Collaborative Reasoning Distillation via Cross-Feedback and Coherent Curation

Taehoon Kim **Seunggeun Cho** **Dongsu Han**
KAIST
{taehoon0813, sgn.cho, dhan.ee}@kaist.ac.kr

Abstract

Reasoning capabilities are critical for advancing Large Language Models, yet current approaches either require massive computational budgets or struggle to effectively distill reasoning to smaller models. Standard distillation methods rely on outcome-based rewards, failing to distinguish between sound reasoning and lucky guesses. We propose **Collaborative Reasoning Distillation (CRD)**, a framework that enhances reasoning in compact models through three innovations: (1) interactive cross-feedback where teachers iteratively critique each other's reasoning, (2) fine-grained step-wise quality assessment capturing logical validity independent of final answers, and (3) coherence-aware step stitching that synthesizes complementary strengths. Students are trained via Reasoning Quality Optimization (RQO) with budget constraints. Our model, CRD-4B, achieves 97.3% on MATH-500 and 70.3% on AIME'25, surpassing baselines while using only 50K training examples, up to 12 times smaller than the datasets of comparable models.

1 Introduction

Large Language Models (LLMs) have exhibited impressive reasoning capabilities [Wei et al., 2023]. However, advancing reasoning abilities requires massive computational investment. Cutting-edge models demand thousands of GPUs [Jiang et al., 2024], pushing estimated hardware costs into the $800 million range [Cottier et al., 2025]. Although training smaller models from scratch appears cost-effective, the resulting systems almost always trail far behind in reasoning quality despite significant resource outlays [Fu et al., 2023, Magister et al., 2023].

Knowledge Distillation (KD) [Hinton et al., 2015] offers a practical framework for transferring capabilities from large models to compact students. While KD has been widely applied to vision and general NLP tasks, transferring complex reasoning capabilities through distillation remains challenging [Li et al., 2022, Tian et al., 2024]. The student's reasoning ability is inherently upper-bounded by the teacher, and any limitations or reasoning flaws in the teacher are inevitably inherited by the student, creating a ceiling effect that constrains achievable performance.

Fortunately, the growing ecosystem of diverse LLMs creates an opportunity. Different models exhibit distinct strengths across reasoning domains. Some excel at mathematical deduction, others at symbolic manipulation, logical analysis, or domain-specific reasoning. While recent work has explored leveraging multiple teachers for distillation [Tian et al., 2024], existing approaches rely on simple aggregation or ensemble voting, without mechanisms to identify and combine high-quality reasoning steps across teachers.

The core challenge is that standard distillation methods derive learning signals from binary correctness or format compliance, ignoring the logical validity of intermediate steps [Stanton et al., 2021]. This allows reasoning shortcuts and logical hallucinations to persist as long as they coincidentally yield correct answers. While Chain-of-Thought (CoT) prompting [Wei et al., 2023] provides explicit reasoning paths useful for distillation [Li et al., 2022], single-teacher CoT suffers from limited

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

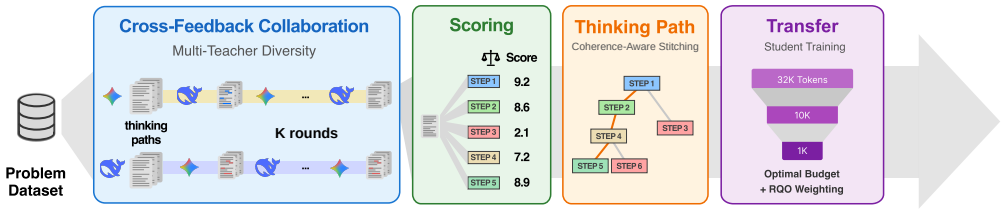


Figure 1: Overview of the CRD Framework. Multiple teachers iteratively critique and refine each other's reasoning through **Cross-Feedback**. An **LLM-as-a-Judge** then assigns step-wise quality scores, and the **Thinking Path** algorithm stitches high-quality steps across teachers while pruning flawed steps. The curated dataset is used to train students via **RQO** with Optimal Budget curriculum.

diversity, bias propagation, and inconsistent quality. Recent approaches like Group Relative Policy Optimization (GRPO) [Shao et al., 2024] have made progress in aligning model outputs. Advanced reasoning models such as DeepSeek-R1 [Guo et al., 2025] demonstrate that extended test-time computation enhances reasoning capabilities. However, longer reasoning chains substantially increase inference latency and computational cost, limiting deployment in latency-sensitive applications.

To address this limitation, we propose a complementary approach. Rather than requiring extensive test-time computation, we distill insights from teacher reasoning by assessing the logical quality of intermediate steps and synthesizing high-quality reasoning patterns across multiple teachers. This allows student models to achieve comparable reasoning quality with far fewer final training trajectories and shorter reasoning at inference.

To achieve this vision, we introduce **Collaborative Reasoning Distillation (CRD)**, a framework combining multiple teachers with complementary strengths, assessing step-level reasoning quality, and synthesizing high-quality reasoning patterns. Our key contributions are:

- **Interactive Multi-Teacher Cross-Feedback**: An iterative mechanism where diverse teachers critique and refine each other's reasoning, producing a richer candidate pool with corrected errors and complementary solution strategies.
- **Fine-grained Step-wise Quality Assessment**: A dual-judge system (JudgeLM and xVerify) that evaluates the logical validity of each reasoning step independent of the final answer. Unlike binary outcome-based rewards, this system assigns granular quality scores to each step, enabling the identification of high-quality reasoning patterns even within incorrect solutions.
- **Coherence-Aware Step-Level Stitching**: A Viterbi-style algorithm that constructs hybrid reasoning chains by selecting high-quality steps across multiple teachers, enabling solutions no single teacher could produce alone.

Together, these three components produce a curated training dataset rich in diverse, high-quality reasoning examples with explicit step-level quality signals. We train students on this dataset using Reasoning Quality Optimization (RQO) with curriculum-based budget constraints to achieve both reasoning quality and inference efficiency.

CRD substantially improves reasoning across mathematical, scientific, and programming benchmarks. On the DeepSeek-R1-Distill-Qwen-1.5B student, CRD-1.5B achieves +7.2% improvements on MATH-500, +6.8% on GPQA Diamond, and +10.3% on LiveCodeBench. On Qwen3 students, CRD-1.7B gains +2.0% on AIME'24 and +5.4% on AIME'25 over same-parameter baselines. Crucially, CRD achieves these results using only 50K training examples, demonstrating that high-quality step-level curation can make reasoning distillation substantially more data-efficient.

2 Methodology

Current reasoning distillation methods primarily rely on imitation learning from a single teacher or outcome-based reinforcement learning (e.g., DeepSeek-R1, GRPO). While effective for final answer accuracy, these approaches treat reasoning as a monolithic sequence, failing to capture the granular quality of intermediate steps.

We introduce **Collaborative Reasoning Distillation (CRD)**, a framework that shifts from imitation to collaborative synthesis. As illustrated in Figure 1, CRD transforms multiple teacher outputs

into a curated training dataset through three integrated components. First, diverse teachers generate initial reasoning chains through iterative cross-feedback. Rather than treating each teacher independently, teachers critique and refine each other's outputs, producing high-quality candidates that reflect complementary strengths. Second, we assess the logical validity of each step using a multi-judge system. Fine-grained quality scores capture reasoning quality beyond binary correctness, identifying high-quality patterns even in failed attempts. Third, we construct hybrid reasoning chains via coherence-aware step stitching, combining high-quality steps from different teachers while maintaining logical flow. Finally, we train students via Reasoning Quality Optimization (RQO) with curriculum-based budget constraints, enabling efficient reasoning within token limits. In our experiments, we refer to these components as Cross-Feedback, Judging, Thinking Path, and Optimal Budget.

2.1 Interactive Multi-Teacher Cross-Feedback

Recent studies reveal that LLMs exhibit complementary error profiles. Models like DeepSeek-R1 are prone to computational errors and statement skipping, omitting intermediate steps [Abdollahi et al., 2025], while models like Gemini exhibit shallow reasoning with prediction bias [Gao et al., 2025]. A student trained on a single teacher is often biased toward that teacher's error profile, motivating approaches that leverage these asymmetric weaknesses for mutual correction [Du et al., 2023].

