ICCDi: Incremental Distillation with Multi-Agent Process Supervision for Small Language Model Reasoning

Anonymous ACL submission

Abstract

Multi-agent structured distillation has emerged as a promising paradigm to enhance the reasoning capabilities of small language models by leveraging structured interactions among multiple agents. However, existing methods typically rely on complete interaction trajectories for supervision, which lack process-level signals aligned with the student's own reasoning path and introduce additional computational cost. To address these issues, a novel distillation framework is proposed, namely **I**ncremental **C**onsensus-based **C**orrection **Di**stillation (ICCDi) that provides multi-agent process supervision. ICCDi dynamically constructs step-level supervision signals along the student's reasoning trajectory. Based on these signals, we design structured training objectives that enable the student to learn high-quality reasoning paths, estimate step-wise reliability and align decision preferences at critical error points, thereby achieving more effective reasoning knowledge transfer. Extensive experiments on five challenging reasoning benchmarks demonstrate that ICCDi significantly improves reasoning accuracy while maintaining strong token efficiency, outperforming both single-teacher and existing multi-teacher distillation baselines.

1 Introduction

Large language models (LLMs) have shown strong capabilities on complex reasoning tasks (Yang et al., 2025; Team et al., 2026), but their high computational cost poses significant challenges for real-world deployment. To improve inference efficiency, recent research has increasingly focused on smaller language models. However, small language models generally exhibit greater instability on complex reasoning tasks. How to improve the stability and reliability of reasoning in small language models has become an important research problem.

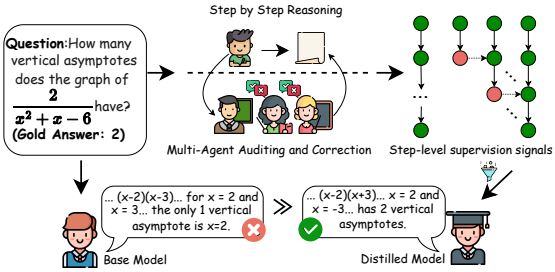


Figure 1: Overview of the ICCDi. Student-generated steps are audited via multi-agent consensus to produce reliable step-level supervision, which is then converted into structured distillation objectives for improving reasoning.

One common strategy in existing research to improving the reasoning capability of small language models is knowledge distillation, which transfers the knowledge from larger teacher models to smaller student models (Hinton et al., 2015). While effective in reducing model size, traditional distillation typically follows a static training paradigm. The student model merely receives generic training signals, lacking iterative guidance regarding specific reasoning errors from the teacher (Shinn et al., 2023; Hsieh et al., 2023; Wang et al., 2024a). Consequently, much of the teacher's reasoning knowledge cannot be effectively transferred to the student model. Recent works have explored multi-agent collaboration frameworks inspired by mechanisms of human communication, collaboration and reflection (Wu et al., 2023b; Chen et al., 2024a; Zhou et al., 2025). In these frameworks, multiple agents separately reason about a problem and critique or revise each other's reasoning processes before reaching a consensus. These multi-agent interactions introduce diverse reasoning perspectives and generate rich intermediate reasoning signals that benefit the training of small language models.

Despite these advantages, leveraging multi-agent frameworks to improve small language mod-

els faces two critical challenges. **(i) Lack of structured supervision for student models.** Existing multi-agent frameworks are primarily outcome-oriented, focusing on obtaining the final answer. Their interaction procedures are not designed to provide structured supervision tailored to the learning student models. **(ii) Noisy supervision from complex interactions.** Multi-agent interaction trajectories are complex and contain many low-quality reasoning, which can introduce noise and weaken the effect of training for small models.

To address these challenges, we propose **I**ncremental **C**onsensus-based **C**orrection distillation (**ICCDi**), a structured distillation framework aimed to enhance the reasoning capabilities of small language models. As illustrated in fig. 1, ICCDi leverages multi-agent consensus to incrementally audit the student model's reasoning process and construct step-level supervision signals aligned with the student model's reasoning distribution. Based on these signals, ICCDi further optimizes the student model through a set of complementary step-level training objectives, enabling it to learn structured reasoning behaviors rather than final answers.

In summary, the main contributions of this paper include:

- We propose ICCDi, which constructs step-level supervision signals by incrementally auditing the student model's reasoning process, thereby enabling process-level supervision aligned with the student model's reasoning distribution.

- We design a set of structured step-level training objectives that enable the student model to learn acceptable reasoning trajectories, estimate the reliability of reasoning steps, and align correct decision preferences at critical branching points.

