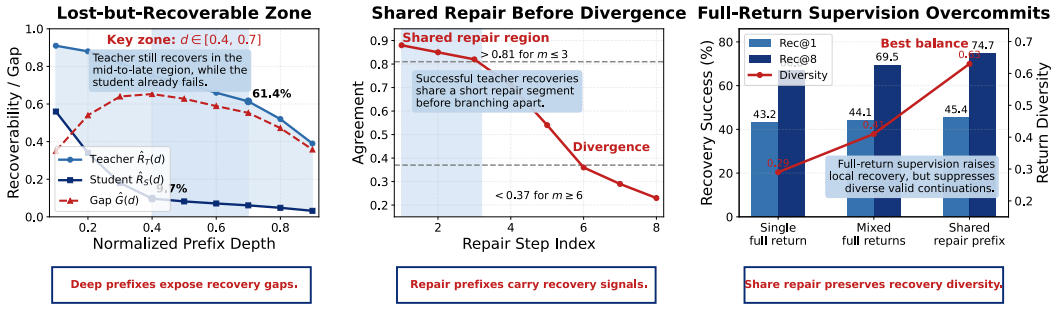


Figure 1: Empirical motivation for **CoTrek**. **(a)** A broad lost-but-recoverable zone emerges at mid-to-late reasoning depths: teacher recoverability remains high while student recoverability drops sharply, yielding a substantial deep prefix recovery gap. **(b)** Successful teacher recoveries from the same deviated prefix consistently share a short repair segment before diverging into distinct valid continuations. **(c)** Distilling full recovery trajectories improves local recovery but suppresses downstream diversity, whereas supervising only the shared repair prefix achieves the best balance between recovery success and return diversity. These findings motivate targeted recovery supervision on deep, teacher-recoverable prefixes.

the student actually reaches [7, 8]. More recent efforts refine signal quality, trajectory selection, and optimization stability [18, 19, 20, 21]. Amid this diversity, a natural question arises: *what kind of distillation behavior most effectively supports long-horizon chain-of-thought reasoning?*

On-policy distillation improves trajectory continuation through teacher scoring, trace selection, and optimization [18, 19, 20, 21]. A small reasoning error can propagate through subsequent steps and keep the student on an ineffective path. The situation mirrors human problem solving. A weaker student memorizes one corrected solution end to end. A stronger student first isolates a short local repair that returns the derivation to solid ground, then continues from there. These observations motivate explicit recovery learning on student-generated deviated prefixes.

Our empirical study makes this bottleneck concrete. Across multiple benchmarks, we observe a substantial recovery gap at intermediate and deep prefixes: strong teachers frequently reach the correct answer from deviated student states, while students exhibit much lower recovery success. We term this asymmetry the **deep prefix recovery gap**. Two further regularities sharpen the picture. First, successful teacher recoveries from the same deviated prefix tend to share a short repair segment before diverging into distinct valid continuations. Second, supervising the student on full recovery trajectories ties it too tightly to a single branch and erodes downstream diversity. Long-CoT distillation should integrate continuation and recovery learning, with supervision concentrated on shared repair segments.

How can on-policy distillation teach students to recover from deep, recoverable reasoning deviations?

To address this challenge, we introduce **CoTrek**, a recovery-centric framework for on-policy distillation in long chain-of-thought reasoning. **CoTrek** starts from student-generated deep prefixes that contain a deviation and admit successful teacher recovery. From each such prefix, the framework samples multiple teacher recoveries, extracts the short repair segment they share, and trains the student to produce this repair from its own deviated state. The remaining reasoning proceeds on policy. To extend this ability to longer horizons, **CoTrek** follows a recovery-guided horizon expansion strategy: the student first repairs shallow deviations and advances to deeper ones only after recovery at the current depth becomes reliable. We evaluate long-horizon scaling through recovery performance across increasing prefix depths and generation lengths.

Our contributions are as follows:

❶ **Empirical Study:** We identify the deep prefix recovery gap in long chain-of-thought on-policy distillation, and reveal two regularities of successful recovery: a shared repair segment before later divergence, and the tendency of full return supervision to overcommit the student to one branch.

❷ **Recovery-Centric Framework:** We propose **CoTrek**, a framework that mines recoverable deep prefixes, extracts shared repair segments from successful teacher recoveries, and trains the student to return to an effective reasoning path through recovery-guided horizon expansion.

❸ **Experimental Evaluation:** Extensive experiments on long chain-of-thought benchmarks demonstrate that `CoTrek` delivers (I) **substantial performance gains**, improving Vanilla OPD by up to 6.84 percentage points in final accuracy and 9.27 points in recovery success; and (II) **strong long-horizon robustness**, preserving gains on deep prefixes, with accuracy improvements over Vanilla OPD growing from 2.1 to 8.9 percentage points across increasing generation-length regimes.

2 Related Work

Long-Chain-of-Thought Distillation. The success of OpenAI o1 [1] has sparked broad interest in distilling long chain-of-thought (long-CoT) reasoning from strong teachers into smaller students [2, 3, 4, 5, 6]. Early efforts adopt off-policy supervised fine-tuning on complete teacher traces. Recent studies examine how their usefulness varies across depth: P-ALIGN [18] shows that later reasoning steps often introduce redundancy and truncates traces to minimal learnable prefixes; Mole-Syn [22] argues that effective transfer hinges on preserving stable reasoning structures rather than surface-form imitation; and a learnability gap has been documented where long-CoT traces from strong teachers consistently hurt small students [23]. These studies investigate teacher-trace selection and structure. `CoTrek` investigates recovery from student-generated deviated prefixes.

On-Policy Distillation. On-policy distillation (OPD) addresses the train-inference distribution mismatch of off-policy methods [16, 17], a form of exposure bias [24] that accumulates errors over long generations. MiniLLM [7] formalized OPD for LLMs under reverse KL, and GKD [8] unified on-/off-policy interpolation across multiple divergences. Subsequent work has extended OPD along several axes, including entropy-aware stabilization [10], relaxed imitation [11], context distillation [12], and long-context self-distillation [13], with growing adoption in industry pipelines [14, 9, 15]. Meanwhile, diagnostic studies have examined failure modes such as repetition and length inflation [25], sampled-token failures [26], dual-path signal calibration [19], and long-horizon performance ceilings [20]. These efforts improve signal quality and optimization stability during student rollouts. `CoTrek` investigates recovery supervision on deviated student prefixes.

Learning from Prefixes. Recent studies characterize variation in supervision across student prefixes. OPD training signals have been shown to concentrate in output prefixes, motivating prefix-only supervision for efficiency [27]. SCOPE [19] calibrates supervision differently for correct versus incorrect rollouts, and teacher reward quality has been found to degrade with trajectory depth [20]. GSPO [21] further replaces token-level importance ratios with sequence-level optimization for long-horizon stability. These findings reveal that prefix quality and long-horizon supervision are highly non-uniform. We study lost-but-recoverable deep prefixes and extract shared repair cores from their successful teacher recoveries.

3 Preliminary

3.1 Long-CoT On-Policy Distillation

Let x denote an input problem with gold answer $a^\star$. We consider a student policy π_θ and a teacher policy π_T. Given x, the student autoregressively generates a long chain-of-thought response $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim \pi_\theta(\cdot \mid x)$, and we write $\hat{y}_{\le t}$ for the prefix up to step t.

