

Distilling Long-CoT Reasoning through Collaborative Step-wise Multi-Teacher Decoding

Taewon Yun¹, Jisu Shin¹, Jeonghwan Choi¹, Seunghwan Bang², Hwanjun Song^{1*}

¹Korea Advanced Institute of Science and Technology

²Ulsan National Institute of Science and Technology

{ytaewon0415, songhwanjun}@kaist.ac.kr

Abstract

Distilling large reasoning models is essential for making Long-CoT reasoning practical, as full-scale inference remains computationally prohibitive. Existing curation-based approaches select complete reasoning traces post-hoc, overlooking collaboration among heterogeneous teachers and lacking dynamic exploration, which leads to redundant sampling and missed complementary reasoning. We introduce CoRD, a collaborative multi-teacher decoding framework that performs step-wise reasoning synthesis guided by predictive perplexity-based scoring and beam search. This enables heterogeneous LRMs to jointly construct coherent reasoning trajectories while efficiently preserving diverse, high-potential hypotheses. Experiments show that CoRD produces higher-quality reasoning data and achieves near teacher-level student performance with fewer, structured supervision signals, without substantial efficiency overhead. CoRD further generalizes well to out-of-domain and open-ended settings. The dataset and model are available at <https://github.com/DISL-Lab/CoRD>.

1 Introduction

Rapid progress in large reasoning models (LRMs), such as Deepseek-R1 (Guo et al., 2025), has unlocked new capabilities beyond conventional language understanding, enabling complex problem solving (Plaat et al., 2024; Li et al., 2025a). The key lies in *test-time scaling*, which enhances reasoning by allowing models to deliberate longer, explore broader solution paths, and allocate more computation, often leading to long chain-of-thought (Long-CoT) reasoning (Qu et al., 2025; Chen et al., 2024). Yet, the high computational cost and complexity of LRMs hinder deployment, making reasoning distillation into smaller models essential for real-world applications and a growing focus of research (Zhang et al., 2025; Li et al., 2025a).

The goal of *reasoning distillation* is to extract high-quality reasoning trajectories from teacher models and transfer them into smaller students through sequence-level knowledge distillation (Kim and Rush, 2016). Here, a reasoning trajectory refers to a structured sequence of intermediate steps, including strategic shifts, reflective self-corrections, and hypothesis revisions, that collectively lead to the final solution. However, identifying high-quality trajectories is particularly challenging in the Long-CoT setting, as they often span thousands of tokens and evolve through dynamic *Aha moments* (Guo et al., 2025). While approaches based on process reward models (PRMs) or Monte Carlo Tree Search (MCTS) are effective for short and static reasoning tasks (Park et al., 2025; Yao et al., 2025; Yin et al., 2025), they become impractical when applied to Long-CoT reasoning: reward shaping prematurely eliminates reasoning paths that may initially appear suboptimal but are essential for transferring deliberative reasoning patterns, and the search space grows exponentially with trajectory length.

In this regard, recent studies, such as S1 (Zhang et al., 2025) and LIMO (Ye et al., 2025), have adopted a *curation*-based approach, which first generates complete candidate reasoning traces from multiple (or even identical) teacher models and then selects high-quality ones for distillation. Despite their simplicity, they fail to harness the collaborative potential of multiple heterogeneous teacher models to jointly discover complementary reasoning strategies and compose novel solution paths that no single teacher could produce in isolation. That is, they waste computation on discarded candidates and inherently lack the ability to adjust exploration dynamically due to their post-hoc design.

To address these limitations, we propose CoRD (Collaborative Reasoning Decoding) in Figure 1, a paradigm shift toward a collaborative step-wise decoding process driven by multi-teacher interaction, enabling heterogeneous teachers to jointly construct

*Corresponding Author.

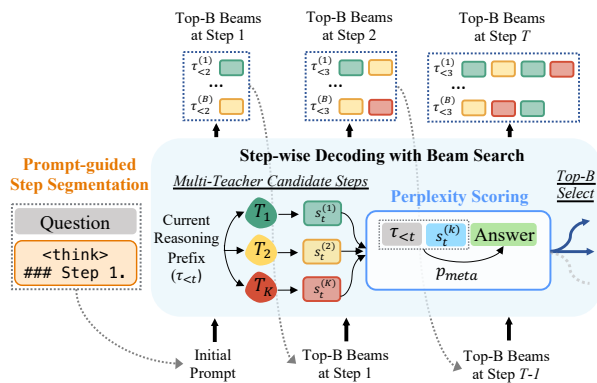


Figure 1: Overview of CoRD: Teacher LRMs collaboratively decode reasoning steps via prompt-guided segmentation. At each step, candidate steps are evaluated via predictive perplexity, retaining the top- B reasoning trajectories for subsequent decoding. The gray dotted line indicates the auto-regressive flow of reasoning steps.

strategically evolving reasoning trajectories. Instead of generating complete trajectories upfront, CoRD treats each reasoning step as the minimal unit of generation, allowing teacher LRMs to collaboratively propose and integrate step proposals during their decoding. At each decoding step, we evaluate the quality of the proposed reasoning steps based on a *predictive perplexity score*, which quantifies how well the ground-truth answer is predicted given the current reasoning prefix. This scoring reflects how naturally the reasoning is expected to progress toward the correct solution, enabling early identification and adaptive selection of promising paths without requiring full trajectories. Unlike curation-based approaches, this step-wise evaluation fosters synergistic collaboration among heterogeneous teachers, while unlike MCTS, it avoids repeated rollouts and improves computational efficiency.

Although predictive perplexity offers an effective local signal of short-term consistency, it lacks awareness of long-term payoff. To address this, we integrate *beam search* into our decoding framework, which maintains multiple high-potential trajectories in parallel and preserves reasoning paths that may initially seem sub-optimal but ultimately lead to superior solutions, including strategic shifts and self-corrections often overlooked by reward-driven methods. By formulating reasoning distillation as *step-wise collaborative decoding with beam search*, we transform reasoning from one-shot selection into incremental generation.

Our evaluation on five close-ended or open-ended reasoning benchmarks compares CoRD against two multi-teacher distillation baselines (Curation and Integration) as well as state-of-the-art methods

(S1 and LIMO-v1/v2), highlighting three main contributions: (1) We propose CoRD, a novel multi-teacher reasoning distillation framework that reformulates post-hoc reasoning selection into step-wise collaborative decoding, (2) We introduce prompt-guided step segmentation, predictive perplexity scoring, and beam search as core mechanisms, each empirically outperforming alternative designs; and (3) Compared to baselines, CoRD produces higher-quality reasoning data and distills student models that approach or even surpass teachers.

2 Related work

Test-time Scaling. LLMs achieve stronger performance when their generation is guided by reasoning, such as CoT, rather than directly producing answers (Wei et al., 2022). Test-time scaling further enhances this ability by allocating additional inference-time computation, enabling models to perform Long-CoT reasoning. Techniques like multi-pass inference (He et al., 2024), which compares multiple attempts, and self-reflection (Huang et al., 2025), which iteratively revises intermediate steps, have demonstrated substantial improvements. However, these gains entail significant computational overhead (Snell et al., 2024), motivating the distillation of test-time scaling capabilities from LRMs into smaller students (Yeo et al., 2025).

Reasoning Distillation. Reasoning distillation transfers reasoning from large teacher models to lightweight students by distilling complete reasoning trajectories at the sequence-level, not by token-level logit matching (Hu et al., 2025; Kim et al., 2025). For short-CoT reasoning, PRMs ensure sequence-level quality by filtering incorrect steps (Lai et al., 2024; Wu et al., 2025b). MCTS (Yao et al., 2025) pairs this correctness-based filtering with exploration, expanding approved steps and synthesizing them into complete reasoning paths. However, Long-CoT reasoning distillation is more challenging, since PRM overlooks reasoning that could improve through revising intermediate errors, while MCTS struggles with the rapidly expanding search space. Thus, curation is widely adopted, following a generate-then-select strategy in which LRMs first produce complete reasoning trajectories and then select candidates using simple heuristics (Zhang et al., 2025; Ye et al., 2025). However, this strategy samples blindly, with no guarantee of valid reasoning or strong training signals, leading to discarded computation.