- Extensive experiments on multiple mathematical and scientific reasoning benchmarks show that ICCDi improves reasoning accuracy while also enhancing token efficiency.

2 Related Work

We review research progress closely related to the work in this paper from three perspectives.

2.1 Multi-Agent for Reasoning

Multi-agent reasoning frameworks leverage multiple agents to improve reasoning reliability through independent generation or structured interaction (Guo et al., 2024; Zhang et al., 2024b). Existing approaches generally follow two directions: the first focuses on generating multiple candidate outputs followed by joint evaluation or votingg (e.g., Xiong et al., 2023; Chen et al., 2024c; Sun et al., 2024); the second is role-based debate, where agents interact under predefined roles to refine reasoning (e.g., Wang et al., 2024c; Liang et al., 2024; Hsu et al., 2025). Recent work shows that such interactions produce rich intermediate signalse.g., disagreements, rebuttals, and correctionswhich can serve as supervision for training small models. Representative methods include MAGDi (Chen et al., 2024b), which models multi-agent reasoning as structured graphs for distillation, and D&R (Zhou et al., 2025), which leverages debate and self-reflection to construct preference-level supervision.

2.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is a widely used technique for transferring knowledge from large teacher models to smaller student models, offering an effective trade-off between performance and efficiency. Early approaches focus on white-box settings, leveraging teacher output distributions and internal representations (Gu et al., 2026), while more recent work adopts symbolic distillation for black-box scenarios and instruction tuning (Xu et al., 2024; Zhu et al., 2024). For complex reasoning tasks, Chain-of-Thought (CoT) prompting enables distillation to move beyond final-answer supervision toward structured reasoning processes (Wang et al., 2023; Dai et al., 2025), with diverse prompting strategies further improving performance on multi-step tasks (Mitra et al., 2023). Some methods also exploit multi-model interactions, such as debate, to extract richer supervision signals (Chen et al., 2024b; Zhou et al., 2025). However, most existing approaches rely on static supervision that is independent of the student's actual reasoning trajectory, lacking dynamic, process-level signals aligned with on-policy generation.

2.3 Process Supervision

Outcome-based supervision can lead models to produce correct answers relying on flawed or incomplete intermediate reasoning steps (Uesato et al., 2022). To address this inconsistency between reasoning processes and outcomes, process supervision introduces step-level signals for finer-grained guidance beyond final answers (Lightman et al., 2023; Shao et al., 2024). A common approach is to introduce a step-level evaluator, known as a Process Reward Model (PRM), which evaluates the validity of each reasoning step within a chain of thought under a given context (Wang et al., 2024a; Zhang et al., 2024a; Qu et al., 2024). However, training PRMs typically relies on human-annotated process supervision signals, which poses significant challenges in terms of data scalability and annotation consistency. Moreover, PRMs depend on a single evaluation source to score intermediate steps, making their supervision signals susceptible to subjective bias or model hallucinations (Pan et al., 2024; Lu et al., 2024a). When the evaluation signals are unreliable, process supervision may introduce noise and negatively impact the learning process.

3 Method

In this section, we present **ICCDi**, an incremental distillation framework that constructs step-level supervision signals through multi-agent consensus over the student's reasoning trajectory. These signals provide fine-grained guidance aligned with the student's reasoning process, enabling effective learning of structured reasoning behaviors. Section 3.1 describes how step-level supervision signals are constructed via incremental reasoning and multi-agent consensus auditing. Section 3.2 then introduces the distillation procedure, where these signals are used to train the student model through three complementary step-level objectives.

3.1 Step-Level Supervision Construction

As illustrated in fig. 2, ICCDi constructs step-level supervision signals through a student-centered incremental reasoning framework. This process consists of two key components: (i) incremental generation and (ii) multi-agent consensus auditing.

(i) Incremental generation. For small language models M_{std} with limited reasoning capability, generating a complete reasoning trajectory in a single pass often leads to error accumulation. To address it, ICCDi decomposes reasoning into a sequence of step-wise decisions. At the t-th step, the student model receives the *Verified Context* C_{t-1} which contains previously accepted reasoning steps and generates a candidate reasoning step under instruction constraints $\mathcal{P}_{step}$:

$$s_t = M_{std}(q, C_{t-1}, \mathcal{P}_{step}) \tag{1}$$

where q denotes the input question. Each generated step is then passed to the teacher panel for validation.