In on-policy distillation (OPD), supervision is computed on prefixes visited by the current student. For each position t of a student rollout $\hat{y}$, both models produce next-token distributions over vocabulary $\mathcal{V}$:

$$p_t(v) \triangleq \pi_\theta(v \mid x, \hat{y}_{<t}), \qquad q_t(v) \triangleq \pi_T(v \mid x, \hat{y}_{<t}). \tag{1}$$

A generic OPD objective minimizes the discrepancy between these distributions along student-generated trajectories:

$$\mathcal{L}_{\text{base}}(\theta) = \mathbb{E}_{x,\ \hat{y}\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} \ell(p_t, q_t) \right], \tag{2}$$

where $\ell(\cdot,\cdot)$ can be instantiated as sampled-token reverse KL, full-vocabulary reverse KL, or a subset-based variant (see Section C for our concrete choice). Standard OPD thus improves how the student continues from prefixes it already visits. Our focus is recovery from student prefixes that have drifted off track and admit successful teacher continuations.

3.2 Recoverability of Deviated Prefixes

We study student-generated prefixes that arise from failed long-CoT rollouts. Let $\hat{y} \sim \pi_\theta(\cdot \mid x)$ be a rollout whose final answer is incorrect. A prefix $p = \hat{y}_{\le t}$ from such a rollout is called a deviated prefix, indicating that it already lies on a reasoning path that does not lead to $a^\star$ under the current student trajectory.

A deviated prefix may nonetheless be solvable by a stronger model. We formalize this intuition through teacher recoverability and student recoverability:

$$R_T(p) \triangleq \mathbb{P}_{y \sim \pi_T(\cdot|x,p)}[\mathrm{Ans}(y) = a^\star], \qquad R_S(p) \triangleq \mathbb{P}_{y \sim \pi_\theta(\cdot|x,p)}[\mathrm{Ans}(y) = a^\star]. \tag{3}$$

A deviated prefix is recoverable if $R_T(p)$ is sufficiently high: the teacher can still reach the correct answer despite the student's prior deviation. In practice, both quantities are estimated via Monte Carlo rollouts (details in Section A.1).

The deep-prefix recovery gap at prefix p is defined as

$$G(p) \triangleq R_T(p) - R_S(p). \tag{4}$$

To characterize this effect across reasoning depth, let $\mathcal{P}_d$ denote the set of deviated prefixes at depth bucket d (see Section A.2 for bucket construction). The depth-wise recovery gap is

$$G(d) \triangleq \mathbb{E}_{p \in \mathcal{P}_d}[R_T(p) - R_S(p)]. \tag{5}$$

Our empirical study (Section 4) reveals three regularities. First, $G(d)$ remains large across a broad region of deep prefixes, meaning many failed student prefixes are still recoverable for the teacher. Second, successful teacher recoveries from the same prefix often share a short repair segment before later continuations diverge. Third, supervising a full teacher return tends to overcommit the student to one branch. These observations motivate a recovery-centric distillation framework that supervises shared repair behavior.

4 Empirical Analysis

We apply the definitions from Section 3 to analyze recovery behavior in long-CoT distillation. Unless otherwise specified, all findings are based on QWEN3-4B-BASE-GRPO $\rightarrow$ QWEN3-1.7B-BASE on AIME 2024, AMC 2023, MATH-500, AND OLYMPIADBENCH. We analyze a partially trained student obtained after 180 OPD steps, construct deviated prefixes by truncating incorrect rollouts at normalized depths $d \in \{0.1, 0.2, \ldots, 0.9\}$, and estimate teacher and student recoverability with $K_T = K_S = 8$ rollouts per prefix. The main results are shown in Figure 1; full details are deferred to Section A.1.

4.1 Finding 1: The Lost-but-Recoverable Zone

Finding 1

Many deviated student prefixes remain recoverable by a stronger teacher. Across a broad mid-to-late depth region ($d \in [0.4, 0.7]$), the teacher still recovers the correct answer in the majority of cases, whereas the student almost never does, revealing a systematic and exploitable recovery gap.

Figure 1(a) plots $\hat{R}_T(d)$, $\hat{R}_S(d)$, and $\hat{G}(d)$ across normalized prefix depth. Two patterns stand out. First, teacher recoverability decreases only gradually with depth. Even at $d = 0.7$, the teacher still recovers the correct answer in 61.4% of cases. Second, student recoverability drops much faster and falls below 9.7% by $d = 0.4$. As a result, $\hat{G}(d)$ peaks in the mid-to-late region and remains substantial even at $d = 0.8$. This reveals a broad lost-but-recoverable zone. In this regime, the student has effectively failed, yet the teacher can still return to the correct answer. This recovery gap motivates targeted supervision on deviated student prefixes.

Table 1: Effect of supervision target on recovery and diversity from recoverable prefixes. Shared repair prefix supervision achieves the best balance of recovery success and downstream diversity.

Supervision Target	Rec@1	Rec@8	Diversity
Single full return	43.2	66.8	0.29
Mixed full returns	44.1	69.5	0.41
Shared repair prefix	45.4	74.7	0.63

4.2 Finding 2: Shared Repair Before Divergence

Finding 2

Successful teacher recoveries from the same deviated prefix share a common early structure. They consistently share a short initial repair segment, with high agreement for the first $\sim$3 reasoning steps, before diverging into diverse valid continuations.

We next examine common structure among successful teacher recoveries. We select prefixes with $\hat{G}(p) \geq \gamma$ and sample $K = 10$ successful teacher continuations from each prefix. We then segment each continuation into reasoning steps and compute cross-recovery agreement $\mathrm{Agree}_m(\mathcal{C}(p))$ at each step index m.

Figure 1(b) shows a clear pattern. Agreement is high for the first few steps, with $\mathrm{Agree}_m > 0.81$ for $m \leq 3$, and then drops sharply, with $\mathrm{Agree}_m < 0.37$ for $m \geq 6$. Successful teacher recoveries thus begin with a short shared repair segment and only later branch into different valid paths. The shared repair prefix provides a concentrated target for recovery supervision.

4.3 Finding 3: Full-Return Supervision Overcommits

Finding 3

Full-return supervision reduces multi-sample recovery and return diversity compared with shared repair-prefix supervision. Distilling only the shared repair prefix preserves both recovery ability and downstream exploration.

Findings 1 and 2 suggest that recovery supervision should focus on the shared repair prefix. To test this directly, we compare three supervision strategies on the same set of recoverable prefixes: **(i)** a single randomly chosen successful teacher continuation, denoted as single full return; **(ii)** a mixture of multiple full successful continuations, denoted as mixed full returns; **(iii)** only the shared repair prefix, after which the student continues on policy, denoted as shared repair prefix. Figure 1(c) and Table 1 report Rec@1, Rec@8, and return diversity.

Single full-return supervision reduces return diversity and underperforms on Rec@8. Mixed full-return supervision improves Rec@8 and return diversity over a single full return. Shared repair prefix supervision achieves the best Rec@1 and the strongest Rec@8, while preserving much higher return diversity.

5 Methodology

5.1 Mining Recoverable Deep Prefixes

Given a training problem x, we sample a student rollout $\hat{y} \sim \pi_\theta(\cdot \mid x)$. If $\hat{y}$ yields the correct answer, it is used only for the base OPD objective in Equation (2). If $\hat{y}$ is incorrect, we collect candidate deviated prefixes $p_m = \hat{y}_{\leq t_m}$ at a set of depth positions $\mathcal{T}_{d_j}(x)$ within the current depth bucket d_j (see Section A.2 for bucket construction details).