Collaborative Distillation. Distillation from multiple LLMs is a long-standing paradigm, harnessing teacher diversity to curate training data (Song et al., 2025b; Ma et al., 2025). Beyond diversity, subsequent studies explore collective synergies to produce outcomes unattainable by isolated models. They construct collective responses achieved either through collective MCTS that selects compelling reasoning steps across models (Yao et al., 2025) or through direct integration of their responses (Wang et al., 2024). In this vein, recent efforts extend these ideas by leveraging LRMs as additional sources of diversity, but mainly through simple curation (Li et al., 2025b; Ye et al., 2025).

3 Multi-Teacher Reasoning Distillation

We begin by formalizing our multi-teacher reasoning distillation setting. Let x be a reasoning problem and \mathcal{T} be a set of K large reasoning models (LRMs) acting as teachers. In the *curation*-based setting, which has been adopted in prior approaches such as S1 and LIMO, the k -th teacher in \mathcal{T} generates a complete reasoning trajectory $\tau^{(k)} = (s_1^{(k)}, \dots, s_{T-1}^{(k)}, s_T^{(k)})$ conditioned on the problem x , where s_T denotes the final answer and the preceding steps (s_1, \dots, s_{T-1}) represent a Long-CoT reasoning process¹. Then, the distillation dataset is constructed by collecting all trajectories generated by the K teachers and selecting the highest-quality one for each instance based on a quality function $Q(x, \tau)$. Formally, the final dataset over N training instances is defined as:

$$\mathcal{D}_{\text{curation}} = \{(x_i, \tau(x_i)^*)\}_{i=1}^N, \text{ where} \quad (1)$$

$$\tau(x_i)^* = \operatorname{argmax}_{\tau^{(k)} \in \{\tau^{(1)}, \dots, \tau^{(K)}\}} Q(x_i, \tau^{(k)}).$$

While this method is simple and effective for selecting high-quality trajectories from multiple teachers, it relies on post-hoc evaluation and thus cannot leverage multiple teacher LRMs to collaboratively explore and refine reasoning paths.

To overcome these limitations, we facilitate *step-wise collaboration* among teacher LRMs during trajectory construction. At step t , each teacher proposes a candidate next reasoning step $s_t^{(k)}$ conditioned on the current prefix $\tau_{<t}$, which consists of all reasoning steps selected before t . Then, a selection criterion $S(\cdot)$ evaluates each extended trajectory $\tau_{<t} \oplus s_t^{(k)}$, where \oplus denotes appending a candidate step from any k -th teacher to the current

¹Each teacher can generate multiple reasoning trajectories, but we assume one per teacher for simplicity of formulation.

reasoning prefix. As a result, the distillation dataset over N instances is defined as:

$$\mathcal{D}_{\text{step-wise}} = \{(x_i, \tau(x_i)^*)\}_{i=1}^N \text{ where} \quad (2)$$

$$\tau(x_i)^* = \left\{ (s_1^*, \dots, s_T^*) \mid \right.$$

$$\left. s_t^* = \operatorname{argmax}_{s_t \in \{s_t^{(1)}, \dots, s_t^{(K)}\}} S(\tau_{<t} \oplus s_t^{(k)}), \forall t \right\}.$$

This pipeline enables the composition of complementary reasoning steps from multiple teachers. However, it also introduces challenges such as defining reasoning steps, evaluating their quality, and efficiently managing a larger search space.

4 CoRD: Collaborative Reasoning Decoding for Reasoning Distillation

To instantiate the step-wise collaboration in Eq. (2), we conceptualize it as a *step-wise auto-regressive decoding* process where each reasoning step acts as a "token" and teacher-proposed steps form the "decoding vocabulary," enabling efficient exploration of a broader search space.

In this section, we present three core components of our approach, CoRD: (i) Defining consistent steps across diverse Long-CoT trajectories, (ii) Designing a selection criterion to accurately evaluate partial reasoning, and (iii) Capturing global deliberative processes beyond local quality.

4.1 Prompt-guided Step Segmentation

A starting point of step-wise collaborative decoding is addressing the difficulty of consistently segmenting reasoning trajectories into discrete steps, as different LRMs often produce Long-CoT processes with varying granularity, structure, and progression. A straightforward solution is the *line-break* step unit (Feng et al., 2023; Lai et al., 2024), which segments reasoning at line breaks (*e.g.*, `\n\n`) into short chunks, offering a uniform structure but little semantic coherence. Similarly, the *prefix*-based approach (Li et al., 2025c) identifies steps using explicit textual markers (*e.g.*, `wait`), adding semantic cues; however, both the frequency of such markers and the content within each step vary widely across LRMs, hindering direct comparison.

To this end, we introduce *prompt-guided step segmentation*, which inserts explicit markers into the reasoning to divide it into semantically coherent and functionally distinct steps at a consistent level, regardless of the teacher. Specifically, we embed "`<think> ### Step`" in the initial prompt, guiding LRMs to structure their reasoning into clearly separated steps during generation. This simple yet

Prompt-guided LRM Reasoning	
### Step 1. Understanding the problem	Okay, so we have four circles that are all mutually externally tangent ···
\n\n ### Step 2. Recalling Descartes' Circle Theorem	Descartes' Circle Theorem relates the curvatures of four mutually tangent circles. The formula is: \n\n
$k_4 = k_1 + k_2 + k_3 \pm 2\sqrt{k_1k_2 + k_2k_3 + k_3k_1}$	\n\n Where $k = 1/r$ for each circle ··· Hmm, but wait, if you have four circles all externally tangent, one of them might be the outer circle enclosing the other three. Wait, maybe ···

Table 1: Comparison of step segmentation. Red, Yellow, and Blue represent line-break, prefix, and prompt-guided segmentation, respectively.

effective method, shown in Table 1, ensures that each reasoning step is marked and its content logically segmented. As a result, superficial cues such as line breaks or prefix tokens (e.g., \n\n or wait) appear naturally within a single step rather than being mistaken for boundaries, enabling more faithful segmentation and reliable cross-model comparison.

4.2 Perplexity-based Step Selection

Another crucial aspect is defining the selection criterion $S(\cdot)$ in Eq. (2), which decides the most promising candidate step among those proposed by teacher LRMs. Thus, we view collaborative decoding as a *step-level* extension of the autoregressive decoding framework: At each decoding step t , each teacher generates a candidate reasoning step $s_t^{(k)}$ conditioned on the current prefix $\tau_{<t}$. These K proposals collectively form the *decoding vocabulary* $\mathcal{V}_t = \{s_t^{(1)}, s_t^{(2)}, \dots, s_t^{(K)}\}$, where the conventional notion of a token vocabulary is replaced by a set of reasoning steps proposed by multiple teachers.

For the scoring function, we introduce a separate model, referred to as the *meta-prover* (MP), which estimates the conditional probability of the ground-truth answer given a partial reasoning trajectory (See Appendix A for the prompt used to compute perplexity)². Specifically, at decoding step t , let $\tau_{<t}$ denote the reasoning prefix up to the previous step, and s_t^k a next reasoning step proposed by the k -th teacher LRM. When this step is appended, the updated reasoning state becomes $\tau_{<t} \oplus s_t^k$. Then, the meta-prover p_{meta} models the joint conditional probability of the answer tokens given this updated prefix, from which the *predictive perplexity score* used to evaluate candidate steps is derived as:

$$S(\tau_{<t} \oplus s_t^{(k)}) = \exp\left(\frac{1}{M} \log p_{meta}(A | \tau_{<t} \oplus s_t^{(k)})\right) \quad (3)$$

$$p_{meta}(A | \tau_{<t} \oplus s_t^k) = \prod_{m=1}^M p_{meta}(a_m | \tau_{<t} \oplus s_t^k, a_{<m})$$

where $A = (a_1, \dots, a_M)$ denotes the ground-truth answer represented as a sequence of tokens, yielding a bounded predictive perplexity score in the $[0, 1]$.

That is, the selected step is determined by the predictive perplexity score, where a higher value indicates that the extended reasoning trajectory better predicts the correct answer. Thus, the step with the highest score s_t^* is chosen from the entire decoding vocabulary \mathcal{V}_t at time t .