We introduce a cross-feedback loop where N distinct teachers (we use $N = 2$, DeepSeek-R1 and Gemini 2.5 Flash Thinking) engage in iterative critique and refinement. Rather than simple ensemble voting, each teacher reviews peer outputs, identifies logical gaps and errors, and generates improved reasoning by synthesizing complementary insights. This interactive process produces reasoning chains of higher logical depth and diversity than any single teacher could generate alone.

For the initial round ($k = 0$), each teacher T_i independently generates a baseline reasoning chain $R_i^{(0)}$ for the given problem x. In subsequent refinement rounds ($k \in \{1, 2, \ldots, K\}$), each teacher T_i receives a structured input consisting of the original problem x and the reasoning chains from all other teachers in the previous round $\{R_j^{(k-1)} \mid j \neq i\}$. A carefully designed review prompt instructs the teacher to examine these peer chains, identify logical gaps and errors, suggest corrections, and synthesize insights from multiple perspectives (see Appendix A for the complete prompt). The teacher then generates an improved reasoning chain $R_i^{(k)}$ that incorporates this feedback.

After K rounds of cross-feedback refinement, we collect the complete set of reasoning chains including both initial and refined outputs from all teachers. Our experiments show that performance converges at $K = 2$ to 3 rounds, with additional iterations providing diminishing returns (Appendix G).

Cross-feedback substantially improves dataset quality. Comparing initial and refined outputs, mathematical reasoning accuracy increased from 64.1% to 83.5%, and code generation success rate improved from 52.3% to 73.9%. Our ablation study shows this translates to +3.4%–7.1% student performance improvements across benchmarks (Section 3.3).

2.2 Step-wise Reasoning Quality Assessment

Current models like DeepSeek-R1 [Guo et al., 2025] primarily rely on outcome-based rewards, where intermediate steps receive identical signals regardless of logical validity. This sparse reward structure fails to distinguish between efficient reasoning and verbose or lucky guesses, treating all correct-answer paths equivalently.

To address this limitation, we introduce a fine-grained step-wise quality assessment system that directly evaluates the logical validity of each intermediate step, independent of the final answer. Instead of deriving rewards from format compliance or outcome correctness, we assign continuous scalar scores to every step, capturing logical validity and reasoning efficiency [Lightman et al., 2023, Li and Li, 2025]. This enables students to learn the distinction between high-quality reasoning paths and low-quality paths that coincidentally reach the correct answer, and to internalize the nuanced differences among multiple correct solutions.

We employ two specialized judge models to evaluate each step: **JudgeLM** [Zhu et al., 2025], which achieves 16× speedup over GPT-4o with comparable accuracy, and **xVerify** [Chen et al., 2025], a lightweight 0.5B model with exceptional efficiency.

To capture algorithmic uncertainty, each judge evaluates every step twice, yielding a set of four independent scores. To ensure robustness against stochastic outliers, we apply a **25% trimmed mean** aggregation strategy: the minimum and maximum scores are discarded, and the remaining two intermediate scores are averaged to produce the final step quality score v_t. In our ablations, using both judges provides larger gains than using either judge alone, suggesting that JudgeLM and xVerify contribute complementary quality signals rather than redundant scores (Appendix H).

After step-wise scoring, we perform domain-specific verification combining LLM evaluation with specialized tools. For code, we execute solutions and penalize steps after failures. For math, we compare against ground truth and penalize steps after divergence points. For logic, we verify inference rule correctness.

2.3 Coherence-Aware Step Stitching

The third component of CRD is **Coherence-Aware Step Stitching**, which we call the *Thinking Path* algorithm. Standard distillation selects the best *complete* chain from a single teacher, inheriting that teacher's limitations even when other teachers excel at specific steps. In contrast, CRD enables step-level synthesis: combining Teacher 1's novel problem setup with Teacher 2's rigorous derivation with Teacher 1's verification, even if neither teacher alone produces a complete correct solution.

This capability is possible because we have both fine-grained quality scores (Section 2.2) and multiple diverse reasoning chains (Section 2.1). The challenge is selecting steps that maintain logical coherence. Simply picking the highest-scoring step at each position creates Frankenstein chains where steps do not flow naturally into each other.

We formulate this as a coherence-aware path-finding problem on a directed acyclic graph (DAG) [Yao et al., 2023], where nodes represent reasoning steps $s_t^{(i)}$ from teacher T_i at position t. We define the optimal hybrid path $\hat{R}$ as the sequence that maximizes both step-wise quality and semantic coherence between consecutive steps.

To handle granularity differences across teachers, we align steps using embedding similarity with lookahead merging: if consecutive steps from one teacher better match a single step from another, they are merged into one node. When similarity falls below a threshold, steps remain as parallel branches rather than forced alignments.

Let $S_t = \{s_t^{(1)}, \ldots, s_t^{(N)}\}$ be the set of candidate steps at position t. The score for transitioning from step $u \in S_{t-1}$ to step $v \in S_t$ balances two objectives:

$$\text{Score}(u \to v) = \lambda \cdot \underbrace{\mathcal{V}(v)}_{\text{Quality}} + (1-\lambda) \cdot \underbrace{\log(\epsilon + C(u,v))}_{\text{Coherence}} \tag{1}$$

where $C(u,v)$ is the transition-coherence score between consecutive steps, computed from the cosine similarity of their embeddings and rescaled to $[0,1]$, $\mathcal{V}(v)$ is the normalized step-quality score, and ϵ is a small constant for numerical stability. We use $\lambda = 0.7$ based on the sensitivity analysis in Figure 4.

We find the globally optimal path using a Viterbi-style dynamic programming algorithm:

$$\hat{R} = \arg\max_{s_1,\ldots,s_L} \sum_{t=1}^{L} \text{Score}(s_{t-1} \to s_t) \tag{2}$$

This enables *emergent solutions* that transcend individual teacher limitations. A problem might require Teacher 1's creative insight for setup, Teacher 2's careful arithmetic for calculation, and Teacher 1's verification logic for confirmation. By combining these complementary strengths at the step level, the resulting trajectory represents the collective best reasoning that no single teacher could produce in isolation. The complete algorithm with pseudocode is provided in Appendix C. Among 2,694 tasks with incorrect pre-stitch trajectories, 528 (19.6%) become correct after stitching (Appendix I).

2.4 Reasoning Transfer via RQO

Once the reasoning trajectories are synthesized via step-level stitching and assigned fine-grained quality scores, we transfer these capabilities to the student model π_θ using a reinforcement learning approach that directly leverages the step-wise quality signals.

Reasoning Quality Optimization (RQO). Standard preference-based methods like DPO treat all preference pairs equally, ignoring the magnitude of quality differences. In contrast, RQO directly utilizes the fine-grained quality scores from Section 2.2 to weight learning signals. For each reasoning chain, we aggregate the step-level quality scores to obtain a trajectory-level quality signal $v_R = \sum_t v_t$, where v_t is the quality score of step t. For chain pairs (R_{win}, v_{win}) and (R_{lose}, v_{lose}) where $v_{win} > v_{lose}$, we compute the quality margin $\Delta v = v_{win} - v_{lose}$ and apply exponential weighting:

$$w(\Delta v) = \frac{e^{\beta \Delta v} - 1}{Z} \tag{3}$$

where β controls the sensitivity to quality differences and Z is a normalization constant. The loss function becomes:

$$\mathcal{L}_{\text{RQO}} = -\mathbb{E}\left[w(\Delta v) \cdot \log \sigma\left(\pi_\theta(R_{win}|q) - \pi_\theta(R_{lose}|q)\right)\right] \tag{4}$$

This formulation ensures that chains with larger quality margins receive stronger learning signals, enabling the student to learn not just from correct vs. incorrect distinctions, but from the nuanced quality differences captured by step-level assessment.