For each candidate p_m, we estimate $\hat{R}_T(p_m)$ and $\hat{R}_S(p_m)$ via Monte Carlo rollouts (Section A.1) and retain only those with high teacher recoverability and a sufficiently large gap:

$$\mathcal{P}_{\mathrm{rec}}(x) = \left\{ p_m \;\middle|\; \hat{R}_T(p_m) \geq \tau_T,\ \hat{R}_S(p_m) \leq \tau_S,\ \hat{G}(p_m) \geq \gamma \right\}. \tag{6}$$

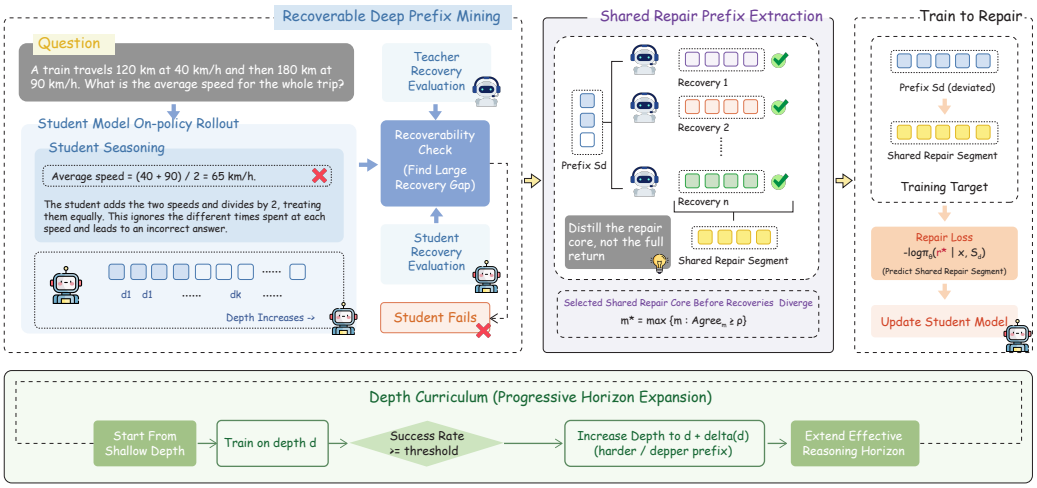


Figure 2: Overview of **CoTrek**.

CoTrek selects prefixes with high teacher recoverability and low student recoverability for repair training. Threshold settings and batch-level handling are specified in Section G.4.

5.2 Shared Repair Prefix Extraction

For each recoverable prefix $p \in \mathcal{P}_{\text{rec}}(x)$, we sample K teacher continuations conditioned on (x, p) and retain the subset $\mathcal{C}(p) = \{c^{(1)}, \ldots, c^{(K_p)}\}$ that reaches the correct answer. Successful teacher recoveries can branch into diverse valid continuations after the initial repair. We therefore extract only the shared repair prefix—the longest initial segment on which successful recoveries still agree.

Concretely, we segment each recovery into reasoning steps $c^{(k)} = (s_1^{(k)}, \ldots, s_{L_k}^{(k)})$ and let $\phi_m(c^{(k)}) = (s_1^{(k)}, \ldots, s_m^{(k)})$ denote its first m steps (see Section A.3 for segmentation rules). We select the largest step count that preserves high cross-recovery agreement:

$$m^\star = \max\left(\{0\} \cup \left\{ m \in \{1, \ldots, \min_k L_k\} \;\middle|\; \min_{1 \le j \le m} \text{Agree}_j(\mathcal{C}(p)) \ge \rho \right\}\right), \tag{7}$$

where $\text{Agree}_m(\mathcal{C}(p))$ measures average pairwise similarity among the first m steps and ρ is a threshold (Section A.3). The shared repair prefix $r^\star(p)$ is then defined as the medoid of $\{\phi_{m^\star}(c^{(k)})\}_{k=1}^{K_p}$. Intuitively, $r^\star(p)$ captures the shared repair segment that redirects reasoning to a reliable path before later valid branches diverge.

5.3 Recovery-Centric Distillation

The training objective of **CoTrek** combines the standard OPD loss with a recovery loss:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{base}}(\theta) + \lambda\, \mathcal{L}_{\text{repair}}(\theta), \tag{8}$$

where $\mathcal{L}_{\text{base}}$ is defined in Equation (2) and the repair loss teaches the student to execute shared repair behavior:

$$\mathcal{L}_{\text{repair}}(\theta) = \mathbb{E}_x \left[\frac{1}{|\mathcal{P}_{\text{rec}}(x)|} \sum_{p \in \mathcal{P}_{\text{rec}}(x)} -\log \pi_\theta(r^\star(p) \mid x, p) \right]. \tag{9}$$

An empty prefix set contributes zero repair loss. **CoTrek** distills only the shared repair core. After producing the repair prefix, the student continues the remaining reasoning on policy from student-visited states. If a mini-batch contains no recoverable prefixes, we set $\mathcal{L}_{\text{repair}} = 0$ and update with $\mathcal{L}_{\text{base}}$ alone (see Section C for additional practical considerations including the choice of λ).

5.4 Recovery-Guided Horizon Expansion

Recovery becomes harder as prefix depth grows. We therefore expand training depth gradually, guided by the depth-wise recovery gap defined in Equation (5).

Let $d_1 < d_2 < \cdots < d_J$ denote ordered depth buckets. Training begins from the shallowest bucket and advances once the student demonstrates reliable recovery at the current depth:

$$d_j \to d_{j+1} \quad \text{if} \quad \hat{R}_S(d_j) \geq \eta \ \text{ or } \ \hat{G}(d_j) \leq \epsilon, \tag{10}$$

where η and ϵ are thresholds. This curriculum keeps training aligned with the student's present recovery ability: the student first learns to repair shallower deviations, then progressively extends this skill to deeper prefixes, improving long-horizon distillation without forcing full long returns from the start. Implementation details of the expansion schedule, including checkpoint frequency and bucket overlap, are provided in Section A.4.

6 Experiments

6.1 Setup

Datasets. We evaluate `CoTrek` on long-CoT mathematical reasoning benchmarks, including AIME 2024, AMC 2023, MATH-500 [28], and OlympiadBench [29]. To analyze long-horizon behavior more directly, we additionally group examples by normalized prefix depth and report depth-wise recovery metrics. We train on decontaminated DeepScaleR-Preview-40K; preprocessing and the 500-problem validation split are detailed in Section G.4.

Evaluation. Accuracy is avg@16 (temperature 0.6, top-p 0.95), averaged over three seeds; see Section G.5 for variance and paired bootstrap tests. To evaluate recovery behavior, we additionally report Rec@1 and Rec@8 from deviated prefixes, the depth-wise recovery gap $\hat{G}(d)$, and return diversity among successful recoveries.

Baselines. We compare `CoTrek` with four groups of baselines. First, we include the base student and off-policy long-CoT distillation baselines, including SFT and P-ALIGN [18]. Second, we compare with standard OPD methods such as MiniLLM [7] and GKD [8]. Third, we include recent OPD variants that improve signal calibration or optimization stability, including SCOPE [19] and GSPO-style OPD [21]. Finally, we compare with recovery variants that use the same recoverable prefixes but replace the shared repair prefix with full teacher returns or remove recovery-guided horizon expansion. These variants isolate the effects of shared repair supervision and horizon expansion.

Implementation details. Unless otherwise specified, we use Qwen3-4B-Base-GRPO [14] as the teacher and Qwen3-1.7B-Base as the student. Recoverability is estimated with $K_T = K_S = 8$ rollouts per prefix, and successful teacher recoveries are sampled with $K = 10$ continuations. OPD comparisons use the same 200-step budget and decoding configuration; we additionally match total compute, including teacher sampling and prefix mining (Section G.2). For `CoTrek`, the repair loss weight is set to $\lambda = 0.4$ and depth buckets follow normalized prefix ratios. Additional training details are provided in Appendix C; measured costs are reported in Appendix G.2.

6.2 Main Results (RQ1)

Table 2 reports final accuracy and recovery behavior across four long-CoT benchmarks. `CoTrek` consistently outperforms standard OPD and recent OPD variants across student models. On Qwen3-1.7B-Base, `CoTrek` improves Vanilla OPD by up to 6.84 percentage points in final accuracy and 9.27 points in Rec@8. The gains are especially clear on harder benchmarks such as AIME 2024 and OlympiadBench, where long derivations create more opportunities for deep-prefix drift. Compared with prefix-oriented distillation, `CoTrek` achieves higher final accuracy. Compared with SCOPE and GSPO-style OPD, `CoTrek` also achieves higher recovery success and return diversity. Additional experiments extend these gains to code and Mistral (Section G.1). At matched compute, `CoTrek` exceeds GSPO-OPD by 2.80 points; the mining proxy retains 51.02 accuracy at 1.13× Vanilla OPD GPU-hours (Section G.2).