4.3 Step-wise Decoding with Beam Search

The selection in Eq. (3) unfolds auto-regressively, progressively extending the reasoning trajectory until the special `</think>` token signals its completion. When the pre-defined token budget is exhausted before this point, the sequence is terminated by appending `</think>` immediately. The teacher selected at the final decoding step generates the final answer based on the completed reasoning.

However, the greedy decoding above suffers from a fundamental limitation. By always choosing the locally optimal step, it can prematurely commit to sub-optimal paths, discarding alternatives that enable strategic shifts to emerge later in Long-CoT reasoning. On the other hand, MCTS estimates global utility by rolling out complete reasoning trajectories at each step, it becomes computationally prohibitive for Long-CoT reasoning due to the extensive search space. To address the trade-off between them, we integrate *beam search* into our decoding pipeline, which maintains the top- B most promising partial reasoning trajectories at each step instead of pursuing a single path. At decoding step t , we denote the beam from the previous step as $\mathcal{B}_{t-1} = \{\tau_{<t}^{(1)}, \tau_{<t}^{(2)}, \dots, \tau_{<t}^{(B)}\}$. The beam is then updated by extending every prefix with candidate steps from its decoding vocabulary $\mathcal{V}_t^{(b)}$, producing a total of $B \times K$ proposals at step t . From these, the top- B updated trajectories with the highest predictive perplexity scores are selected:

$$\mathcal{B}_t = \text{Top-}B(C_t) \quad \text{where} \quad (4)$$

$$C_t = \{\tau_{<t}^{(b)} \oplus s_t^{(k)} \mid \tau_{<t}^{(b)} \in \mathcal{B}_{t-1}, s_t^{(k)} \in \mathcal{V}_t^{(b)}\}.$$

Compared to greedy decoding, beam search retains alternative reasoning paths that enable strategic shifts, with more reasonable overhead than MCTS.

²We use QwQ-32B as the meta-prover, the strongest LRM in our pool: {R1-Qwen-32B, QwQ-32B, Phi4-Reasoning-Plus}; results with other meta-provers are reported in Appendix B.

4.4 Computational Complexity

We analyze the computational complexity of CoRD using Big-O notation and compare it with greedy decoding and MCTS, as well as the curation-based method, to clarify the computational overhead by its step-wise generation and beam search.

Let T denote the length of a reasoning trajectory (*i.e.*, the number of generated steps), and let M denote the meta-prover cost. For a fair comparison consistent with our experimental setup, all methods generate B reasoning trajectories in total.

CoRD. At each decoding step, CoRD generates a total of $K \times B$ proposals and scores them using the meta-prover. With cached key-value states, each expansion requires only an incremental forward pass, yielding KMB expansions per step and an overall complexity of $O(TKMB)$. The greedy decoding is a special case of CoRD with beam size = 1.

MCTS. This retains a single reasoning trajectory at each step, but it estimates rewards via full rollouts every step. As rollouts complete the remaining trajectory from $\tau_{<t}$, their expected cost decreases with depth and is approximated as $\log(T)$. Repeating this process up to B runs under the budget leads to a total complexity of $O(TK \log(TMB))$.

Curation. Curation generates full reasoning trajectories in a single pass. Each of the K teachers produces a trajectory of length T , scored once by the meta-prover after generation. This can be repeated up to B rollouts under the budget, after which the highest-scoring trajectory is selected post-hoc. This results in a total complexity of $O(TKB)$

Taken together, CoRD incurs lower complexity than MCTS. Although it is more expensive than greedy decoding or curation, we show that (i) CoRD yields higher-quality Long-CoT trajectories that cannot be obtained by simply increasing the sample budget of greedy decoding or curation, enabled by step-wise collaborative decoding, (ii) the meta-prover overhead (M) is negligible in practice. See details in Section 5.3.3 for reasoning quality and Appendix G.4 for efficiency under an identical answer-reaching sample budget, respectively.

5 Evaluation

In this section, we evaluate the quality of reasoning data generated by CoRD and the performance of a student model trained on it, demonstrating how reasoning quality influences final task outcomes.

Baselines. We compare CoRD against two baselines,

Curation and Integration, both of which leverage multiple teachers for reasoning distillation using a post-hoc approach, as detailed below.

- **Curation:** This pipeline is the standard approach used in S1 and LIMO, where each teacher LRM generates a complete trajectory, all are scored as a whole, and the highest-scoring one is selected. For fairness, we apply the same scoring in Eq. (3).

- **Integration:** This pipeline performs a post-hoc process in which an external integrator (GPT5-mini) merges the complete reasoning trajectories generated by multiple teachers into a single trajectory, selecting and combining consistent parts from each. Refer to the merging prompt in Table 17.

The key distinction of these methods, including CoRD, lies in whether multiple teachers are used independently (Curation), merged post-hoc (Integration), or collaboratively combined during step-wise collaborative decoding (CoRD).

Teacher Configuration. We consider two multi-teacher configurations: (i) *homogeneous*, where all teachers share the same architecture but differ due to sampling with different temperatures in $\{0.5, 0.6, 0.7\}$; and (ii) *heterogeneous*, where teachers vary in architecture to provide complementary reasoning. For the homogeneous setup, we fix the teacher LRM as QwQ-32B, while for the heterogeneous setup, we additionally include R1-Distil-Qwen-32B (abbreviated as R1-Qwen-32B) and Phi4-Reasoning-Plus alongside QwQ-32B. The sampling temperature is fixed at 0.6 for the three teachers.

Reasoning Data Distillation. To distill Long-CoT reasoning, we use the LIMO-v1 dataset, which contains 817 question–solution pairs curated from millions of mathematical problems via multi-stage filtering based on difficulty and reasoning depth. We then augment the reasoning traces over the dataset using two baseline pipelines³, including CoRD, and train three student models, R1-Qwen-7B/14B/32B, through supervised fine-tuning on each of the constructed datasets. All the trained students are evaluated on two widely used mathematical reasoning benchmarks, AIME24 and AIME25. Refer to Appendix C for detailed training configurations.

Hyperparameters. We set the beam size of CoRD to 4, producing four partial trajectories at each decoding step. For a fair comparison under the same compute budget, we equalize the total number of gen-

³For reasoning generation, we set the maximum output to 20,480 tokens, allocating 16,384 for <think> reasoning and 4,096 for the final answer to prevent overthinking.

Teacher Config.	Distillation Pipeline	Answer Accuracy	Predictive Perplexity
Homo.	Curation	77.4	0.664
	Integration	88.6	0.215
	CoRD	90.0	0.726
Hetero.	Curation	84.8	0.652
	Integration	91.2	0.223
	CoRD	93.1	0.774

Table 2: Quality of the generated reasoning across three distillation pipelines under two teacher configurations: Homogeneous (Homo.) and Heterogeneous (Hetero.). Best values for each setup are highlighted in bold.

erated reasoning trajectories across Curation and Integration by adjusting the number of rollouts, generating four complete trajectories per teacher.

5.1 Reasoning Quality Comparison

We evaluate the quality of the generated reasoning across three pipelines: Curation, Integration, and CoRD. A high-quality Long-CoT reasoning is expected to satisfy two criteria: (i) *answer accuracy*, where the final answer in the reasoning trajectory matches the ground-truth, ensuring task correctness, and (ii) *predictive perplexity*, where the predictive perplexity conditioned on the reasoning is high, reflecting progress consistency. Table 2 compares Long-CoT reasoning quality across three distillation pipelines under homogeneous and heterogeneous teachers. While all teachers are fixed to QwQ-32B in the homogeneous setup, additional results with alternative teacher choices are in Appendix D.

Highlight. CoRD achieves the highest answer accuracy and predictive perplexity for its generated Long-CoT reasoning, with the advantage becoming more pronounced under the heterogeneous setup, where diverse teacher signals with complementary reasoning styles interact step by step to reinforce each other, suppress unstable trajectories early, and explore alternative solution paths. This leads to a richer and more consistent reasoning dynamics than in the homogeneous setting, where teachers offer limited diversity despite temperature variations.

Detailed Analysis. The observed quality gap can be attributed to two fundamental aspects of multi-teacher distillation: (i) *complementarity exploitation*, which concerns how effectively diverse reasoning signals are combined, and (ii) *collaborative composition*, which captures how those signals interact during the reasoning process itself.