Dynamic Length Regularization. Training with strict token constraints from the start fails due to the learnability gap [Ding et al., 2025]: students cannot learn complex reasoning patterns when forced to be concise prematurely. Following the Long-to-Short paradigm [Wu et al., 2025, Hammoud et al., 2025], we implement a curriculum-based length penalty that progressively constrains token budgets (32K→10K→1K). The effective reward includes an exponential penalty for exceeding the current budget B:

$$R_{\text{budget}}(l) = R_{\text{base}} \cdot \exp(-\alpha \max(0, l - B)) \tag{5}$$

where l is the response length and α controls penalty severity. This soft constraint [Li et al., 2025a] enables the model to first learn correct reasoning at longer lengths, then gradually compress into efficient solutions without performance degradation.

3 Evaluation

Training Data. Our training dataset comprises 50K reasoning instances from four diverse sources: DeepScaleR [Luo et al., 2025] for mathematical reasoning, Eurus-2-RL [Cui et al., 2025] for code generation, Reasoning Gym [Stojanovski et al., 2025] for logical reasoning, and SCP-116K [Lu et al., 2025] for STEM problems (Appendix D).

Recent findings show that reasoning distillation exhibits a performance valley between 1K–30K examples [He et al., 2025]. Our 50K dataset size avoids this valley while remaining significantly smaller than prior methods (220K–800K examples), enabled by multi-stage quality filtering that ensures high-quality reasoning across the dataset.

Contamination Analysis. We systematically verified training-evaluation separation. For DeepScaleR, we identified 11 overlapping problems with MATH-500 (2.2%), which were excluded during our 50% subsampling; AIME'24 and AIME'25 showed zero overlap. For Eurus-2-RL (455K math, 25K code problems), we sampled 15K code problems for training and found no contamination across all code benchmarks. LiveCodeBench v5 problems (August 2024–February 2025) post-date all code training sources: APPS (2021), CodeContests (2022), and TACO (2023).

Benchmarks. We evaluate on mathematics (MATH-500 [Hendrycks et al., 2021], AIME'24 [Mathematical Association of America, 2024], AIME'25 [Mathematical Association of America, 2025]), code (LiveCodeBench [Jain et al., 2024]), and graduate-level STEM reasoning (GPQA-Diamond [Rein et al., 2023]). Detailed descriptions are in Appendix E.

Models. We use DeepSeek-R1 and Gemini 2.5 Flash Thinking as teachers, running $K = 2$ cross-feedback rounds to generate refined reasoning chains. For students, we evaluate DeepSeek-R1-Distill-Qwen (1.5B) and Qwen3 (1.7B, 4B, 30B-A3B MoE) to demonstrate generalization across dense and mixture-of-experts architectures.

Table 1: Performance comparison across benchmarks (Pass@1). We report accuracy (%) on mathematical, coding, and general reasoning benchmarks. The best results are highlighted in **bold**. Gray rows indicate DeepSeek-R1 and distilled models included as reference points for parameter efficiency comparison.

	Math			Code	Graduate-Level STEM
Model	**MATH-500**	**AIME'24**	**AIME'25**	**LCB**	**GPQA-Diamond**
Tiny Scale (<2B)					
DeepSeek-R1-Distill-7B	92.8	55.5	–	37.6	49.1
Gemini-Flash-Distill-1.5B	85.3	32.5	24.6	17.2	31.6
DeepSeek-R1-Distill-1.5B	83.9	28.9	22.0	16.9	32.9
ProRL-1.5B†	**91.9**	**48.1**	33.3	23.8	**41.7**
CRD-1.5B†	91.1	46.9	**35.1**	**27.2**	39.7
DeepSeek-R1-Distill-7B	92.8	55.5	–	37.6	49.1
Qwen3-1.7B†	93.4	48.3	36.8	33.2	40.1
CRD-1.7B†	**94.5**	**50.3**	**42.2**	**37.3**	**42.9**
Small Scale (≤4B)					
DeepSeek-R1-Distill-32B	94.3	72.6	60.0	57.2	62.1
Qwen3-4B†	97.0	73.8	65.6	54.2	55.9
CRD-4B†	**97.3**	**76.2**	**70.3**	**57.4**	**58.7**
Large Scale (≥30B)					
EXAONE Deep 32B	95.7	72.1	65.8	59.5	66.1
Hermes 4 70B	95.5	73.5	67.5	50.5	66.1
DeepSeek-R1 (671B, A37B)	97.3	79.8	70.0	63.5	71.5
Qwen3-30B-A3B†	98.0	80.4	70.9	62.6	65.8
CRD-30B-A3B†	**98.3**	**82.5**	**73.2**	**65.9**	**67.2**

† Models marked with † share the same baseline model; our method generally outperforms them across benchmarks.

3.1 Performance Across Model Sizes

A key question in reasoning distillation is whether small models can achieve strong performance through higher-quality supervision rather than relying solely on parameter scaling. Table 1 demonstrates that CRD answers this affirmatively through step-wise quality assessment and coherence-aware synthesis, achieving competitive performance across all model sizes.

At the tiny scale (1.5B), CRD-1.5B achieves 91.1% on MATH-500 compared to the baseline DeepSeek-R1-Distill's 83.9%. On AIME'25, CRD-1.5B reaches 35.1%, slightly exceeding ProRL-1.5B [Liu et al., 2025] (33.3%), the current state-of-the-art at the 1.5B scale, while using 2.7× less training dataset with a similar RL-based approach. The gray rows contextualize this result: CRD-1.5B approaches the 7B distilled model on MATH-500 (91.1 vs. 92.8, **4.7× smaller**), while CRD-1.7B (94.5%) surpasses the 14B distilled model (93.9%) on MATH-500 (**8.2× smaller**). This suggests that higher-quality supervision can partly substitute for parameter scaling at small model sizes.

At the small scale (4B), CRD-4B achieves 97.3% on MATH-500 and 70.3% on AIME'25, approaching saturation on standard benchmarks. CRD-4B exceeds Hermes 4 70B [Teknium et al., 2025] (**17× smaller**) and EXAONE Deep 32B [Bae et al., 2026] on AIME'25 and outperforms 32B distilled model (**8× smaller**) on all math and code benchmarks, demonstrating that CRD provides an alternative to parameter scaling. Figure 2a visualizes this parameter efficiency across model scales.

At the large scale (30B), CRD-30B-A3B improves across diverse reasoning domains. It improves mathematical reasoning on AIME'25 from 70.9% to 73.2% (+2.3%), code generation on LiveCodeBench from 62.6% to 65.9% (+3.3%), and STEM reasoning on GPQA-Diamond from 65.8% to 67.2% (+1.4%). These results show that CRD consistently improves performance across all model scales and problem types. Remarkably, CRD-30B-A3B surpasses its **own teacher** DeepSeek-R1 on math and code benchmarks, despite using **12× fewer active parameters** (3B vs 37B). This demonstrates that CRD enables students to exceed teacher capabilities through cross-teacher synthesis.

Comparison with Multi-Teacher Approaches. We compare CRD against existing multi-teacher distillation methods to isolate the benefit of step-level quality curation. Table 2 reports results using the same Qwen2.5-1.5B and Llama-3.1-8B student backbones and the same pool of CRD-generated teacher candidates, isolating the effect of the distillation and curation strategy.

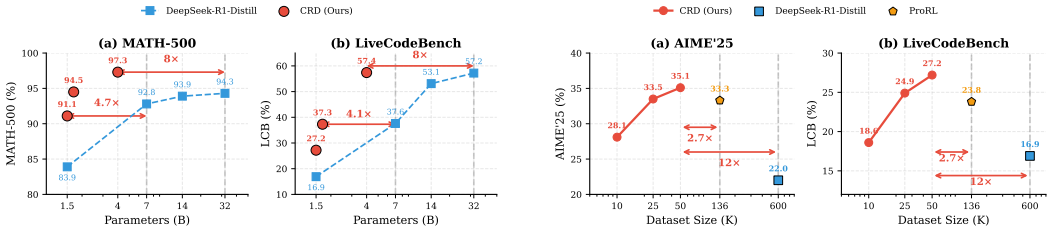
- **TinyLLM** [Tian et al., 2024] uses teacher-forcing CoT to transfer reasoning from multi-teachers, but does not perform step-level quality scoring.