Table 2: Main performance comparison on long-CoT reasoning benchmarks. Acc. is final answer accuracy and Rec@8 is multi-sample recovery success from deviated prefixes; both are reported as percentages. The best result is **bolded** and the second-best is underlined. Green subscripts indicate absolute gains over Vanilla OPD within the same student group (percentage points for accuracy and recovery).

Method	AIME24 Acc. ↑	AMC23 Acc. ↑	MATH-500 Acc. ↑	Olympiad Acc. ↑	Avg. Acc. ↑	Rec@8 ↑	Diversity ↑
Qwen3-4B-Base-GRPO (Teacher)	55.00	78.60	83.20	42.80	64.90	86.10	0.66
Qwen3-1.7B-Base (Student)							
Zero-Shot	18.30	42.60	45.10	18.90	31.23	39.80	0.46
SFT	24.70	51.40	55.90	24.30	39.08	48.60	0.42
P-ALIGN	27.60	54.80	58.70	26.10	41.80	52.40	0.45
Vanilla OPD	31.80	58.20	62.40	29.70	45.53	56.20	0.39
SCOPE	34.40↑2.60	60.10↑1.90	64.00↑1.60	31.80↑2.10	47.58↑2.05	60.50↑4.30	0.42↑0.03
GSPO-OPD	35.10↑3.30	60.80↑2.60	64.80↑2.40	32.70↑3.00	48.35↑2.82	62.10↑5.90	0.44↑0.05
`CoTrek`	**38.64**↑6.84	**64.40**↑6.20	**67.70**↑5.30	**36.10**↑6.40	**51.71**↑6.18	**65.47**↑9.27	**0.58**↑0.19
Qwen2.5-3B-Instruct (Student)							
Zero-Shot	22.10	46.80	50.30	21.20	35.10	45.50	0.48
SFT	29.80	55.50	60.10	28.00	43.35	54.40	0.43
P-ALIGN	32.00	58.60	62.70	30.00	45.83	57.90	0.46
Vanilla OPD	36.30	62.40	66.20	33.90	49.70	62.70	0.41
SCOPE	38.10↑1.80	64.00↑1.60	67.60↑1.40	35.40↑1.50	51.28↑1.58	66.20↑3.50	0.43↑0.02
GSPO-OPD	38.70↑2.40	64.20↑1.80	68.10↑1.90	36.20↑2.30	51.80↑2.10	68.80↑6.10	0.46↑0.05
`CoTrek`	**42.00**↑5.70	**67.60**↑5.20	**71.00**↑4.80	**39.20**↑5.30	**54.95**↑5.25	**70.20**↑7.50	**0.57**↑0.16

Table 3: Ablation study on Qwen3-1.7B-Base. Each component corresponds to one empirical finding in Section 4.

Variant	Avg. Acc. ↑	Rec@1 ↑	Rec@8 ↑	Diversity ↑	Gap Red. ↑
Full `CoTrek`	**51.71**	**45.40**	**65.47**	**0.58**	**41.3**
w/o Recoverable Prefix Mining	48.62	40.10	58.64	0.44	24.7
Full Return Supervision	49.13	43.80	59.20	0.31	27.4
Random Repair Segment	48.91	39.70	57.82	0.43	22.9
w/o Horizon Expansion	49.44	41.20	60.16	0.50	29.8
w/o Repair Loss	45.53	31.80	56.20	0.39	0.0

6.3 Model Analysis (RQ2)

To understand why `CoTrek` improves final accuracy, we analyze recovery behavior across training steps and prefix depths. Figure 3b shows that Vanilla OPD improves early continuation but leaves a large recovery gap throughout the mid-to-late depth region. In contrast, `CoTrek` steadily reduces the gap after repair training begins. The recovery improvements accompany the final-accuracy gains in Table 2.

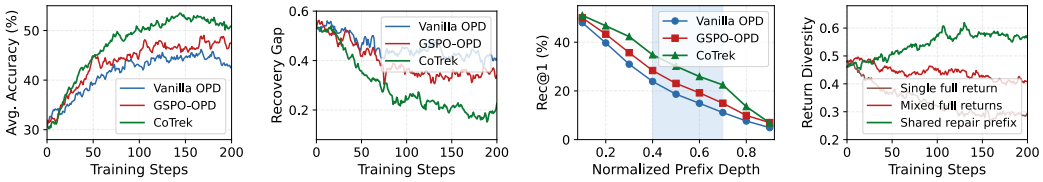
Figure 3c further shows that the improvement is concentrated in the lost-but-recoverable zone. The gain is smaller at very shallow prefixes, where the student can often still continue correctly, and also smaller at very late prefixes, where even the teacher recovery rate drops. The largest gain appears at $d \in [0.4, 0.7]$, precisely the region where our empirical analysis found the largest recovery gap.

Finally, Figure 3d shows that shared repair-prefix supervision preserves return diversity during training. Full-return supervision reduces diversity as training proceeds, while shared repair prefix supervision maintains diverse valid continuations.

6.4 Hyperparameter Analysis (RQ3)

We next study design choices that control recovery learning.

Repair-prefix agreement threshold. We vary the agreement threshold ρ used to extract shared repair prefixes. A small ρ admits longer and less consistent repair targets, which partially reintroduces full-return overcommitment. A very large ρ makes the repair prefix too short and weakens recovery. The best performance appears at an intermediate value.



(a) Final Accuracy (b) Recovery Gap (c) Depth-wise Recovery (d) Return Diversity

Figure 3: Training dynamics and recovery behavior of **CoTrek**. (a) **CoTrek** reaches higher final accuracy than Vanilla OPD and GSPO-OPD. (b) **CoTrek** closes the recovery gap faster and more stably during training. (c) The recovery gain is most visible in mid-to-late prefix depths, matching the lost-but-recoverable zone identified in Section 4. (d) Shared repair prefix supervision preserves higher return diversity than full-return supervision.

Recovery-gap threshold. We vary the recovery-gap threshold γ for prefix mining. Training on low-gap prefixes gives limited gains, since the student can already repair many of them. Training on overly hard prefixes also hurts, since teacher recovery becomes sparse and noisy. The best regime is the lost-but-recoverable zone, where the teacher can still recover but the student cannot.

Depth expansion strategy. We compare recovery-guided horizon expansion with fixed-depth and random-depth schedules. Fixed shallow training improves early recovery but fails to scale. Fixed deep training is unstable because recoverable prefixes are sparse at the beginning of training. Recovery-guided expansion gives the best tradeoff across prefix depths.

6.5 Ablation Study (RQ4)

Table 3 ablates the core components of **CoTrek**. Removing recoverable prefix mining leads to a clear drop in both accuracy and recovery success, indicating that failed prefixes are not equally useful. Replacing the shared repair prefix with a full teacher return reduces Rec@1, Rec@8, and diversity, consistent with our finding of full-return overcommitment. Using random repair segments also hurts performance. Finally, removing recovery-guided horizon expansion mainly hurts deeper prefix regimes.

7 Conclusion

We proposed **CoTrek**, a recovery-centric framework for long-CoT on-policy distillation. By mining recoverable deep prefixes and distilling shared repair segments, **CoTrek** teaches the student to return to effective reasoning paths after deviation. Experiments show consistent gains in final accuracy and recovery behavior, highlighting recovery as a key factor for scalable long-CoT distillation.

Acknowledgements and Disclosures

This work was supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2026A1515010176).

The authors declare no competing interests.

References

[1] Aaron Jaech and OpenAI. Openai o1 system card, 2024.