First, regarding complementarity exploitation, the strength of CoRD is evident when contrasted with Curation. The latter follows a generate-then-select

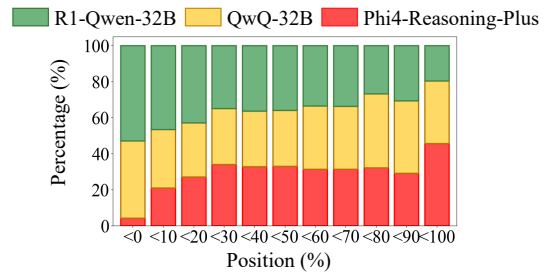


Figure 2: Teacher selection hit rates (%) in CoRD over reasoning progress where decoding steps are mapped to a 0–100% scale to align varying trajectory lengths.

strategy, where each teacher produces complete reasoning independently and complementary signals are never exchanged, leading to the lowest answer accuracy. In contrast, CoRD integrates signals via step-wise collaborative decoding, improving both metrics by reinforcing complementary reasoning

Second, in terms of collaborative composition, CoRD differs fundamentally from Integration. While this post-hoc fusion improves answer accuracy over Curation, it cannot shape the reasoning process and rather compresses it into less deliberative Short-CoT forms, leading to very low predictive perplexity. In contrast, CoRD composes reasoning incrementally, allowing complementary signals to guide each step and yielding deeper, more coherent trajectories preserving benefits of test-time scaling.

Analysis of Collaboration Dynamics. To understand the source of CoRD’s advantage, we examine teacher selection hit rates in Figure 2, which measure how often each teacher’s step candidate is selected over normalized reasoning progress. CoRD exhibits a specialized allocation pattern, where each teacher is selected for the reasoning phase that best matches its strengths rather than being uniformly sampled from a shared pool. R1-Qwen-32B and QwQ-32B dominate selection in the early phases ($\leq 40\%$), which correspond to problem formulation and constraint analysis, while Phi4-Reasoning-Plus increasingly takes over in the late phases ($\geq 80\%$), where prior steps must be synthesized into a conclusion. Such specialization is possible because each teacher conditions on the shared prefix $\tau_{<t}$ accumulated from prior steps, so the predictive perplexity scoring captures not only local step quality but also how well a candidate aligns with the current trajectory context. Prompt-guided step segmentation and beam search further reinforce these collaboration dynamics, producing more distinct and stable specialization patterns compared to their counterparts (see Appendices G.1 and G.3).

Teacher Model Performance						
Model Name	AIME24			AIME25		
R1-Qwen-32B	71.6			53.8		
QwQ-32B	77.9			66.7		
Phi4-Reasoning-Plus	78.9			67.9		
Student Model Performance: {R1-Qwen-7B/14B/32B}						
Distillation Pipeline	AIME24			AIME25		
	7B	14B	32B	7B	14B	32B
wo. Distillation	51.3	68.1	71.6	37.5	50.6	53.8
Curation-Homo	55.8	72.5	74.2	40.2	54.7	62.7
Integration-Homo	7.9	7.1	11.9	5.4	6.3	6.9
CoRD-Homo	58.5	73.7	75.8	42.9	59.3	64.4
Curation-Hetero	56.6	68.1	75.0	42.1	54.6	62.1
Integration-Hetero	8.3	7.5	12.7	3.8	4.0	9.0
CoRD-Hetero	60.8	74.8	79.6	45.6	62.3	70.2

Table 3: Distillation performance comparison across three pipelines under two teacher configurations. The upper block reports teacher performance, while the lower block shows student performance on AIME24 and AIME25 with and without reasoning distillation.

5.2 Student Performance Comparison

We evaluate how the reasoning quality in Table 2 translates into student model performance after distillation. Following recent protocols (Ye et al., 2025; Guo et al., 2025), the student’s reasoning and answers are generated with a maximum output length of 32,784 tokens and a temperature of 0.6. We report Pass@1, the proportion of test questions where the model’s first generated answer matches the correct solution, on AIME24 and AIME25. Pass@1 is computed as the average accuracy over 16 runs to ensure a stable performance estimate. Table 3 shows the Pass@1 of student models (R1-Qwen-7B/14B/32B) with different sizes, each trained using three distillation pipelines under two teacher configurations.

Result. CoRD consistently delivers the highest Pass@1 across all student model sizes and teacher settings, demonstrating substantial improvements over the two baselines. The gains are particularly pronounced under heterogeneous teachers, where CoRD effectively integrates complementary reasoning signals. Remarkably, the 32B student model distilled with CoRD even surpasses the performance of all individual teacher models on both AIME24 and AIME25, indicating that the collaborative signals distilled through step-wise reasoning go beyond simple teacher imitation. This highlights the ability of CoRD to preserve and enhance high-quality reasoning patterns during training, enabling students to approach or exceed teacher-level performance.

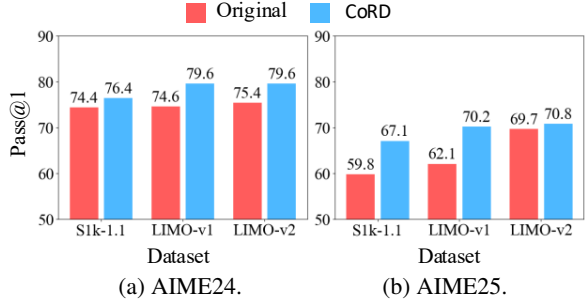


Figure 3: Performance comparison of student models trained on CoRD’s reasoning data and curated datasets from S1k-1.1 and LIMO-v1/v2, respectively.

Relation to Quality Metrics. The performance trends align closely with the reasoning quality analysis in Table 2. Predictive perplexity strongly correlates with student performance, as it captures how well the reasoning guides the model toward the correct solution. In contrast, answer accuracy, which focuses only on the final outcome, fails to translate into comparable gains, as seen in Integration, which achieves higher accuracy than Curation but yields significantly poorer distillation performance because it collapses reasoning into short-form CoT and loses valuable supervision signals. This trend is consistent across another student architecture and a stronger integrator (see Appendices E–F).

Comparison with S1 and LIMO. We compare our reasoning data with prior curation-based datasets, S1k-1.1 and LIMO-v1/v2. This comparison highlights the advantage of our collaborative decoding over static curation, showing its ability to generate higher-quality reasoning that yields stronger, more stable distillation. By applying CoRD to base datasets with varying sizes and question distributions (S1k-1.1 with 1,000 questions, LIMO-v1 with 817, and LIMO-v2 with 800), we demonstrate consistent performance gains regardless of the dataset. Figure 3 presents the Pass@1 comparison, where the same student model (R1-Qwen-32B) is trained on equal amounts of data from either the original curated reasoning or our CoRD, ensuring a fair comparison.

The student model distilled on CoRD outperforms those trained on the original datasets on both benchmarks, with particularly larger gains on AIME25, which is more challenging. These results show that while curation-based approaches rely on manual dataset design and filtering, step-wise decoding like CoRD automatically produces higher-quality reasoning data and improve distillation performance.

Segmentation Method	Reasoning Qual.		Distillation Perf.	
	Acc.	PP.	AIME24	AIME25
Line-break	88.4	0.734	76.7	67.7
Prefix	91.3	0.747	77.1	67.3
Prompt-guide	93.1	0.774	79.6	70.2

Table 4: Comparison of CoRD across three step units (Acc.=answer accuracy; PP.=predictive perplexity).

5.3 Component-wise Analysis

CoRD has three key components for effective and efficient Long-CoT reasoning synthesis: (i) prompt-guided step segmentation, (ii) perplexity-based step selection, and (iii) decoding with beam search. We evaluate each component individually to understand its contribution to the overall performance.

5.3.1 Effect of Step Segmentation

We examine how different step units in CoRD’s collaborative decoding affect reasoning quality and distillation performance under the heterogeneous teacher setup. We compare our prompt-guided step segmentation with two alternatives, *line-break* and *prefix-based* methods (see Appendix G.1 for details). Table 4 compares three step-segmentation variants for the student model R1-Qwen-32B. The prompt-guided step unit proves most effective, capturing both style consistency and semantic parity that enable multiple teacher LRMs to reason within a shared step. In contrast, the prefix-based approach aligns better with semantic boundaries but lacks style consistency, while the line-break approach maintains style consistency but fails to achieve semantic alignment, limiting collaborative synergy. These results demonstrate that well-structured step segmentation is essential for maximizing multi-teacher collaboration and producing high-quality supervision signals.