Table 2: Multi-teacher comparison on Qwen-2.5-1.5B, Llama-3.1-8B.

Method	GSM8K	MATH-500	HumanEval	LCB
Qwen-2.5-1.5B (Base)	73.2	55.3	61.6	14.8
TinyLLM	81.8	68.1	64.3	15.7
FAIR	82.3	73.7	65.8	16.3
MAGDi	85.9	72.6	64.9	18.4
CRD (Ours)	**95.2**	**91.1**	**72.3**	**27.2**
Llama-3.1-8B (Base)	84.5	51.9	72.6	16.2
FAIR	86.2	54.8	75.4	18.5
MAGDi	86.9	56.1	76.2	18.8
CRD (Ours)	**92.6**	**62.3**	**85.1**	**22.4**

Table 3: Ablation study on CRD components. Base model is DeepSeek-R1-Distill-Qwen-1.5B.

Configuration	MATH-500	AIME'25	LCB	GPQA-Diamond
Base Model	83.9	22.0	16.9	32.9
+ Cross-Feedback	87.3 +3.4	29.1 +7.1	22.8 +5.9	36.7 +3.8
+ Thinking Path	89.4 +2.1	32.6 +3.5	25.4 +2.6	37.8 +1.1
+ Optimal Budget	89.7 +0.3	33.2 +0.6	25.8 +0.4	38.0 +0.2
+ Judging **(CRD)**	**91.1** +1.4	**35.1** +1.9	**27.2** +1.4	**39.7** +1.7
Total Gain	**↑7.2**	**↑13.1**	**↑10.3**	**↑6.8**



(a) Parameter efficiency comparison.

(b) Data efficiency comparison.

Figure 2: (**a**) CRD achieves comparable performance to DeepSeek-R1-Distill models with 4.7–8× fewer parameters. CRD-1.5B approaches the 7B model on MATH-500 (4.7×), while CRD-4B surpasses the 32B model (8×). (**b**) CRD achieves superior performance with 12× less data than DeepSeek-R1-Distill (600K reasoning samples) and 2.7× less than ProRL (136K), while outperforming both on AIME'25 and LCB.

- **FAIR** [Li et al., 2025b] employs peer-review between teachers to identify student mistakes and filter rationales via acceptance threshold.
- **MAGDi** [Chen et al., 2024] represents multi-agent interactions as graphs and uses contrastive learning between correct and incorrect reasoning.

CRD outperforms all baselines by substantial margins: +18.5% over MAGDi on MATH-500, +9.3% on GSM8K, and +8.8% on LiveCodeBench. This gap stems from CRD's **step-level** granularity: while TinyLLM, FAIR, and MAGDi operate at the chain level, selecting or weighting complete solutions, CRD evaluates each reasoning step independently. A chain with one flawed step is entirely discarded by chain-level methods, but CRD extracts its valid steps and combines them with correct steps from other teachers. This synthesis is particularly impactful on multi-step reasoning (MATH-500: +18.5%), where a single error propagates to incorrect answers.

Generalization Beyond the Training Distribution. Table 4 reports two held-out evaluations. On BoxNet, an out-of-distribution planning task used by ProRL and excluded from our training data, CRD-1.5B is comparable to ProRL at pass@1 and higher at pass@2 and pass@4. On the text-only subset of HLE [Phan et al., 2026], CRD-30B-A3B improves over its Qwen3-30B-A3B base by 3.23 points.

Table 4: Generalization beyond the training distribution. Left: BoxNet planning (100 procedurally generated instances, our re-evaluation of all models, temperature 0.6, top-p 0.95, 32K tokens). Right: HLE text-only subset (official protocol).

BoxNet	pass@1	pass@2	pass@4
DeepSeek-R1-Distill-1.5B	0.0	0.0	0.0
ProRL-1.5B	**7.1**	13.8	22.4
CRD-1.5B	6.8	**14.6**	**25.3**

HLE	Text-only
Qwen3-30B-A3B	11.31
CRD-30B-A3B	**14.54**

3.2 Data Efficiency: Quality Over Quantity

Figure 2b demonstrates CRD's data efficiency. At 50K samples, CRD-1.5B reaches 35.1% on AIME'25, surpassing DeepSeek-R1-Distill (22.0% on 600K reasoning samples) and ProRL (33.3% on 136K samples). CRD achieves this superior performance using **12**$\times$ and **2.7**$\times$ less data, respectively. Notably, CRD-1.5B exhibits consistent data scaling behavior within our curated dataset, improving monotonically from 28.1% at 10K to 35.1% at 50K. This curve is shifted upward by quality-aware curation, so CRD achieves with 10K samples what baselines require 600K to match. DeepSeek-R1-Distill uses 800K total samples (600K reasoning, 200K non-reasoning); we compare against the reasoning portion for fair evaluation.

This efficiency stems mainly from curation, since naive SFT on the same 50K trajectories already reaches 90.5% on MATH-500 (Table 5). RQO adds a consistent further gain by weighting preference pairs with their step-derived quality margins.

CRD improves reasoning distillation efficiency by replacing scale-driven supervision with quality-aware data selection. Instead of relying on a large pool of distillation corpus, CRD constructs a compact 50K-example training set using the same teacher models, yielding a 12$\times$ smaller final trajectory set and a 2.7$\times$ smaller dataset than ProRL's 136K examples. Although CRD generates multiple candidate chains per prompt through cross-feedback, only high-quality trajectories are retained for student learning. As a result, the 1.5B student requires only 1,632 training steps and 8.8 H100 GPU-hours, compared with 7,800 steps and 42 H100 GPU-hours for the 600K distillation baseline and more than 2K steps for ProRL. This supports our main claim that improving the quality and coherence of reasoning supervision can reduce the amount of final training data required by the student.

3.3 Ablation Study: Understanding Each Component

Table 3 isolates the contribution of each CRD component, revealing that they address distinct limitations in reasoning distillation. At the step level, algebra step correctness improves from 85.8% to 92.2% (Appendix F), and cross-feedback converges after two rounds (Appendix G). Appendix J shows representative student outputs.

Cross-Feedback provides the largest individual gain (+3.4–7.1%), with particularly strong improvements on complex benchmarks: +7.1% on AIME'25 and +5.9% on LiveCodeBench. This aligns with our motivation that teachers exhibit complementary error profiles. DeepSeek-R1 excels at creative problem formulation but makes computational errors. Gemini provides rigorous derivations but occasionally shallow reasoning. Cross-feedback enables mutual correction, producing refined candidates that repair errors found in individual teacher outputs.

Thinking Path contributes +1.1–3.5% by enabling step-level fusion across teachers. When Teacher 1's novel approach fails due to a logic bug and Teacher 2's standard approach fails due to a calculation error, Thinking Path stitches Teacher 1's creative setup with Teacher 2's correct computation, yielding a correct solution that neither teacher produced alone.

Optimal Budget and **Judging** provide incremental refinements (+0.2–0.6% and +1.4–1.9% respectively). Optimal Budget implements curriculum-based length constraints that help students learn efficient reasoning without premature truncation [Ding et al., 2025]. Judging adds fine-grained quality signals that enable the student to distinguish between high-quality and mediocre reasoning paths, even when both reach correct answers.

Same-data SFT vs. RQO. Table 5 compares RQO with naive SFT trained on the same 50K Thinking Path trajectories from the same DeepSeek-R1-Distill-Qwen-1.5B checkpoint. Naive SFT already reaches 90.5% on MATH-500, so most of the gain comes from the curated data. RQO adds consistent gains on all four benchmarks (+1.15 points on average).

Single-Teacher vs Multi-Teacher. Figure 3 illustrates the performance gap between single-teacher distillation and CRD's multi-teacher approach. Both Gemini-Flash-Distill and DeepSeek-R1-Distill apply the same scoring and student-training pipeline to a single teacher's outputs, but remove cross-feedback and cross-teacher step stitching. CRD consistently outperforms both across all benchmarks, with particularly large gains on AIME'24 (+14–18%) and LCB (+10%), showing that combining two heterogeneous teachers outperforms either teacher alone.