[2] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20275–20321, 2025.

[3] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. 2025.

[4] NovaSky Team. Sky-t1: Fully open-source reasoning model with o1-preview performance in $450 budget. https://novasky-ai.github.io/posts/sky-t1, 2025. Accessed 2025-01-09.

[5] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. 2024.

[6] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. O1 replication journey: A strategic progress report – part 1. 2024.

[7] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. 2023.

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

[9] Kevin Lu and Thinking Machines Lab. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation, 2025. Thinking Machines Lab: Connectionism.

[10] 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. 2026.

[11] Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. 2026.

[12] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. 2026.

[13] Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models. 2026.

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

[15] GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chengxing Xie, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chen Li, Chenghua Huang, Chengwei Hu, Chenhui Zhang, Chenzheng Zhu, Congfeng Yin, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding,

Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xunkai Zhang, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering. 2026.

[16] 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.

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

[18] Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. 2026.

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

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

[21] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. 2025.

[22] Qiguang Chen, Yantao Du, Ziniu Li, Jinhao Liu, Songyao Duan, Jiarui Guo, Minghao Liu, Jiaheng Liu, Tong Yang, Ge Zhang, Libo Qin, Wanxiang Che, and Wenhao Huang. The molecular structure of thought: Mapping the topology of long chain-of-thought reasoning. 2026.

[23] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. 2025.

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

[25] Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying opd: Length inflation and stabilization strategies for large language models. 2026.

[26] Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. 2026.

[27] Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. 2026.

[28] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. NeurIPS Datasets and Benchmarks, 2021.

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

[30] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. arXiv preprint arXiv:2403.07974, 2024.

[31] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[32] Mistral AI. Mistral-7B-Instruct-v0.3: Model card. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3. Accessed November 3, 2026.

[33] Agentica. DeepScaleR-Preview-Dataset. https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset, 2025. Accessed November 3, 2026.

[34] LiveCodeBench. LiveCodeBench Code Generation Lite: Dataset card and release versions. https://huggingface.co/datasets/livecodebench/code_generation_lite. Release v5; accessed November 3, 2026.

[35] EvalPlus. HumanEval+: Dataset card. https://huggingface.co/datasets/evalplus/humanevalplus. Accessed November 3, 2026.

[36] Sentence Transformers. all-MiniLM-L6-v2: Model card. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. Accessed November 3, 2026.

[37] Beijing Academy of Artificial Intelligence. bge-base-en-v1.5: Model card. https://huggingface.co/BAAI/bge-base-en-v1.5. Accessed November 3, 2026.

[38] Alibaba DAMO Academy. gte-large: Model card. https://huggingface.co/thenlper/gte-large. Accessed November 3, 2026.

A Additional Method Details

This appendix provides implementation details omitted from the main paper, covering recoverability estimation, prefix construction and filtering, shared repair prefix extraction, depth scheduling, and computational cost.

A.1 Recoverability Estimation

Teacher and student recoverability $R_T(p)$ and $R_S(p)$ are defined in Equation (3). In practice, we estimate them via Monte Carlo sampling:

$$\hat{R}_T(p) = \frac{1}{K_T} \sum_{k=1}^{K_T} \mathbf{1}\left[\mathrm{Ans}(y_T^{(k)}) = a^\star\right], \qquad \hat{R}_S(p) = \frac{1}{K_S} \sum_{k=1}^{K_S} \mathbf{1}\left[\mathrm{Ans}(y_S^{(k)}) = a^\star\right], \tag{11}$$

where $y_T^{(k)} \sim \pi_T(\cdot \mid x, p)$ and $y_S^{(k)} \sim \pi_\theta(\cdot \mid x, p)$. The estimated recovery gap is $\hat{G}(p) = \hat{R}_T(p) - \hat{R}_S(p)$, and for depth bucket d we average over all deviated prefixes: $\hat{G}(d) = \frac{1}{|\mathcal{P}_d|} \sum_{p \in \mathcal{P}_d} \hat{G}(p)$.

Unless otherwise specified, teacher and student rollouts share the same decoding configuration used at final evaluation. Answer verification uses exact match for deterministic answers and benchmark-specific normalization for mathematical expressions.

A.2 Candidate Prefix Construction and Filtering

Depth buckets. For a failed student rollout $\hat{y}$ of length T, we convert each position t to a normalized depth ratio $r_t = t/T$ and partition the ratio interval into J ordered buckets $\mathcal{B}_1, \ldots, \mathcal{B}_J$. The candidate set for bucket d_j is $\mathcal{T}_{d_j}(x) = \{t \mid t/T \in \mathcal{B}_j\}$. In the main experiments, we place a small number of evenly spaced truncation points inside each active bucket to keep recoverability estimation tractable.

Filtering. A candidate prefix p enters the recovery training set $\mathcal{P}_{\mathrm{rec}}(x)$ only if $\hat{R}_T(p) \geq \tau_T$, $\hat{R}_S(p) \leq \tau_S$, and $\hat{G}(p) \geq \gamma$ (cf. Equation (6)). The first condition ensures teacher viability, the second excludes prefixes the student can already repair, and the third retains only prefixes with a sufficiently informative gap. The threshold configuration and batch-level handling are specified in Section G.4.

This procedure is summarized in Algorithm 1.

A.3 Shared Repair Prefix Extraction

This subsection details shared repair prefix extraction, summarized in Algorithm 2.

Step segmentation. Each successful teacher recovery $c^{(k)} \in \mathcal{C}(p)$ is segmented into reasoning steps $c^{(k)} = (s_1^{(k)}, \ldots, s_{L_k}^{(k)})$ using benchmark-specific rules: sentence boundaries and explicit step delimiters for natural-language traces, model-emitted step markers when available, and statement blocks for code-like reasoning. This step-level view is used only for repair-prefix extraction; the base OPD loss operates on full token sequences.

Agreement and shared length. For the first m steps, we define the step prefix $\phi_m(c^{(k)}) = (s_1^{(k)}, \ldots, s_m^{(k)})$ and measure cross-recovery agreement:

$$\mathrm{Agree}_m(\mathcal{C}(p)) = \frac{2}{K_p(K_p-1)} \sum_{1 \leq i < j \leq K_p} \mathrm{Sim}\Big(\phi_m(c^{(i)}),\ \phi_m(c^{(j)})\Big), \tag{12}$$

where Sim is embedding cosine similarity over step strings. The default encoder is all-MiniLM-L6-v2 (22M parameters, 384 dimensions); alternative encoders and a lexical control are evaluated in Section G.3. The shared repair length is the largest initial interval of sustained agreement, with $m^\star = 0$ if no nonempty segment passes, as defined in Equation (7) and implemented in Algorithm 2.

Medoid selection. Among the repair-prefix candidates $\mathcal{R}(p) = \{\phi_{m^\star}(c^{(k)})\}_{k=1}^{K_p}$, we select the medoid:

$$r^\star(p) = \arg\min_{r \in \mathcal{R}(p)} \sum_{r' \in \mathcal{R}(p)} \mathrm{Dist}(r,\, r'), \tag{13}$$

where Dist is the sequence distance induced by Sim.

Repair-prefix supervision. Full teacher returns often diverge substantially after the initial repair stage: we typically observe $\phi_{m^\star}(c^{(i)}) \approx \phi_{m^\star}(c^{(j)})$ yet $c^{(i)} \not\approx c^{(j)}$. Full-return supervision reduces continuation diversity. Shared repair-prefix supervision targets common re-entry behavior and leaves later reasoning flexible. Algorithm 2 provides the complete pseudocode.