5.3.2 Effect of Step Selection Criterion

To understand the impact of selection strategies in CoRD, we compare our predictive perplexity-based selection with four alternatives: two trajectory-level approaches that select the entire reasoning post-hoc (Random Selection and Max-length Selection) and two step-wise approaches that use either a Process Reward Model (PRM) based on Qwen2.5-Math-PRM-72B or Binary Judgment from LLMs.

Table 5 summarizes the reasoning quality and distillation performance under five different selection criteria. The results show that our predictive perplexity achieves the highest scores on both reasoning quality metrics. While a high perplexity

Selection Method	Reasoning Qual.		Distillation Perf.	
	Acc.	PP.	AIME24	AIME25
Random Selection	80.4	0.494	69.0	61.9
Max-length Selection	80.0	0.502	68.8	59.0
PRMs	82.6	0.591	75.0	64.6
Binary Judgment	91.7	0.626	77.7	66.3
Predictive Perplexity	93.1	0.774	79.6	70.2

Table 5: Comparison of CoRD across five reasoning selection methods with different selection levels and criteria (Acc.=answer accuracy; PP.=predictive perplexity).

Decoding Strategy	Reasoning Qual.		Distillation Perf.	
	Acc.	PP.	AIME24	AIME25
Greedy	81.6	0.719	76.7	66.5
MCTS	89.6	0.755	75.8	66.3
Beam Search	93.1	0.774	79.6	70.2

Table 6: Comparison of CoRD across decoding strategies (Acc.=answer accuracy; PP.=predictive perplexity).

score is expected, the significantly lower values from other methods indicate their failure to anticipate and guide future reasoning steps effectively. A more direct comparison comes from the performance of student models trained on the resulting reasoning data, where the predictive perplexity-based approach consistently achieves the best results.

Alternative strategies show clear limitations. Random and Max-length Selection introduce noise and fail to ensure reasoning quality. PRM partially filters errors but often removes trajectories that could self-correct into higher-quality reasoning. Binary Judgment provides only discrete labels instead of continuous scores, producing a sparse signal that struggles to capture subtle quality differences.

5.3.3 Effect of Decoding Strategy

The final component of CoRD is the decoding strategy, which, rather than relying on local greedy decisions, aims to explore and preserve diverse reasoning paths. We compare CoRD against two decoding variants, Greedy Decoding and MCTS. For MCTS, we use the same perplexity-based scoring, while utilizing expansion and backpropagation based on upper confidence bound (Kocsis and Szepesvári, 2006). We generate four reasoning trajectories in both variants to match the computational budget.

Table 6 compares three decoding variants. The results show that beam search delivers the strongest reasoning quality and distillation performance by enabling balanced exploration and collaboration. In contrast, greedy decoding keeps a single hypothesis and enforces locally optimal choices at each step, leading to short-sighted and unstable exploration. MCTS assigns trajectory-level rewards via

Distillation Pipeline	MATH500	TaTQA	PubMedQA
wo. Distillation	92.1	87.3	86.0
Curation-Homo	93.5	80.5	86.1
Integration-Homo	74.1	73.3	84.0
CoRD-Homo	93.9	90.0	90.6
Curation-Hetero	93.4	88.2	88.4
Integration-Hetero	72.3	73.1	83.0
CoRD-Hetero	94.8	95.2	91.8

Table 7: Distillation performance comparison across three pipelines under two teacher configurations on MATH500, TaTQA, and PubMedQA.

full rollouts, which makes it less synergistic. Its search biases toward stronger teachers, even when weaker ones are better at specific steps, weakening complementarity (see Appendix G.3).

Furthermore, Appendix G.4 analyzes efficiency in terms of wall-clock time against Curation and MCTS. CoRD runs in roughly half the computation time (49.0%) of MCTS with negligible meta-prover overhead, and achieves substantially higher reasoning quality than Curation at modest additional cost, demonstrating more effective use of computation.

5.4 Generalization of CoRD

We apply CoRD to two additional arithmetic tasks and one open-ended reasoning task to evaluate generalization, as summarized in Table 7.

Arithmetic Reasoning. We test CoRD and two baselines (trained using R1-Qwen-32B in Table 3) on MATH500 (Hendrycks et al., 2021) and TaTQA (Zhu et al., 2021). Here, MATH500 shares a similar problem structure with AIME (in-domain), whereas TaTQA requires table-based reading comprehension (out-of-domain). As shown in the 2nd and 3rd columns of Table 7, CoRD outperforms other methods in Pass@1, indicating that its distilled reasoning transfers robustly beyond AIME.

Open-ended Reasoning. To assess CoRD beyond mathematical tasks, we evaluate it on PubMedQA (Jin et al., 2019), an open-domain biomedical QA benchmark with long, free-form answers. Since PubMedQA requires domain-specific, paragraph-level reasoning, we construct a new reasoning-distillation dataset of 456 samples and train a student model (R1-Qwen-32B) accordingly (see Appendix H for implementation details for open-ended tasks.). As shown in the 4th column of Table 7, CoRD achieves the highest Pass@1, demonstrating its effectiveness on open-ended, domain-specific reasoning tasks.

6 Conclusion

We presented CoRD, which redefines reasoning distillation as a dynamic, step-wise decoding process. By enabling collaborative construction of reasoning trajectories among teacher LRMs, CoRD produces richer supervision and significantly improves student performance under moderate compute budgets. These results highlight that fine-grained collaboration and progress-aware evaluation are key to efficiently scaling Long-CoT reasoning distillation.

Limitations

Our evaluation primarily focused on the monolingual AIME24 and AIME25 benchmarks, and it remains unclear whether the proposed method can generalize to multilingual settings. Recent work suggests translating English reasoning traces into other languages to enhance multilingual capabilities, given that large language models (LLMs) are predominantly trained on English corpora (Wu et al., 2025a). We will explore whether our approach can effectively enable cross-lingual transfer of high-quality reasoning in future work.

Additionally, our distillation setup employs only SFT. While our primary focus has been on extracting high-quality reasoning traces, recent studies have explored leveraging preference learning such as direct preference optimization (DPO) to better align models to bridge the disparity between LRMs and suboptimal reasoning patterns like Short-CoT (Yang et al., 2025). Extending this line of inquiry, we aim to enhance distillation performance in future work by fostering richer interplay between our high-quality reasoning and complementary preference-aligned datasets.

Ethical Considerations

Our work aims to enhance distillation performance through collaborative decoding among LRMs. All training data are generated by publicly available LRMs and do not involve human subjects or sensitive information. Therefore, no additional ethical concerns are raised during the data collection or training phase.

Scientific Artifacts

The reasoning generation in our experiments is produced using a total of 4 language models. For open-source models, we utilized publicly available checkpoints from Hugging Face, and for the proprietary model, we accessed them through paid

APIs in OpenAI. Detailed model and checkpoint information are provided in Appendix A.

Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (RS-2022-NR068758), and the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT) (No. 02-26-01-0181). This work was also supported by the National Supercomputing Center with supercomputing resources including technical support (KSC-2025-CRE-0470), and the Korea Basic Science Institute (National research Facilities and Equipment Center) grant funded by the Korea government(MSIT) (No. RS-2026-25492133).