Table 5: Same-data comparison of training objectives (DeepSeek-R1-Distill-Qwen-1.5B, 50K Thinking Path trajectories).

Method	MATH-500	AIME'25	LCB	GPQA-Diamond
Naive SFT	90.5	33.4	25.7	38.9
RQO (CRD)	**91.1**	**35.1**	**27.2**	**39.7**

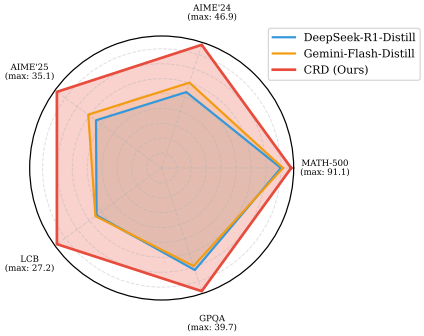


Figure 3: Single-teacher vs. multi-teacher comparison on Qwen2.5-1.5B models.

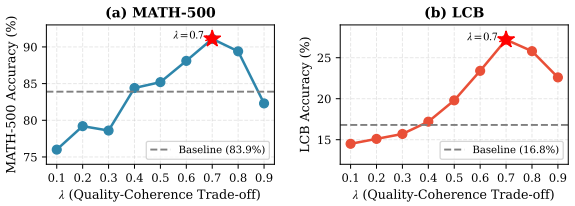


Figure 4: Sensitivity analysis of λ (quality-coherence trade-off). Performance peaks at $\lambda = 0.7$; Smaller λ values over-prioritize coherence, while larger values produce less coherent chains.

Lambda sensitivity. Figure 4 shows the sensitivity of the coherence-quality trade-off parameter λ in Thinking Path. At $\lambda \leq 0.5$, excessive weight on transition coherence leads to selecting plausible-sounding but low-quality steps (MATH-500 drops to 76.0%). At $\lambda > 0.8$, local quality dominates and coherence is underweighted, producing disconnected reasoning chains (82.3%). The optimal value, $\lambda = 0.7$, balances high-quality step selection while maintaining logical flow.

4 Related Work

Reasoning and Distillation. Transferring reasoning capabilities from large models to smaller students via knowledge distillation is a key research area. Prior work has explored distilling Chain-of-Thought demonstrations [Wei et al., 2023], leveraging multiple models for enhanced knowledge transfer [Tian et al., 2024], and iterative refinement techniques [Saunders et al., 2022, Liang et al., 2024]. Contrastive Representation Distillation [Tian et al., 2022] aligns student and teacher embeddings for finer-grained transfer. Recent work demonstrates that budget forcing techniques can yield significant reasoning improvements with small curated datasets [Muennighoff et al., 2025]. LIMO [Ye et al., 2025] shows that a few hundred carefully curated examples can elicit strong mathematical reasoning, while CRD improves each trajectory through cross-teacher feedback and step-level synthesis across mathematics, coding, logic, and STEM. Building upon these foundations, our Collaborative Reasoning Distillation (CRD) framework generates high-quality reasoning datasets through integrated multi-teacher generation, cross-feedback refinement, and fine-grained quality assessment using multi-judge evaluation.

Multi-Teacher Distillation Methods. Recent work has explored various approaches to leverage multiple teacher models for knowledge distillation. TinyLLM uses multi-task SFT where the student imitates both answers and teacher-specific rationales. FAIR uses peer review among teachers to filter rationales with an acceptance threshold. MAGDi distills multi-agent interaction graphs with a contrastive objective between correct and incorrect reasoning. Our CRD framework takes a complementary approach by introducing interactive refinement (cross-feedback) before distillation and explicit step-wise quality scoring during training. This enables the student to learn from the collaborative synthesis of multiple perspectives, resulting in 3.6–23% improvements over these methods when using identical training data (see Section 3.1).

Efficient Reasoning Distillation. Advanced reasoning models such as DeepSeek-R1 [Guo et al., 2025] have demonstrated that extended test-time computation can substantially enhance reasoning capabilities. CRD is complementary to test-time scaling: its primary goal is to improve the quality of distilled supervision, while the Optimal Budget curriculum can additionally encourage shorter student reasoning at inference time. This aligns with recent findings that concise reasoning can

achieve comparable accuracy to verbose chains [Xu et al., 2025], and that reasoning structure often matters more than output length [He et al., 2025]. Our Optimal Budget curriculum enables students to produce effective solutions using a fraction of the tokens required by teacher models, making advanced reasoning accessible for latency-sensitive deployments without sacrificing problem-solving capability.

LLM-as-a-Judge. LLM-as-a-Judge serves as a scalable alternative to human evaluation for assessing intermediate reasoning steps [Gu et al., 2025]. Unlike outcome-based evaluation, step-level assessment provides dense quality signals that help models learn the distinction between logically sound reasoning and lucky guesses. EvalPlanner [Saha et al., 2025] demonstrates that learned evaluation criteria can effectively assess reasoning quality, providing a foundation for our multi-judge approach.

5 Conclusion

This paper presents Collaborative Reasoning Distillation (CRD), a framework that transfers reasoning capabilities from multiple foundation models to compact students through multi-teacher cross-feedback, fine-grained quality assessment, and coherence-aware step synthesis. CRD achieves strong reasoning performance in small models on the evaluated benchmarks: CRD-4B reaches 97.3% on MATH-500, 70.3% on AIME'25, and 57.4% on LiveCodeBench, using only 50K training examples, $12\times$ fewer final training trajectories than standard distillation. CRD demonstrates that step-level synthesis enables solving problems where individual teachers fail.

Limitations and Future Work

CRD relies on LLM teachers and judges and may inherit some of their biases, so plausible but incorrect steps can occasionally receive high scores. Stitching selects paths that score well under our quality and coherence criteria, which does not by itself guarantee that every selected chain is logically correct. We evaluate one teacher pair (DeepSeek-R1 and Gemini 2.5 Flash), and other teacher combinations remain to be explored. The reported training cost covers student optimization only and does not include teacher generation or curation. We compare RQO with naive SFT on the same data but not with DPO. Our out-of-distribution evidence is limited to BoxNet, so we do not claim general distributional robustness. All results come from single training runs because of the cost of LLM-scale experiments, so we do not report error bars.

While our current rubric provides a robust baseline, reliance on static criteria limits scalability across diverse domains. A promising future direction is to transition from hand-crafted rules to *learned quality models* trained on preference data. Such models could dynamically adapt to domain-specific nuances and enable more scalable supervision without manual rubric engineering. Furthermore, extending our framework to handle heterogeneous teacher feedback could mitigate potential sycophancy, paving the way for more robust collaborative reasoning systems. Future work includes larger and all-open-weight teacher pools such as QwQ-32B and GPT-OSS, and stronger teachers for expert-level benchmarks such as HLE. We also plan controlled comparisons with on-policy distillation and SFT-based baselines, formal bounds on stitching failures, and a larger human audit of stitched chains.

Acknowledgments and Disclosure of Funding

We appreciate anonymous reviewers for providing constructive feedback and suggestions. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST); No. RS-2026-25507250, Development of a Fabric Platform for Ultra-low Latency, Lossless, and High-bandwidth AI Data Centers; No. RS-2024-00436680, Collaborative Research Projects with Microsoft Research), and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00406715).

References

Mohammad Abdollahi, Khandaker Rifah Tasnia, Soumit Kanti Saha, Jinqiu Yang, Song Wang, and Hadi Hemmati. Demystifying errors in llm reasoning traces: An empirical study of code execution simulation, 2025. URL https://arxiv.org/abs/2512.00215.

Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Kijeong Jeon, Gerrard Jeongwon Jo, Hyunjik Jo, Jiyeon Jung, Hyosang Kim, Joonkee Kim, Seonghwan Kim, Soyeon Kim, Sunkyoung Kim, Yireun Kim, Yongil Kim, Youchul Kim, Edward Hwayoung Lee, Haeju Lee, Honglak Lee, Jinsik Lee, Kyungmin Lee, Sangha Park, Yongmin Park, Sihoon Yang, Heuiyeen Yeen, Sihyuk Yi, and Hyeongu Yun. Exaone deep: Reasoning enhanced language models, 2026. URL https://arxiv.org/abs/2503.12524.