A.4 Recovery-Guided Horizon Expansion

Training begins from the shallowest bucket d_1 and advances to d_{j+1} when the student becomes reliable at d_j, according to the rule in Equation (10). Bucket-level statistics are computed as

$$\hat{R}_S(d_j) = \frac{1}{|\mathcal{P}_{d_j}|} \sum_{p \in \mathcal{P}_{d_j}} \hat{R}_S(p), \qquad \hat{G}(d_j) = \frac{1}{|\mathcal{P}_{d_j}|} \sum_{p \in \mathcal{P}_{d_j}} \hat{G}(p). \tag{14}$$

To avoid oscillation, we evaluate the expansion condition only at validation checkpoints rather than every update. Once a deeper bucket is activated, shallower buckets remain in the sampling pool with reduced probability so that earlier recovery skills are maintained.

Algorithm 1 MINERECOVERABLEPREFIXES: Recoverability Estimation and Prefix Filtering

Require: Failed rollout $\hat{y}$, depth bucket d_j, student π_θ, teacher π_T
Require: Sample counts K_T, K_S, thresholds τ_T, τ_S, γ

$\mathcal{P}_{\text{rec}} \leftarrow \emptyset$
for all $t \in \mathcal{T}_{d_j}(x)$ **do**
  $p \leftarrow \hat{y}_{\le t}$
  Sample K_T teacher and K_S student continuations from p
  Compute $\hat{R}_T(p)$ and $\hat{R}_S(p)$ via Equation (11)
  if $\hat{R}_T(p) \ge \tau_T$ **and** $\hat{R}_S(p) \le \tau_S$ **and** $\hat{R}_T(p) - \hat{R}_S(p) \ge \gamma$ **then**
    $\mathcal{P}_{\text{rec}} \leftarrow \mathcal{P}_{\text{rec}} \cup \{p\}$
  end if
end for
return $\mathcal{P}_{\text{rec}}$

Algorithm 2 EXTRACTREPAIRPREFIX: Shared Repair Prefix Extraction

Require: Prefix p, teacher π_T, agreement threshold ρ, sample count K

Sample K teacher continuations from (x, p); retain successful ones as $\mathcal{C}(p)$
if $|\mathcal{C}(p)| < 2$ **then**
  return no repair target
end if
Segment each $c^{(k)} \in \mathcal{C}(p)$ into reasoning steps
$m^\star \leftarrow 0$
for $m = 1$ to $\min_k L_k$ **do**
  if $\text{Agree}_m(\mathcal{C}(p)) \ge \rho$ **then**
    $m^\star \leftarrow m$
  else
    break
  end if
end for
if $m^\star = 0$ **then**
  return no repair target
end if
Select medoid $r^\star(p)$ from $\{\phi_{m^\star}(c^{(k)})\}_{k=1}^{K_p}$
return $r^\star(p)$

Recovery difficulty grows with depth: if training starts immediately on the deepest prefixes, repair supervision becomes sparse and noisy. Gradual expansion first closes the gap at shallower depths, then extends the horizon only after the student has developed stable recovery behavior, making long-horizon learning substantially more stable in practice.

A.5 Computational Cost

The extra cost of `CoTrek` comes from recoverability estimation and teacher recovery sampling. For each failed rollout with M candidate prefix positions, recoverability estimation requires $O(M(K_T + K_S))$ rollouts. For each selected recoverable prefix, repair extraction costs $O(K + K^2)$, where the quadratic term arises from pairwise agreement computation. Repair extraction applies to the filtered prefixes with large recovery gaps.

B Appendix Algorithms

C Additional Training Details

Base OPD instantiation. $\mathcal{L}_{\text{base}}$ denotes the base on-policy distillation objective. In our experiments, we instantiate it with a standard token-level or sequence-level OPD objective depending on the backbone setting. The framework of `CoTrek` is agnostic to this choice, since the repair loss operates on recoverable student prefixes and can be added on top of either variant.

Algorithm 3 Recovery-Guided Horizon Expansion

Require: Current bucket index j, thresholds η, ϵ, total buckets J

Compute $\hat{R}_S(d_j)$ and $\hat{G}(d_j)$ via Equation (14)
if $\hat{R}_S(d_j) \geq \eta$ **or** $\hat{G}(d_j) \leq \epsilon$ **then**
    $j \leftarrow \min(j+1,\ J)$ ▷ Advance to deeper bucket
end if
return j

Repair weight λ. The scalar λ in Equation (8) controls the strength of recovery supervision and is selected from a small validation grid. A moderate value is preferred: too small and recovery learning remains weak; too large and training overfocuses on repair prefixes at the expense of the base continuation objective.

Sampling teacher recoveries. For each recoverable prefix, we sample teacher recoveries with the same decoding family used in teacher evaluation, using nonzero temperature and standard nucleus or top-k truncation to encourage diversity. Only successful recoveries enter $\mathcal{C}(p)$. Pairwise agreement requires at least two successes; prefixes with fewer successes or no nonempty agreed segment are skipped for repair extraction in that update.

Unrecoverable prefixes. Not all failed prefixes are recoverable. Typical cases excluded by our filtering include $\hat{R}_T(p) \approx 0$ (the teacher itself cannot solve from the prefix) and $\hat{R}_T(p) \approx \hat{R}_S(p)$ (the gap is uninformative). Such prefixes contribute only to $\mathcal{L}_{\text{base}}$.

Batches without recoverable prefixes. If a mini-batch contains no recoverable prefixes, we set $\mathcal{L}_{\text{repair}} = 0$ and update with $\mathcal{L}_{\text{base}}$ alone, avoiding unstable gradients from sparse repair targets.

D Additional Evaluation Details

Recovery success. For a set of deviated prefixes $\mathcal{P}$, we define single-sample and multi-sample recovery success:

$$\text{Rec@1} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbf{1}[\text{Ans}(y_p) = a^\star], \qquad \text{Rec@}k = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbf{1}\left[\exists\, i \leq k :\ \text{Ans}(y_p^{(i)}) = a^\star\right], \tag{15}$$

where $y_p \sim \pi_\theta(\cdot \mid x, p)$ and $y_p^{(1)}, \ldots, y_p^{(k)}$ are independent student samples.

Return diversity. To evaluate whether the student overcommits to one return branch, we compute the average pairwise distance among successful student recoveries from the same prefix:

$$\text{Div}(p) = \frac{2}{|\mathcal{Y}_p^{\text{succ}}|(|\mathcal{Y}_p^{\text{succ}}| - 1)} \sum_{i<j} \text{Dist}(y_p^{(i)},\, y_p^{(j)}), \tag{16}$$

where $\mathcal{Y}_p^{\text{succ}}$ is the set of successful recoveries. Higher scores indicate that the student preserves more diverse valid continuations after repair.

Depth-wise breakdown. For each depth bucket d, we report $\text{Acc}(d)$, $\text{Rec@1}(d)$, $\text{Rec@}k(d)$, and $\hat{G}(d)$. This breakdown verifies that gains are concentrated in regimes where the recovery gap is large.

E Discussion of Design Choices

Supervision target. Full-return supervision trains the student on an entire teacher trajectory. Successful teacher recoveries diverge after an initial shared repair stage (Section A.3). Full-return supervision reduces continuation diversity. Shared repair-prefix supervision targets common re-entry behavior and leaves later reasoning flexible.

Student-visited states. Recovery training conditions teacher-derived repair targets on student-generated deviated prefixes. The conditioning states are sampled from the current student policy.

Depth scheduling. If training starts immediately at maximum depth, repair supervision is sparse and noisy because very deep prefixes rarely satisfy the recoverability filter. Gradual expansion first closes the gap at shallower depths, building stable recovery behavior before tackling harder regimes.

F Limitations, Statistical Significance, and Compute Resources

Limitations. `CoTrek` relies on the existence of recoverable deviated prefixes. When the teacher cannot recover from a prefix, or when successful teacher recoveries are highly inconsistent, the extracted repair signal may become weak. The method also introduces additional sampling cost for estimating prefix recoverability and extracting shared repair segments. Our evaluation covers mathematical reasoning and code generation.