References

- Marah Abidin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. 2025. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*.
- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.
- Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Chengbo He, Bochao Zou, Xin Li, Jiansheng Chen, Junliang Xing, and Huimin Ma. 2024. Enhancing llm reasoning with multi-path collaborative reactive and reflection agents. *arXiv preprint arXiv:2501.00430*.
- Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Zhongyuan Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, et al. 2025. Can large language models detect errors in long chain-of-thought reasoning? In *ACL*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *NeurIPS*.
- Xiao Hu, Xingyu Lu, Liyuan Mao, YiFan Zhang, Tianke Zhang, Bin Wen, Fan Yang, Tingting Gao, and Guorui Zhou. 2025. Why distillation can outperform zero-rl: The role of flexible reasoning. *arXiv preprint arXiv:2505.21067*.
- Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiabin Huang. 2025. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*.
- Minwu Kim, Anubhav Shrestha, Safal Shrestha, Aadim Nepal, and Keith Ross. 2025. Reinforcement learning vs. distillation: Understanding accuracy and capability in llm reasoning. *arXiv preprint arXiv:2505.14216*.
- Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.
- Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *ECML*.
- Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack.
- Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.
- Yang Li, Youssef Emad, Karthik Padthe, Jack Lanchantin, Weizhe Yuan, Thao Nguyen, Jason Weston, Shang-Wen Li, Dong Wang, Ilya Kulikov, et al. 2025a. Naturalthoughts: Selecting and distilling reasoning traces for general reasoning tasks. *arXiv preprint arXiv:2507.01921*.
- Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025b. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*.
- Zhiyuan Li, Yi Chang, and Yuan Wu. 2025c. Think-bench: Evaluating thinking efficiency and chain-of-thought quality of large reasoning models. *arXiv preprint arXiv:2505.22113*.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Weicheng Ma, Hefan Zhang, Ivory Yang, Shiyu Ji, Joice Chen, Farnoosh Hashemi, Shubham Mohole, Ethan Gearey, Michael Macy, Saeed Hassanpour, et al. 2025. Communication is all you need: Persuasion dataset construction via multi-llm communication. In *NAACL*.

- Sungjin Park, Xiao Liu, Yeyun Gong, and Edward Choi. 2025. Ensembling large language models with process reward-guided tree search for better complex reasoning. In *NAACL*.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Bäck. 2024. Reasoning with large language models, a survey. *CoRR*.
- Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. 2025. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Hwanjun Song, Jeonghwan Choi, and Minseok Kim. 2025a. Ext2gen: Alignment through unified extraction and generation for robust retrieval-augmented generation. In *WSDM*.
- Hwanjun Song, Taewon Yun, Yuho Lee, Jihwan Oh, Gihun Lee, Jason Cai, and Hang Su. 2025b. Learning to summarize from llm-generated feedback. In *NAACL*.
- Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.
- Linjuan Wu, Hao-Ran Wei, Baosong Yang, and Weiming Lu. 2025a. From english to second language mastery: Enhancing llms with cross-lingual continued instruction tuning. In *ACL*.
- Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2025b. Enhancing mathematical reasoning in llms by stepwise correction. In *ACL*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Wang Yang, Hongye Jin, Jingfeng Yang, Vipin Chaudhary, and Xiaotian Han. 2025. Thinking preference optimization. *arXiv preprint arXiv:2502.13173*.
- Huanjin Yao, Jiaying Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. 2025. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. In *NeurIPS*.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. In *COLM*.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. In *ICML*.
- Huifeng Yin, Yu Zhao, Minghao Wu, Xuanfan Ni, Bo Zeng, Hao Wang, Tianqi Shi, Liangying Shao, Chenyang Lyu, Longyue Wang, et al. 2025. Towards widening the distillation bottleneck for reasoning models. *CoRR*.
- Wenyuan Zhang, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. 2025. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*.
- Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *ACL*.

Model Name	Checkpoints
R1-Qwen-32B (Guo et al., 2025)	deepseek-ai/DeepSeek-R1-Distill-Qwen-32B
QwQ-32B (Yang et al., 2024)	Qwen/QwQ-32B
Phi4-Reasoning-Plus (Abdin et al., 2025)	microsoft/Phi-4-reasoning-plus
GPT5o-mini	gpt5o-mini (OpenAI)

Table 8: Checkpoints of the 4 reasoning generation models. For open-source models, we use publicly available checkpoints from Huggingface, while for proprietary model, we utilize paid API services in OpenAI.

A Reasoning Generation and Selection Details.

For reasoning generation in the main experiment, we use the LIMO-v1, LIMO-v2, and S1k-1.1 datasets, which contain 817, 800, and 1000 samples, respectively. We utilize publicly available checkpoints from Hugging Face and paid the API service, as described in Table 8. The user prompt consisted solely of the question text, without any additional context. The system prompt followed the recommended instructions in accordance with each model’s usage guidelines. The prompts used for predictive perplexity evaluation and the Integration baseline are detailed below.

A.1 Predictive perplexity

Predictive perplexity is computed as described in Section 4.2. We insert the partial reasoning and ground-truth answer into the prompt to calculate the predictive perplexity below:

Predictive Perplexity Calculation Template

$\tau_{<t} \oplus s_t^k$ </think> The final answer is {GT}.

A.2 Integration Prompt

Table 17 presents the prompt used in the Integration baseline. This prompt is designed to integrate individual reasoning outputs from multiple LRMs in a manner consistent with the Long-CoT framework. It guides the integrator to merge reasoning steps into a coherent reasoning trace that preserves the characteristics emphasized in LRMs.

B Results with Other Meta-provers

For step-level guidance via predictive perplexity, we adopt the strongest teacher model as the meta-prover. Unlike approaches that rely on a trained external reward model, this design introduces no

Meta-prover Models	Reasoning Qual.		Distillation Perf.	
	Acc.	PP.	AIME24	AIME25
QwQ-32B (Strong)	93.1	0.774	79.6	70.2
Phi-4 (Moderate)	89.2	0.749	75.9	64.4
R1-Qwen (Weak)	80.5	0.641	68.5	53.2

Table 9: Effect of meta-prover choice on reasoning quality and distillation performance. Strength is assigned based on Pass@1 performance on AIME24 and AIME25 (Acc.=answer accuracy; PP.=predictive perplexity).

Parameter	Value
Batch size	8
Epochs	5
Learning rate	5.0e-6
Max sequence length	20480
LR scheduler type	cosine

Table 10: Hyperparameters of the training configuration.

additional training or deployment dependency, as the teacher model is already available during distillation. Nevertheless, a natural question arises as to whether a stronger meta-prover is always the most appropriate source of guidance. In particular, weaker models may provide more compatible supervision, as different architectures can be specialized for different types of tasks. To examine this possibility, we evaluate alternative meta-provers with varying strengths and analyze their impact on both reasoning quality and distillation performance, as summarized in Table 9.

This results shows that weaker meta-provers reduce reasoning quality and distillation performance, which is an expected outcome in knowledge distillation. This highlight the importance of carefully selecting the meta-prover from the teacher pool, as the choice of meta-prover can affect both reasoning quality and distillation performance.

C Training Details

We fine-tune the student model using supervised fine-tuning (SFT) with DeepSpeed (Stage-3) (Rasley et al., 2020) on $8 \times$ NVIDIA H100 GPUs. Table 10 summarizes the training configurations for SFT. During training, reasoning trajectories are enclosed within <think> tags.

D Results with Other Homogeneous Teacher Model Setups

Table 11 presents the results in homogeneous (single-teacher) settings. CoRD demonstrates that even in single-teacher settings, collective step-wise decoding consistently improves overall data quality across all teacher models, surpassing Curation and

Framework	Teacher Model	Answer Accuracy	Predictive Perplexity	Distillation Performance	
				AIME24	AIME25
Before Training		N/A	N/A	71.6	53.8
Curation	QwQ-32B	77.4	0.664	74.2	62.7
	R1-Qwen-32B	49.6	0.415	62.9	47.9
	Phi4-Reasoning-Plus	67.8	0.527	71.3	60.8
Integration	QwQ-32B	88.6	0.215	11.9	6.9
	R1-Qwen-32B	70.1	0.319	8.5	5.8
	Phi4-Reasoning-Plus	64.2	0.310	7.4	5.6
CoRD	QwQ	90.0	0.726	75.8	64.4
	R1-Qwen	73.2	0.573	69.8	56.0
	Phi4-Reasoning-Plus	84.0	0.628	72.5	63.9

Table 11: Reasoning data quality and distillation performance in homogeneous settings.

Distillation Pipeline	AIME24	AIME25
wo. Distillation	46.5	31.8
Curation-Homo	48.5	33.7
Integration-Homo	1.4	2.0
CoRD-Homo	50.4	37.7
Curation-Hetero	41.3	30.8
Integration-Hetero	1.0	0.2
CoRD-Hetero	54.0	39.8

Table 12: Distillation performance comparison of R1-Llama-8B model across six frameworks.