Ding Chen, Qingchen Yu, Pengyuan Wang, Mengting Hu, Wentao Zhang, Zhengren Wang, Bo Tang, Feiyu Xiong, Xinchi Li, Chao Wang, Minchuan Yang, and Zhiyu Li. xverify: Efficient answer verifier for reasoning model evaluations, 2025. URL https://arxiv.org/abs/2504.10481.

Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. Magdi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models, 2024. URL https://arxiv.org/abs/2402.01620.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models, 2025. URL https://arxiv.org/abs/2405.21015.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456.

Dongyi Ding, Tiannan Wang, Chenghao Zhu, Meiling Tao, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Micota: Bridging the learnability gap with intermediate cot and teacher assistants, 2025. URL https://arxiv.org/abs/2507.01887.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning, 2023. URL https://arxiv.org/abs/2301.12726.

Tianchen Gao, Jiashun Jin, Zheng Tracy Ke, and Gabriel Moryoussef. A comparison of deepseek and other llms, 2025. URL https://arxiv.org/abs/2502.03688.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025. URL https://arxiv.org/abs/2411.15594.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. Train long, think short: Curriculum learning for efficient reasoning, 2025. URL https://arxiv.org/abs/2508.08940.

Muyu He, Muhammad Ali Shafique, Anand Kumar, Tsach Mackey, and Nazneen Rajani. The valley of code reasoning: Scaling knowledge distillation of large language models, 2025. URL https://arxiv.org/abs/2510.06101.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

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, 2024. URL https://arxiv.org/abs/2403.07974.

Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, Yulu Jia, Sun He, Hongmin Chen, Zhihao Bai, Qi Hou, Shipeng Yan, Ding Zhou, Yiyao Sheng, Zhuo Jiang, Haohan Xu, Haoran Wei, Zhang Zhang, Pengfei Nie, Leqi Zou, Sida Zhao, Liang Xiang, Zherui Liu, Zhe Li, Xiaoying Jia, Jianxi Ye, Xin Jin, and Xin

Liu. Megascale: Scaling large language model training to more than 10,000 gpus, 2024. URL https://arxiv.org/abs/2402.15627.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. Explanations from large language models make small reasoners better, 2022. URL https://arxiv.org/abs/2210.06726.

Wendi Li and Yixuan Li. Process reward model with q-value rankings, 2025. URL https://arxiv.org/abs/2410.11287.

Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. Leash: Adaptive length penalty and reward shaping for efficient large reasoning model, 2025a. URL https://arxiv.org/abs/2512.21540.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. Learning from committee: Reasoning distillation from a mixture of teachers with peer-review, 2025b. URL https://arxiv.org/abs/2410.03663.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate, 2024. URL https://arxiv.org/abs/2305.19118.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025. URL https://arxiv.org/abs/2505.24864.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain, 2025. URL https://arxiv.org/abs/2501.15587.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. URL https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason, 2023. URL https://arxiv.org/abs/2212.08410.

Mathematical Association of America. AIME 2024 problems and solutions, 2024. URL https://artofproblemsolving.com/wiki/index.php/2024_AIME_I.

Mathematical Association of America. AIME 2025 problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_I.

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, 2025. URL https://arxiv.org/abs/2501.19393.

Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dan Hendrycks, et al. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://doi.org/10.1038/s41586-025-09962-4.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan & reason for evaluation with thinking-llm-as-a-judge, 2025. URL https://arxiv.org/abs/2501.18099.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators, 2022. URL https://arxiv.org/abs/2206.05802.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work?, 2021. URL https://arxiv.org/abs/2106.05945.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2505.24760.

Ryan Teknium, Roger Jin, Jai Suphavadeeprasit, Dakota Mahan, Jeffrey Quesnelle, Joe Li, Chen Guang, Shannon Sands, and Karan Malhotra. Hermes 4 technical report, 2025. URL https://arxiv.org/abs/2508.18255.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V. Chawla. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation, 2024. URL https://arxiv.org/abs/2402.04616.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2022. URL https://arxiv.org/abs/1910.10699.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. Unlocking efficient long-to-short llm reasoning with model merging, 2025. URL https://arxiv.org/abs/2503.20641.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less, 2025. URL https://arxiv.org/abs/2502.18600.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges, 2025. URL https://arxiv.org/abs/2310.17631.

A Cross-Feedback Review Prompt

This section provides the complete prompt template used in the cross-feedback refinement step (Section 2.1) to instruct teacher models to critically review peer reasoning chains across mathematical, coding, and logical reasoning tasks.

```
[SYSTEM]
You are an expert reasoning evaluator specializing in mathematical problem-solving, code
     generation, and logical deduction. Your task is to critically review a reasoning chain
     from another model and generate an improved solution.

Be rigorous and critical in your evaluation. Do not accept superficially correct answers--
     identify subtle logical gaps, unstated assumptions, and potential edge cases that the
     original solution may have overlooked.

[PROBLEM]
{problem}

[DOMAIN]
{domain}  # One of: MATH, CODE, LOGIC

[PEER REASONING CHAIN]
The following is a reasoning chain from another teacher model. Your task is to review it
     critically and improve upon it.

--- Other Teacher's Output ---
{other_teacher_reasoning_chain}

[REVIEW INSTRUCTIONS]
Analyze the peer chain thoroughly. Your review must cover both general reasoning quality and
     domain-specific criteria.

=== General Reasoning Criteria ===
1. Logical Flow:
   - reasoning:coherence - Are transitions between steps justified?
   - reasoning:completeness - Are all necessary steps present?
   - reasoning:assumptions - Are implicit assumptions stated and valid?

2. Error Detection:
   - reasoning:contradiction - Does any step contradict previous ones?
   - reasoning:circular - Is there circular reasoning?
   - reasoning:leap - Are there unjustified logical leaps?

3. Quality Assessment:
   - reasoning:clarity - Is the reasoning clear and followable?
   - reasoning:efficiency - Is the solution unnecessarily verbose?
   - reasoning:novelty - Does it offer creative insights?

=== Domain-Specific Criteria ===

[If MATH]:
   - math:algebraic_validity - Are algebraic manipulations correct?
   - math:arithmetic - Are calculations accurate?
   - math:constraint_satisfaction - Are problem constraints respected?
   - math:case_coverage - Are all cases (including edge cases) handled?
   - math:theorem_applicability - Are theorems applied with valid preconditions?
   - math:unit_consistency - Are units handled correctly (if applicable)?

[If CODE]:
   - code:compilable - Does the code compile/run without syntax errors?
   - code:correctness - Does it produce correct output for given inputs?
   - code:edge_cases - Are boundary conditions handled?
   - code:efficiency - Is time/space complexity reasonable?
   - code:readability - Is the code well-structured and readable?
   - code:error_handling - Are potential runtime errors handled?

[If LOGIC]:
   - logic:premise_validity - Are premises clearly stated and valid?
   - logic:inference_rules - Are deduction rules correctly applied?
   - logic:quantifier_scope - Are quantifiers (all, some, none) used correctly?
   - logic:negation - Are negations handled properly?
   - logic:counterexample - Has the solution considered counterexamples?
   - logic:soundness - Is the argument logically sound?

[TASK]
Based on your critical review:
1. Identify ALL errors, gaps, and weaknesses (be thorough, not lenient)
2. Note any strengths worth preserving
3. Generate an improved reasoning chain that fixes identified issues
```

```
[OUTPUT FORMAT]
<review>
[Critical analysis of the peer chain]
- Strengths: [What to preserve]
- Weaknesses: [Errors, gaps, issues found - be specific with tags]
- Verdict: [ACCEPT_WITH_MINOR_FIXES | NEEDS_MAJOR_REVISION | REJECT_AND_REDO]
</review>

<think>
[Your improved step-by-step reasoning that addresses all identified issues]
</think>

<answer>
[Final answer]
</answer>
```

B Mathematical Scoring Rubric (T-LCM)

This section provides the full scoring rubric used for mathematical reasoning evaluation. The **T-LCM (Taxonomy-Guided Logic-Consistency Metric)** protocol operates in two phases: domain classification at the problem level, followed by step-wise scoring with explicit decision boundaries.