Statistical significance. Accuracy is evaluated as avg@16 over three random seeds. The $\pm$ values in Table 9 denote standard deviations across runs. Paired bootstrap resampling over problems and seeds gives $p < 0.01$ against GSPO-OPD on four-benchmark average accuracy and $p = 0.03$ on AIME24 alone. For recovery metrics, we additionally bootstrap evaluated prefixes with 1,000 resamples.

Compute resources. The training-cost comparison uses 8 NVIDIA H100 GPUs with 80GB memory over 200 training steps per run, including student rollout generation, teacher recovery sampling, and parameter updates. GPU-hours, wall-clock time, and relative training-token counts are given in Table 6. Recoverability estimation is the main additional cost of `CoTrek`, since it requires $K_T = K_S = 8$ continuation samples per candidate prefix. The total compute cost scales linearly with the number of evaluated prefixes and recovery samples, as discussed in Appendix A.5.

Broader impacts. `CoTrek` aims to improve the efficiency of distilling long reasoning abilities into smaller language models. This can reduce the cost of deploying reasoning models and make long-CoT reasoning more accessible. At the same time, stronger reasoning models may also increase the capability of systems used for harmful purposes if deployed without safeguards. We therefore recommend using `CoTrek` together with standard model safety evaluation, controlled release practices, and downstream monitoring.

G Additional Experiments and Controls

This appendix evaluates the generality, sampling cost, extraction robustness, and recovery mechanism of `CoTrek`.

G.1 Generality beyond Mathematics and Qwen

Code generation. We evaluate a Qwen3-1.7B student on LiveCodeBench-v5 [30] and HumanEval+ [31]. For code generation, we segment teacher recoveries at AST statement boundaries. Table 4 compares the methods. Generic newline segmentation reaches 38.90 average accuracy, while AST boundaries give 39.65.

Table 4: Code generation results with a Qwen3-1.7B student. Scores are percentages.

Method	LiveCodeBench-v5	HumanEval+	Avg.
Vanilla OPD	18.40	51.20	34.80
GSPO-OPD	19.70	53.00	36.35
`CoTrek`	**22.60**	**56.70**	**39.65**

Cross-recovery agreement at step three is 0.81 in mathematics and 0.68 in code, while the mean extracted repair length is 3.4 and 2.1 steps, respectively. Code recoveries branch earlier, leaving a smaller shared repair core.

A fully Mistral-family pair. We use Mistral-Small-24B-Instruct as the teacher and Mistral-7B-Instruct-v0.3 [32] as the student, applying the Qwen-selected hyperparameters without retuning. Table 5 averages performance over AIME24, MATH-500, and OlympiadBench. The lost-but-recoverable region again occurs at normalized depth 0.4–0.7. Cross-recovery agreement exceeds 0.79 for the first three steps and falls below 0.40 after step six, reproducing the two observations that motivate the method.

Table 5: Transfer to a fully Mistral-family teacher–student pair without hyperparameter retuning. Accuracy and Rec@8 are percentages.

Method	AIME24	MATH-500	Olympiad	Avg.	Rec@8
Vanilla OPD	22.50	57.80	26.40	35.57	51.30
GSPO-OPD	24.60	59.90	28.20	37.57	55.80
`CoTrek`	**28.10**	**63.20**	**31.60**	**40.97**	**60.42**

G.2 Compute, Tokens, and a Lightweight Mining Proxy

Measured training cost. We measure per-run costs on 8 H100-80GB GPUs over 200 training steps (Table 6). Total cost includes training, prefix mining, and teacher recovery sampling. The 67.5 GPU-hours of `CoTrek` comprise 48.3 for training, 12.4 for prefix mining, and 6.8 for teacher recovery sampling. Relative to Vanilla OPD, `CoTrek` uses 1.38× the training tokens and approximately 1.46× the GPU-hours. Giving GSPO-OPD the same total compute raises its average accuracy to 48.91, while `CoTrek` reaches 51.71, a 2.80-point difference.

Table 6: Per-run training cost and performance. Relative tokens are normalized to Vanilla OPD; GPU-hours include sampling and mining. Accuracy per GPU-hour uses average accuracy in percentage points.

Method	GPU-h	Wall-clock	Rel. tokens	Avg. Acc.	Acc./GPU-h
Vanilla OPD	46.2	5.8 h	1.00×	45.53	0.986
SCOPE	49.1	6.1 h	1.05×	47.58	0.969
GSPO-OPD	51.8	6.5 h	1.09×	48.35	0.933
GSPO-OPD, matched budget	67.4	8.4 h	1.38×	48.91	0.726
`CoTrek`	67.5	8.4 h	1.38×	**51.71**	0.766
`CoTrek` + verified proxy	52.1	6.5 h	1.13×	51.02	0.979

A lightweight recovery-gap proxy. We train a two-layer MLP on the student's last-layer hidden state at the prefix boundary, using Monte Carlo estimates of the recovery gap as labels. Probe training reuses 40K prefixes collected during the first 30 training steps. The probe reaches AUC 0.83 for distinguishing recoverable from unrecoverable prefixes. We use it to pre-screen candidates, then verify the top 20% with Monte Carlo rollouts before applying the recoverability filter.

Table 7 compares the verified proxy and a cheaper proxy-only variant. Pre-screening with verification reduces mining rollouts by 62% and achieves 51.02 average accuracy, preserving approximately 89% of the full method's gain over Vanilla OPD. Its total GPU-hours are 1.13× Vanilla OPD. Removing verification admits false positives and lowers accuracy. The proxy is used for training-time prefix mining. Inference uses the student policy directly; generation lengths are evaluated in the length-control study below.

Table 7: Mining proxy variants. Mining rollout counts are relative to full Monte Carlo mining.

Setting	Mining rollouts	GPU-h	Avg. Acc.
Full Monte Carlo	1.00×	67.5	51.71
Proxy pre-screen, verify top 20%	0.38×	52.1	51.02
Proxy only, no verification	0.12×	48.4	49.87
Vanilla OPD	—	46.2	45.53

G.3 Robustness to the Similarity Backend

The default similarity backend is all-MiniLM-L6-v2, with 22M parameters and 384-dimensional embeddings. We replace it with bge-base-en-v1.5, gte-large, and ROUGE-L token overlap, and rerun the pipeline on Qwen3-1.7B. Table 8 compares repair-length statistics and downstream performance. Across the three embedding encoders, mean repair length changes by at most 0.2 steps and accuracy ranges from 51.55 to 51.83. The lexical backend preserves most of the gain, whereas a random same-length repair segment loses 2.80 points relative to the default.

Table 8: Sensitivity to the similarity backend. Repair length is measured in reasoning steps; accuracy and Rec@8 are percentages.

Similarity backend	Mean $m^\star$	Std.	Avg. Acc.	Rec@8
all-MiniLM-L6-v2, 22M	3.4	1.1	51.71	65.47
bge-base-en-v1.5, 110M	3.6	1.2	51.55	65.10
gte-large, 335M	3.5	1.1	51.83	65.72
ROUGE-L, no embedding	3.1	1.4	50.94	63.88
Random segment, matched length	—	—	48.91	57.82

G.4 Training Data and Threshold Selection

Training prompts come from DeepScaleR-Preview-40K [33]. Before training, we compare them against AIME 2024, AMC 2023, MATH-500, and OlympiadBench using exact matching, 8-gram overlap, and embedding-based near-duplicate detection with cosine threshold 0.9. These filters remove 412 problems, including MATH training problems appearing in MATH-500. No overlaps are detected by the same filters after removal. We reserve a 500-problem held-out split of the training pool and use it for all threshold selection.