Integrator Models	Reasoning Qual.		Distillation Perf.	
	Acc.	PP.	AIME24	AIME25
GPT-5o-mini	91.2	0.223	12.7	9.0
DeepSeek-V3.2-Exp	96.2	0.199	17.3	12.9

Table 13: Comparison of reasoning quality and distillation performance across two integrators for the Integration baseline in the heterogeneous teacher configuration.

Integration in every case. This confirms that its advantages arise from organized reasoning rather than stochastic diversity under the compute budget.

E Results with R1-Llama-8B

We conduct an additional experiment to examine whether the improvements from CoRD generalize to different LRM families. Specifically, we evaluate DeepSeek-R1-Distill-Llama-8B, whose architecture and pretraining pipeline differ from the Qwen-based teachers (QwQ and R1-Qwen) used in the main experiments. As shown in Table 12, CoRD consistently outperforms all baseline frameworks, confirming that the overall trends remain consistent and that it provides strong, stable distillation signals even for models outside the Qwen family.

F Post-hoc Integration with Stronger Integrator

For the Integration baseline, we select GPT5o-mini as the integrator due to its architectural dif-

ference from the teacher model pool, which helps avoid bias, as well as its comparable performance to the teachers, thereby preventing confounding effects from a strong integrator. Despite this choice, and despite all pipelines using the same student model and scoring procedure, the student model trained under the Integration baseline exhibits a significant performance degradation prior to training. To further investigate whether this degradation is related to integrator capacity, we replace GPT-5o-mini with a stronger integrator, DeepSeek-V3.2-Exp, within the heterogeneous teacher integration setting.

Table 13 summarizes the performance of the Integration baseline across different integrators in the heterogeneous teacher configuration. Although a stronger integrator yields modest improvements, it still fails to reconstruct coherent Long-CoT structures, suggesting that the limitation does not primarily stem from integrator weakness or implementation issues, but rather reflects a fundamental challenge in post-hoc integration for current LLMs. In particular, processing extremely long contexts remains difficult due to lost-in-the-middle (Liu et al., 2024) and needle-in-a-haystack (Kuratov et al., 2024) effects. In the Integration baseline, the integrator must aggregate nearly 30K tokens of teacher Long-CoT reasoning into a coherent trajectory exceeding 4K tokens, which often leads to a collapse into short and shallow Short-CoT outputs. Post-hoc integration at this scale remains unreliable with existing methods, motivating the use of CoRD’s dynamic, step-wise synthesis.

G Additional Experiment Details

G.1 Analysis of Reasoning Dynamics Across Step Segmentations

We analyze how different step segmentation schemes affect reasoning structure and multi-

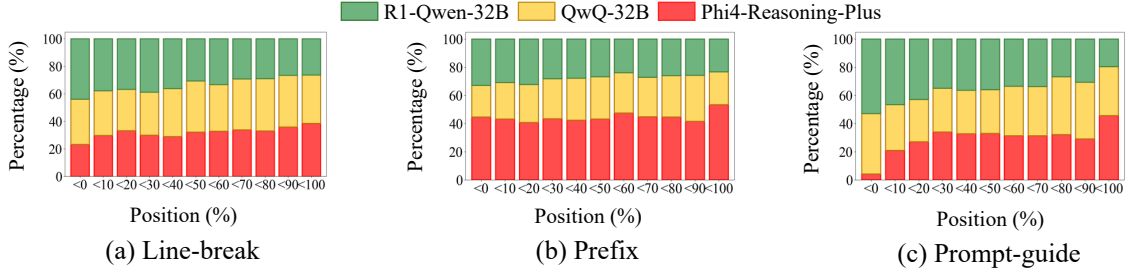


Figure 4: Hit rates of three LLMs during expansion across three step units based on step locations that reflect their relative position ratios within the entire reasoning.

teacher collaboration.

Step unit Configuration. For the step unit selection, the line-break unit simply matches the word `\n\n` as a boundary, whereas the prefix-based unit requires matching prefixes corresponding to reasoning patterns in Long-CoT reasoning. Following (Ye et al., 2025) and (Qu et al., 2025), we selected appropriate prefix terms as follows:

- Self-Verification: "let me check", "let me verify", "double-check", "going back to", "wait"
- Multi-method Validation: "alternatively", "another way", "let's try a different approach", "using another method", "we can also verify"
- Self-Correction: "this is wrong", "the mistake was", "that's impossible", "this contradicts", "the error is"

Comparison of Collaboration Dynamics. We further analyze collaboration dynamics across different step units in Figure 4, which reports each LLM's hit rate during reasoning expansion. We note that early reasoning stages typically involve problem formulation and constraint analysis (He et al., 2025). Prompt-guided step units align with these semantic phases, enabling heterogeneous LLMs to collaborate at stages best suited to their strengths. In this setting, QwQ and R1-Qwen dominate the early steps, while Phi4-Reasoning-Plus contributes more in later stages that require comprehending prior steps for conclusion. In contrast, prefix step units are dominated by a few models, with R1-Qwen selected only about 20–25%, and line-break ones exhibits a similar trend. While line-break step units encourage some stylistic sharing across models, they remain limited in fostering genuine semantic collaboration.

G.2 Binary Judgement Prompt Details

For binary judgment, we adopt the meta-prover prompt from Qu et al. (2025), where the judge

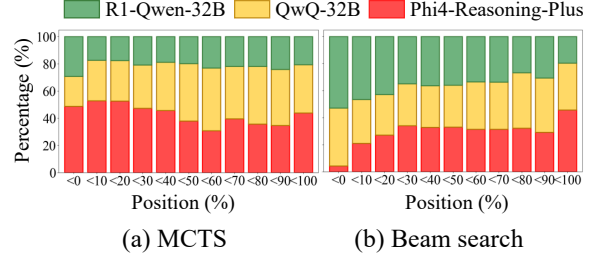


Figure 5: Comparison of hit rates during expansion between beam search and MCTS. The step locations reflect their relative position ratios within the entire reasoning.

completes the current prefix into a final answer and assigns a binary correctness score; results are averaged over 10 independent runs to reduce variance. During rollout, we append the phrase "The final answer is" at the end of the reasoning process to encourage the model to quickly and explicitly produce the final answer without additional unnecessary reasoning steps. The prompt is shown below:

Binary Judgement Prompt

$\tau_{<t} \oplus s_t^k$ Time is up. Given the time I've spent and the approaches I've tried, I should stop thinking and formulate a final answer based on what I already have.</think> The final answer is:

G.3 Analysis of Reasoning Dynamics across Decoding Strategies

We analyze the resulting collaboration dynamics among multiple reasoning trajectories induced by this strategy. Figure 5 compares the hit-rate distributions of different teachers across reasoning positions under MCTS and beam search, revealing how the decoding strategy alters collaborative dynamics. In MCTS, trajectory-level rewards cause the search to converge toward globally stronger teachers, reducing exploration of weaker yet occasionally effective ones. Consequently, complementary reasoning diminishes. In contrast, beam search maintains a

more balanced mixture of teacher contributions throughout, preserving complementary reasoning behaviors. Interestingly, during the early reasoning steps, beam search leverages the local strengths of weaker teachers such as R1-Qwen-32B, which often provide useful intermediate reasoning cues before stronger teachers dominate in later stages. Overall, these results show that MCTS biases collaboration toward high-performing teachers, whereas beam search sustains broader cooperation across reasoning steps.

G.4 Computational Efficiency Analysis

For a more comprehensive assessment, we analyze computational efficiency to characterize how different distillation pipelines trade off compute against reasoning quality and distillation performance.

G.4.1 Wall-clock Time Analysis

We evaluate the empirical computational efficiency of three distillation pipelines, *Curat ion*, *MCTS*, and *CoRD*. As described in Section 4.4, these pipelines primarily differ in how they allocate computation between trajectory generation and verification. *Curat ion* generates complete trajectories in a single pass and applies only lightweight post-hoc scoring, yielding a moderate generation cost with negligible verification overhead. In contrast, *MCTS* repeatedly performs full rollouts and evaluates candidate trajectories, which substantially increases both generation and verification costs. *CoRD* generates a comparable number of trajectories to *Curat ion* but employs step-wise decoding with cached key-value states and lightweight meta-prover scoring for a small ground-truth answer token, thereby maintaining moderate costs for both generation and verification.