B.1 Score Range and Decision Boundaries

Table 6: T-LCM scoring rubric with decision boundaries for mathematical reasoning.

Score	Grade	Description	Decision Boundary
10.0	Rigorous	Flawless execution, all constraints satisfied, all cases covered.	N/A (Perfect)
8.0–9.9	Valid	Correct logic with minor inefficiency (redundancy, non-standard notation).	Does inefficiency obscure argument?
6.0–7.9	Minor Slip	Recoverable execution errors (calculation, sign, transcription, off-by-one).	Can error be fixed by single value change?
3.0–5.9	Structural Flaw	Significant reasoning errors (constraint violation, theorem misuse, missing cases).	Is there any mathematical basis?
0.0–2.9	Critical Failure	Fundamental logic breakdown (hallucination, logical leap, circular reasoning).	Is reasoning entirely baseless?

B.2 Error Taxonomy

We categorize errors into three types based on severity and recoverability:

Type I (Execution Errors, 6.0–7.9): Calculation errors where correct formula is applied but arithmetic is wrong; sign errors with correct equation structure; transcription errors copying values incorrectly; off-by-one boundary errors in discrete domains; unit oversight with approximately correct magnitude.

Type II (Reasoning Errors, 3.0–5.9): Constraint violations ignoring core problem conditions; conceptual misapplication using theorems where preconditions are not met; incomplete case analysis missing critical branches; variable scope errors reusing symbols for different meanings; premature approximation causing significant deviation.

Type III (Logic Breakdown, 0.0–2.9): Hallucination inventing non-existent theorems or values; logical leaps concluding without justification; circular reasoning using conclusions as premises; contradiction with previous steps; irrelevance to the problem.

B.3 Domain-Specific Checklists

The scoring protocol is domain-adaptive. For **algebra**, we check sign consistency, factoring validity, inequality direction, and domain restrictions. For **geometry**, we verify diagram consistency, angle/length properties, and similarity/congruence conditions. For **number theory**, we validate modular arithmetic, divisibility claims, and integer constraints. For **combinatorics**, we examine counting methods, overcounting adjustments, and boundary conditions.

B.4 Full Judge Prompt Template

Below is the complete prompt template used for mathematical reasoning evaluation, inspired by xVerify [Chen et al., 2025]:

```
[SYSTEM]
You are a Mathematics Olympiad Judge. Evaluate the given reasoning step using the T-LCM
    protocol.

[DOMAIN] {domain}

[RUBRIC]
- 10.0 (Rigorous): Flawless deduction, all constraints satisfied, all cases covered.
- 8.0-9.9 (Valid): Correct logic but contains redundancy or non-standard notation.
- 6.0-7.9 (Minor Slip): Execution errors (calculation, sign, transcription, off-by-one) with
    correct structure.
- 3.0-5.9 (Structural Flaw): Reasoning errors (constraint violation, theorem misuse, missing
    cases).
- 0.0-2.9 (Critical Failure): Hallucination, logical leap, circular reasoning, contradiction.

[DECISION BOUNDARIES]
- 8.0 vs 7.9: Does inefficiency obscure the argument? (Yes -> below 8.0)
- 6.0 vs 5.9: Can error be fixed by changing a single value? (Yes -> 6.0+)
- 3.0 vs 2.9: Is there any mathematical basis? (Yes -> 3.0+)

[DOMAIN-SPECIFIC CHECKLIST: {domain}]
{domain_checklist}

[INPUT]
Problem: {problem}
Previous Steps: {history}
Current Step: {current_step}

[OUTPUT FORMAT]
Score: [0.0-10.0]
Error Type: [None | Calculation | Sign | Transcription | Off-by-One | Constraint Violation |
    Theorem Misuse | Missing Case | Hallucination | Logical Leap | Circular | Contradiction |
     Irrelevance]
Rationale: [Brief explanation]
```

C Coherence-Aware Thinking Path Algorithm

This section provides the detailed algorithm for Thinking Path, our coherence-aware pruning method that curates high-quality reasoning steps from multiple teacher outputs while preventing the Frankenstein effect, the selection of high-scoring but logically disconnected steps that form an incoherent chain.

C.1 Algorithm Overview

Unlike naive greedy pruning that independently selects the highest-scoring step at each position, our **Coherence-Aware Thinking Path** algorithm optimizes a joint objective of **Quality** (from the aggregated JudgeLM and xVerify step scores) and **Coherence** (transition coherence between consecutive steps). This formulation is solved efficiently using Viterbi-style dynamic programming.

- **Node Score** $q(s)$: The aggregated step-quality score v_t (25% trimmed mean of the four JudgeLM and xVerify scores; Section 2.2), normalized to $[0, 1]$.
- **Edge Weight**: The transition-coherence score between consecutive steps s_{t-1} and s_t, computed from the cosine similarity of their embeddings and rescaled to $[0, 1]$.

Complexity Analysis. The time complexity of our Coherence-Aware Path Finding is $O(T \cdot N^2)$, where T is the maximum number of reasoning steps per chain and N is the number of teacher models. Since N is small ($N = 2$ in our experiments) and reasoning steps are bounded ($T < 20$ for most problems), the computational overhead is negligible compared to the LLM generation cost of initial generation and cross-feedback. The space complexity is $O(T \cdot N)$ for storing the dynamic programming table and backpointers.

C.2 Implementation

```
def coherence_aware_thinking_path(chains, lambda_=0.7, tau=7.0):
    chains = align_steps(chains)  # embedding-based step alignment with lookahead merging

    num_steps = max(len(c) for c in chains)
    num_teachers = len(chains)

    # DP Table: dp[t][i] = max cumulative score ending at teacher i, step t
    dp = [[-float('inf')] * num_teachers for _ in range(num_steps)]
    parent = [[-1] * num_teachers for _ in range(num_steps)]

    # Initialization (Step 0)
    for i in range(num_teachers):
        if len(chains[i]) > 0:
            dp[0][i] = normalize(chains[i][0]['score'])  # Scale to 0-1

    # Forward Pass (Viterbi Search)
    for t in range(1, num_steps):
        for curr_i in range(num_teachers):
            if t >= len(chains[curr_i]):
                continue

            # 1. Quality score (aggregated JudgeLM/xVerify step score)
            q_score = normalize(chains[curr_i][t]['score'])

            best_prev_val = -float('inf')
            best_prev_idx = -1

            for prev_i in range(num_teachers):
                if t-1 >= len(chains[prev_i]):
                    continue

                # 2. Coherence score C(u, v) (transition coherence)
                c_score = get_transition_coherence(
                    chains[prev_i][t-1]['content'],
                    chains[curr_i][t]['content']
                )

                # 3. Joint Score: Quality + Coherence
                transition_score = dp[t-1][prev_i] + \
                    (lambda_ * q_score) + \
                    ((1 - lambda_) * math.log(c_score + 1e-9))

                if transition_score > best_prev_val:
                    best_prev_val = transition_score
                    best_prev_idx = prev_i

            dp[t][curr_i] = best_prev_val
            parent[t][curr_i] = best_prev_idx

    # Backward Pass (Path Reconstruction)
    final_idx = num_steps - 1
    best_end = max(range(num_teachers), key=lambda i: dp[final_idx][i])

    optimal_path = []
    curr = best_end
    for t in range(final_idx, -1, -1):
        step = chains[curr][t]
        if step['score'] >= tau:  # Final quality threshold
            optimal_path.append(step)
        curr = parent[t][curr]

    return optimal_path[::-1]  # Reverse to chronological order

def get_transition_coherence(prev_content, curr_content):
    """Compute C(prev, curr) with rescaled embedding cosine similarity."""
    return (1 + cosine_similarity(embed(prev_content), embed(curr_content))) / 2  # rescale
         [-1, 1] to [0, 1]
```

C.3 Key Design Choices

The algorithm embodies several important design decisions that distinguish it from simple greedy selection.