One selected configuration is used across the four mathematical benchmarks and both Qwen students:

$$\tau_T = 0.5, \quad \tau_S = 0.125, \quad \gamma = 0.35, \quad \rho = 0.75, \quad \lambda = 0.4, \quad \eta = 0.6, \quad \epsilon = 0.1.$$

The Qwen2.5-3B and Mistral experiments use this configuration without retuning. The default viability thresholds impose $\hat{G}(p) \geq 0.375$, above $\gamma = 0.35$. When too few prefixes qualify, the batch fallback lowers only γ with τ_T and τ_S fixed; default prefix eligibility is unchanged. If no prefix qualifies, we set $\mathcal{L}_{\text{repair}} = 0$ and continue training with $\mathcal{L}_{\text{base}}$.

G.5 Statistical Protocol and Variance

We evaluate accuracy as avg@16 at temperature 0.6 and top-p 0.95 over three random seeds. Table 9 gives the corresponding means and standard deviations. Paired bootstrap resampling over problems and seeds yields $p < 0.01$ for `CoTrek` against GSPO-OPD on four-benchmark average accuracy and $p = 0.03$ on AIME24 alone. The main accuracy comparison uses the four-benchmark average.

Table 9: Three-seed performance on Qwen3-1.7B. Values are percentages; $\pm$ denotes standard deviation across runs.

Method	AIME24	Avg. Acc.	Rec@8
Vanilla OPD	31.80 ± 1.68	45.53 ± 0.82	56.20 ± 1.13
SCOPE	34.40 ± 1.61	47.58 ± 0.79	60.50 ± 1.06
GSPO-OPD	35.10 ± 1.55	48.35 ± 0.74	62.10 ± 0.98
`CoTrek`	$\mathbf{38.64 \pm 1.42}$	$\mathbf{51.71 \pm 0.61}$	$\mathbf{65.47 \pm 0.87}$

G.6 Repair-Core Analysis and Mechanism Controls

Repair-core composition. We inspect 500 sampled repair cores. Of these, 78% begin with explicit error identification followed by a restatement of the violated principle, while 6% simply give the next arithmetic step. The following example shows a student prefix and its extracted repair core.

Student prefix: "Average speed = 40 plus 90, divided by 2, is 65 km/h. So over 300 km the total time is 300/65 = 4.62 h..."

Shared repair core, $m^\star = 3$**:** "Wait, average speed is total distance over total time, not the mean of the two speeds. Total distance is 120 + 180 = 300 km. The first leg takes 120/40 = 3 h and the second takes 180/90 = 2 h."

The successful continuations then diverge: one divides 300 by 5, another uses the weighted harmonic mean, and another checks that the answer lies between 40 and 90. The shared core identifies and repairs the error, after which successful continuations follow different valid paths.

Compact-prefix control. We quantify the influence of context length by comparing original deviated prefixes with compact frontier-model summaries that preserve their reasoning content. As shown in Table 10, reducing mean prefix length to approximately one quarter increases student recoverability by 2.7 points, leaving a recovery gap of approximately 50 points.

Table 10: Recoverability after compacting the same flawed reasoning content. Recovery scores are percentages.

Prefix form	Mean length (tokens)	Teacher Rec.	Student Rec.
Original drifted prefix	3,180	61.4	9.7
Compact summary	790	63.1	12.4

Generation-length control. Mean generation lengths are 6,842 tokens for Vanilla OPD, 7,105 for GSPO-OPD, and 7,238 for `CoTrek`. Thus, `CoTrek` produces approximately 5.8% more tokens than Vanilla OPD. Truncating its generations to Vanilla OPD's mean token budget yields 50.88 average accuracy, 5.35 points above Vanilla OPD.

Curriculum on plain OPD. Table 11 isolates the effect of depth scheduling. Adding the curriculum to plain OPD yields 0.68 points, whereas horizon expansion adds 2.27 points when combined with the repair target. The curriculum strengthens recovery learning by scheduling the repair signal.

Table 11: Depth curriculum with and without recovery supervision on Qwen3-1.7B.

Variant	Avg. Acc. (%)
Vanilla OPD	45.53
Vanilla OPD + depth curriculum	46.21
`CoTrek` w/o horizon expansion	49.44
`CoTrek`	**51.71**

General ability. Table 12 evaluates broader model ability after recovery training. MMLU and IFEval scores remain close to the base student, while GSM8K improves. These results provide a complementary check beyond the competition-math benchmarks.

Table 12: General-ability checks on the Qwen3-1.7B student. Scores are percentages.

Model	MMLU	IFEval	GSM8K
Base student	62.3	54.2	71.4
Vanilla OPD	61.8	53.1	76.9
`CoTrek`	62.1	54.0	78.3

Long-Horizon Scaling Analysis. Grouping evaluation examples by generated length gives accuracy gains over Vanilla OPD of 2.1 points below 4K tokens, 4.7 at 4K–8K, 7.3 at 8K–16K, and 8.9 above 16K. Across student models, the accuracy gains are 6.18 points for Qwen3-1.7B, 5.25 for Qwen2.5-3B, and 5.40 for Mistral-7B, while the verified proxy achieves 51.02 average accuracy at 1.13× Vanilla OPD GPU-hours.

H Additional Asset Sources and License Declarations

Table 13 lists the public assets used in the additional experiments and the licenses declared by their providers for the named releases. The LiveCodeBench evaluation code is released under MIT.

Table 13: Sources and license declarations for additional CoTrek assets.

Asset	Source	Declared license
DeepScaleR-Preview dataset	Dataset card [33]	MIT
LiveCodeBench, release v5	Official release card [34]	Data: `cc`
HumanEval+	Dataset card [35]	Apache-2.0
Mistral-7B-Instruct-v0.3	Model card [32]	Apache-2.0
all-MiniLM-L6-v2	Model card [36]	Apache-2.0
bge-base-en-v1.5	Model card [37]	MIT
gte-large	Model card [38]	MIT

HumanEval+ is evaluated using EvalPlus [31], whose official repository declares Apache-2.0.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the recovery-centric framework, empirical findings, and performance gains. These are supported by Sections 4 to 6, with generality, cost, robustness, and length controls in Appendix G.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Appendix F discusses reliance on teacher recoverability, additional sampling cost, and the evaluation scope of mathematical reasoning and code generation rather than multi-turn agentic or multimodal settings.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete proof?

   Answer: [N/A].

   Justification: The paper is empirical and algorithmic. It does not present formal theoretical theorems or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper?

   Answer: [No].

   Justification: Complete optimizer settings, batch sizes, checkpoint identifiers, and reproduction code are not provided.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The experimental code, configurations, and reproduction scripts are not publicly available.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [No].

   Justification: Complete optimizer settings, batch sizes, and checkpoint identifiers are not specified.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Table 9 reports means and standard deviations over three random seeds. Appendix G.5 reports paired bootstrap tests over problems and seeds; recovery metrics additionally use bootstrap resampling over prefixes, as described in Appendix F.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

Answer: [Yes].

Justification: We report GPU type and memory, GPU-hours, wall-clock time, relative training tokens, and the breakdown of sampling and mining costs in Appendix G.2, alongside the scaling analysis in Appendix A.5.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes].

   Justification: The work uses public benchmarks and model checkpoints for research on reasoning distillation. We do not collect private data, involve human subjects, or conduct experiments that violate the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: We discuss potential efficiency benefits and possible misuse risks of stronger reasoning models in Appendix F.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A].

    Justification: The paper does not release new pretrained models, datasets, or high-risk assets. It proposes a training framework evaluated on public benchmarks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: Licenses and upstream terms are not specified for all benchmark assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce a new dataset, benchmark, or released model checkpoint. The method uses existing public benchmarks and model families.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

15. **Institutional review board approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board approvals or an equivalent approval were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human-subject research, so IRB approval is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [Yes].

Justification: LLMs are central to the proposed method, serving as the teacher and student policies in on-policy distillation. The paper describes the teacher-student setup, rollout generation, recoverability estimation, and repair-prefix distillation in Sections 3, 5 and 6.1.