Table 14 shows the empirical wall-clock computational cost per question under the heterogeneous teacher configuration, measured on NVIDIA H200×4 GPUs. Relative to *Curat ion*, *CoRD* adds only modest computational cost, as it avoids full rollouts by advancing at the step-level and invokes the meta-prover far less frequently than *MCTS*. While *CoRD* is slightly more expensive than *Curat ion*, the additional cost incurred by the meta-prover is small, and the substantial improvement in reasoning quality makes the extra cost reasonable.

G.4.2 Equal Computation Budget Analysis

We further evaluate whether increasing the compute budget allows *Curat ion* to match the reason-

Distillation Pipeline	Step Generation (A)	Meta-prover Evaluation (B)	Total Generation (A+B)
<i>Curat ion</i>	167.1	1.2	168.3
<i>MCTS</i>	567.7	21.5	589.2
<i>CoRD</i>	277.3	11.4	288.7

Table 14: Computational efficiency comparison of three distillation pipelines. Time reported as the average wall-clock time (in seconds) per question measured on NVIDIA H200×4 GPUs. Breakdown shown for step generation (A) and meta-prover evaluation (B).

Decoding Strategy	Reasoning Qual.		Distillation Perf.		Time Sec
	Acc.	PP.	AIME24	AIME25	
<i>Curat ion</i>	84.8	0.652	75.0	62.1	168.3
<i>Curat ion</i> _{x2}	90.3	0.712	74.6	63.8	336.6
<i>CoRD</i>	93.1	0.774	79.6	70.2	288.7

Table 15: Reasoning quality and distillation performance comparison across four methods (Acc.=answer accuracy; PP.=predictive perplexity). *Curat ion* ×2 denotes a variant of *Curat ion* whose computation budget is increased to match that of *CoRD*.

ing quality of *CoRD*. We increase the computation budget for *Curat ion* by doubling the number of completions from four to eight, thereby doubling its total generation cost from 168.3s to match that of *CoRD* (288.7s), as shown in Table 14. Thus, in this increased setup (*Curat ion* ×2), the best reasoning trajectory is selected among the eight candidates based on predictive perplexity and then use it to train the student model (R1-Qwen-32B).

Table 15 shows the reasoning quality and distillation performance comparison under the same heterogeneous teacher configuration. *CoRD* achieves the best balance between efficiency and performance by allocating compute within a single run, effectively balancing exploration and exploitation. However, even when we match the compute of *Curat ion* to that of *CoRD*, the predictive perplexity of *Curat ion* ×2 remains below that of *MCTS* and *CoRD*, and the corresponding student performance does not bring improvement. This indicates that post-hoc pipelines cannot efficiently yield higher quality reasoning even with increased computation, rather leading to many discarded trajectories and substantial computational waste.

H Additional Experimental Details for PubMedQA

Unlike the mathematical domain in LIMO-v1, PubMedQA requires domain-specific, paragraph-level reasoning grounded in scientific evidence and long, free-form conclusions. Because this task demands

qualitatively different reasoning capabilities, we construct a new distillation dataset tailored to PubMedQA and train a student model. We conduct the same comparison across Curation, Integration, and CoRD under an evaluation setup designed for open-ended answers.

Dataset. To match LIMO-v1’s question configuration for effective Long-CoT distillation, we follow LIMO-v1 and retain only difficult and complex questions, where difficulty is operationalized as a low success rate and complexity is proxied by reasoning length. Specifically, for each question we sample three complete reasoning trajectories at temperature 0.6 from either Llama-3.1-8B-Instruct or Qwen2.5-7B-Instruct, without explicitly constraining the generation length. We keep questions where all three samples are incorrect and the reasoning length exceeds 1K tokens (counted using R1-Qwen-32B) for complexity. This filtering yields 456 questions from the initial 213.3K instances.

Reasoning Data Distillation. We distill reasoning data using two post-hoc baselines (Curation and Integration) and our CoRD. For a controlled comparison, we keep the distillation procedure and all hyperparameters identical across methods (*e.g.*, the teacher model pool, sampling settings, and student training configurations), and vary only the generation and meta-prover prompts. We provide a golden grounded paragraph in the reasoning generation prompt to factor out retrieval ability and isolate the quality of the reasoning itself. In the meta-prover prompt, we include the dataset’s reference long-form answer as the target for computing predictive perplexity. We train R1-Qwen-32B on each constructed dataset via supervised fine-tuning.

Evaluation. We use an LLM-as-a-judge with Qwen3-32B to assess answer accuracy and student performance, since exact matching cannot capture diverse valid linguistic formulations in open-ended tasks. We adopt the LLM-as-a-judge prompt from prior work (Song et al., 2025a), which assigns a binary label for response appropriateness. The prompt is shown below:

Selection Method	Reasoning Qual.		Distillation Perf. PubMedQA
	Acc.	PP.	
wo. Distillation	N/A	N/A	86.0
Curation-Homo	62.6	0.180	86.1
Integration-Homo	65.4	0.216	84.0
CoRD-Homo	70.3	0.284	90.6
Curation-Hetero	71.4	0.243	88.4
Integration-Hetero	65.6	0.215	83.0
CoRD-Hetero	75.8	0.339	91.8

Table 16: Quality of the generated reasoning and distillation performance comparison across three distillation pipelines under two teacher configurations.

Evaluation Prompt for PubmedQA

Your task is to evaluate the correctness of the predicted answer based on the true answer.

Instructions:

1. Read the QUERY and then compare the ANSWER and the Predicted ANSWER.
2. Check if the Predicted Answer includes the core content of the True Answer (True/False in text).
3. If the Predicted Answer is correct, return "True". If it is incorrect, return "False".

QUERY: {Question}

TRUE ANSWER: {Reference Answer}

Predicted ANSWER: {Model Answer}

Output Format: {"correctness": "True or False"}

Output (Only JSON):

Reasoning Quality Comparison. Table 16 presents reasoning quality and distillation performance (R1-Qwen-32B) across three distillation pipelines under two teacher configurations. CoRD consistently produces higher-quality reasoning traces and achieves stronger distillation performance than the baselines, and the same relationship between reasoning quality and performance observed in math domains also holds for open-domain task.

Instruction

You are tasked with analyzing multiple reasoning solutions and integrating them into a single, structured JSON output.

1. Integrate All Reasoning

- The reasoning steps are provided inside XML tags such as:

```
<reasoning_step_1> ... </reasoning_step_1>
```

```
<reasoning_step_2> ... </reasoning_step_2>
```

- Merge the content inside all these XML tags into one unified reasoning flow.

- Combine them carefully while maintaining logical flow and context.

2. Assign IDs

- Each sub-thinking process should have its own unique ID.

- Use a hierarchy such as:

"integrated_step1", "integrated_step2" for overall stages of integrated reasoning.

"answer_part" for the final answer section and use `\boxed{ }` format for the final answer.

3. Categorize Reasoning Patterns

Categorize the reasoning according to its type to ensure effective integration:

- Progressive Reasoning: Logical, forward-moving step-by-step problem solving.

Indicators: "Let's solve", "First", "Next", "Then", "Therefore", "We need to", "Given that".

- Verification: Returning to check previous steps for accuracy.

Indicators: "Wait", "Let me check", "Let me verify", "Double-check", "Going back to".

- Multi-method Validation: Using different methods or perspectives to confirm a conclusion.

Indicators: "Alternatively", "Another way", "Let's try a different approach", "Using another method".

- Error Correction Pattern: Identifying and fixing mistakes in reasoning.

Indicators: "This is wrong", "The mistake was", "This can't be right", "The error is", "This contradicts".

4. Return Your Integration in JSON Format

Provide your integrated reasoning in the following JSON structure:

```
{
  "integrated_step1": {"content": "### Step 1. <integrated reasoning text>", "category": "<reasoning pattern>"},
  "integrated_step2": {"content": "### Step 2. <integrated reasoning text>", "category": "<reasoning pattern>"},
  ...
  "integrated_stepN": {"content": "### Step N. <integrated reasoning text>", "category": "<reasoning pattern>"},
  "answer_part": ["<final answer in boxed format>"]
}
```

Target Input Example

Question: {Question}

Reasoning Steps: {Reasoning Steps}

Table 17: Prompt used to integrate reasoning across LRMs.