The **joint optimization** of quality and coherence prevents the Frankenstein effect where high-scoring but disconnected steps form an invalid reasoning chain. The hyperparameter λ controls this trade-off: higher values ($\lambda > 0.5$) prioritize step quality scores, while lower values emphasize smooth transitions between steps. In our experiments, $\lambda = 0.7$ achieved the best balance, allocating 70% weight to quality and 30% to coherence.

The **Viterbi-style dynamic programming** ensures global optimality in $O(T \cdot N^2)$ time, where T is the maximum chain length and N is the number of teachers. This is efficient for typical settings ($N = 2, T < 20$).

D Training Data Composition

Table 7 lists the number of training instances drawn from each source.

Table 7: Composition of the CRD-50K training dataset.

Source Dataset	Domain	Samples
DeepScaleR	Math	20K
Eurus-2-RL	Code	15K
Reasoning Gym	Logical	10K
SCP-116K	STEM	5K
Total		**50K**

E Benchmark Descriptions

We evaluate CRD on nine benchmarks spanning mathematics, code generation, STEM reasoning, planning, and expert-level knowledge:

- **MATH-500** [Lightman et al., 2023]: Competition mathematics problems requiring multi-step reasoning across the seven MATH subjects (prealgebra, algebra, number theory, counting and probability, geometry, intermediate algebra, and precalculus).
- **LiveCodeBench** [Jain et al., 2024]: Code generation benchmark with execution-based evaluation measuring functional correctness.
- **GPQA-Diamond** [Rein et al., 2023]: Graduate-level questions in physics, biology, and chemistry requiring deep domain knowledge.
- **AIME'24** [Mathematical Association of America, 2024] **and AIME'25** [Mathematical Association of America, 2025]: American Invitational Mathematics Examination problems spanning algebra, geometry, number theory, and combinatorics that demand creative multi-step solutions.
- **GSM8K** [Cobbe et al., 2021]: Grade-school math word problems, used for the multi-teacher comparison (Table 2).
- **HumanEval** [Chen et al., 2021]: Python function synthesis evaluated with unit tests, used for the multi-teacher comparison (Table 2).
- **BoxNet** [Stojanovski et al., 2025]: Procedurally generated multi-step planning that moves colored boxes to their targets, used as an out-of-distribution test.
- **HLE** [Phan et al., 2026]: Expert-level questions across many academic fields. We use the text-only subset.

F Detailed Analysis

F.1 MATH-500 Subject Breakdown

MATH-500 covers seven subjects; we analyze algebra, precalculus, and number theory. To quantify the impact of collaborative feedback, we measured per-subject Chain-of-Thought accuracy by matching generated reasoning steps against official solution templates. Algebra shows the clearest improvement: on a 50-problem slice, the DeepSeek-R1-1.5B baseline attains 85.8% step correctness (482/562 steps), whereas CRD-1.5B reaches 92.2% (688/746 steps). Precalculus benefits by 6–8 percentage points in function analysis and limit computations, while number theory improves by 5–7 points in divisibility and modular-arithmetic reasoning.

F.2 Scaling Analysis

Our results reveal consistent scaling patterns. On the DeepSeek-R1-Distill-Qwen-1.5B student, CRD improves MATH-500 by 7.2 points (83.9 $\rightarrow$ 91.1). For Qwen3 students, MATH-500 gains shrink as the base model saturates (1.7B: +1.1, 4B: +0.3, 30B-A3B: +0.3), while gains on the harder AIME'25 persist (+5.4, +4.7, +2.3), suggesting that the marginal benefit of CRD diminishes on near-saturated benchmarks but remains on harder ones.

G Multi-Round Cross-Feedback Analysis

We analyze the effect of multiple cross-feedback rounds on final model performance.

Table 8: Impact of cross-feedback rounds on model performance. Evaluated on Qwen3-1.7B student model. Performance converges at round 2–3.

Round	MATH-500	GPQA-Diamond	LCB	Marginal Gain
Initial (0)	93.4	40.1	33.2	–
Round 1	93.8	41.5	34.7	+0.4 / +1.4 / +1.5
Round 2	94.0	42.2	35.1	+0.2 / +0.7 / +0.4
Round 3	94.0	42.2	35.0	0 / 0 / -0.1

Performance converges at round 2–3, with additional rounds providing no benefit. The marginal utility of cross-feedback diminishes rapidly, suggesting that 2 rounds achieve the optimal quality-efficiency trade-off.

H Judge Reliability

We reduce dependence on any single evaluator with two heterogeneous judges (JudgeLM and xVerify), two evaluations per judge, a 25% trimmed mean over the four scores, and domain-specific verification (execution checks for code and answer-equivalence checks for mathematics). Re-scoring all steps in our evaluation set with GPT-4o, given the same preceding history and current step, yields a mean absolute difference of 0.428 from our aggregated score on the 0–10 scale. On MATH-500, the full dual-judge configuration improves the student by 1.4 points, compared with 0.6 with xVerify alone, 0.4 with JudgeLM alone, and 0.6 with GPT-4o alone. Correlated evaluator biases may remain, and plausible but incorrect reasoning may still receive high scores.

I Stitched Training Example

Table 9 shows a stitched training trajectory for a problem that asks for the fraction of an equilateral triangle lying outside a circle tangent to two of its sides. The original chain derives the first two steps correctly but then assumes only an exterior-center configuration. Cross-feedback produces a corrected continuation, and Thinking Path selects $A_1 \rightarrow A_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_5$.

Table 9: A stitched training trajectory (A: original chain, B: cross-feedback continuation).

Step	Content	Score	Decision
A_1	Derives $\angle BOC = 120^\circ$	9.3	Retain
A_2	Derives $r = s/\sqrt{3}$	9.0	Retain
A_3	Assumes only an exterior center	3.6	Reject
A_4	Applies an absolute value to repair a negative area	0.8	Reject
B_3	Identifies two feasible center configurations	9.4	Retain
B_4	Rejects the exterior case by contradiction	9.1	Retain
B_5	Computes the valid minor-segment area	8.9	Retain

J Student Output Examples

We compare DeepSeek-R1-Distill-Qwen-1.5B and CRD-1.5B on a held-out MATH-500 problem with identical prompts and decoding. The problem asks for $p(8)$, where $p(x)$ is a polynomial of degree 5 such that $p(n) = \frac{n}{n^2-1}$ for $n = 2, 3, \ldots, 7$.

DeepSeek-R1-Distill-Qwen-1.5B (incorrect). The model defines $q(x) = x^2 - 1 - p(x)$ and asserts $q(2) = \cdots = q(7) = 0$, which does not follow from the condition. It then writes this degree-5 polynomial as a product of six linear factors and outputs $p(8) = -657$.

CRD-1.5B (correct). The model constructs $Q(x) = (x^2 - 1)p(x) - x$, notes that $Q(n) = 0$ for $n = 2, \ldots, 7$, keeps the degree-7 structure, and uses $Q(1) = -1$ and $Q(-1) = 1$ to determine the remaining linear factor. It obtains $Q(8) = -259/56$ and $63p(8) - 8 = -259/56$, and returns $p(8) = 3/56$.

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 include the paper's main claims and contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in the Limitations and Future Work section after the Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: NA

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The end-to-end system and methods are thoroughly described. The models, dataset, and hyperparameters used for the evaluation are provided as much as possible.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: We do not release code at this time. The training data come from public datasets (Section 3, Appendix D), and the prompts, scoring rubric, and stitching algorithm are given in Appendices A–C.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All the experiment settings and methods are specified in detail.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: [No]

Justification: Due to the extensive computational requirements of large language model experiments, we limit our evaluation to a single run.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the type of compute resources used for each experiment setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Foundational research on reasoning distillation, not tied to a specific application.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We provide new methodology for distillation.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models, data, and code we used or reproduced are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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 (if any)?

    Answer: [N/A]

    Justification: No human subject was involved in the study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) 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 (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subject was involved in the study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: CRD uses LLMs as core components, with DeepSeek-R1 and Gemini 2.5 Flash Thinking as teachers and JudgeLM and xVerify as step-level judges.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.