(ii) Multi-Agent Consensus Auditing (MACA). Each candidate reasoning step s_t is evaluated by multiple teacher agents to determine its validity under the current context C_{t-1}. The auditing process considers (i) logical consistency with the existing reasoning trajectory, (ii) factual correctness and (iii) whether the step degrades the reasoning process (e.g., circular reasoning or irrelevant expansion). If a step is accepted by consensus, it is appended to the reasoning trajectory. Otherwise, the trajectory is truncated at the current step, and a correction mechanism is triggered to guide subsequent reasoning.

Student Continuation. Following rejection, teacher agents provide a corrected step $\hat{s}_k$ under the same context C_{k-1} and the student model continues generating subsequent steps s_{k+1} from the corrected trajectory. This enables the framework to expose and correct reasoning errors at different stages while maintaining alignment with the student's reasoning distribution.

Teacher Takeover. If the number of rejected steps exceeds a predefined threshold K, a teacher model $\mathcal{T}$ completes the remaining reasoning steps to ensure trajectory completeness. This mechanism is applied only in failure cases to avoid introducing excessive teacher bias. The detailed consensus protocol is described in Appendix B.1.

Through incremental generation and multi-teacher consensus auditing and correction, ICCDi constructs process-level supervision signals aligned with the student model's reasoning trajectory, which are subsequently used for training.

3.2 Distillation

Based on the process-level supervision signals constructed in section 3.1, we design three structured step-level training objectives to train the student model. These objectives operate on different subsets of the supervision signals. This design en-

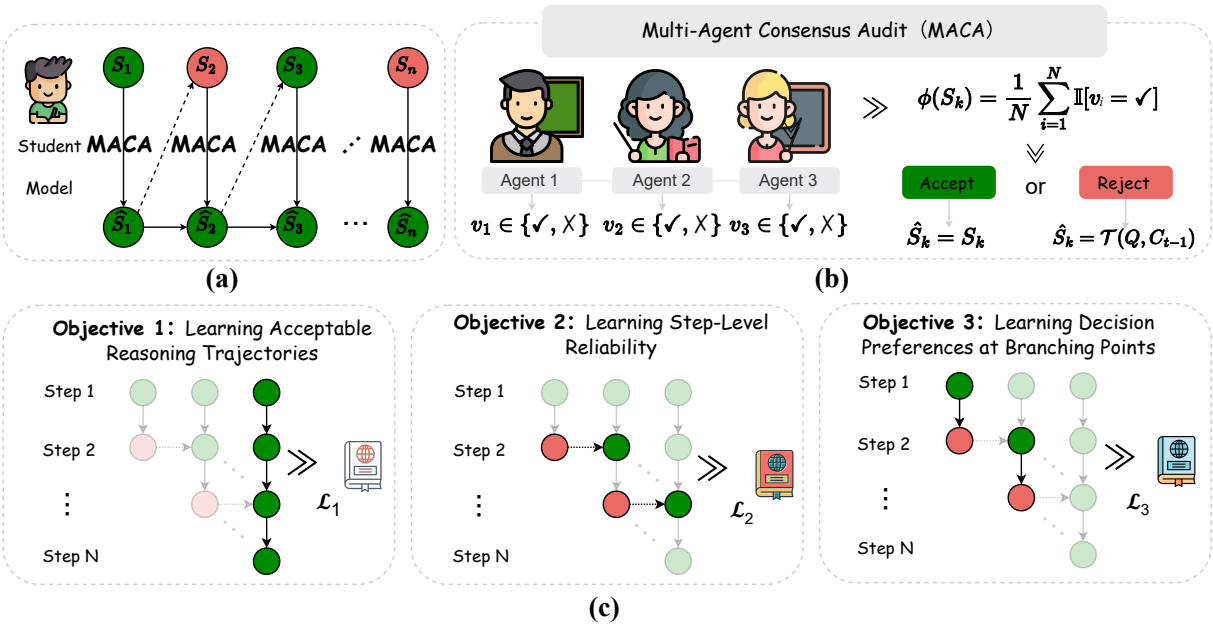


Figure 2: Detailed process of ICCDi. (a) illustrates the construction process of step-level signals, where each reasoning step generated by the student model is reviewed by MACA, ultimately forming a complete step-level supervision signal. (b) Shows the operation process of MACA. (c) presents the three distillation objectives.

ables complementary learning of reasoning generation, evaluation and correction.

Definition. Formally, we denote the student model as π_θ, the input problem as q, and the reasoning trajectory as a sequence of steps $s = (s_1, s_2, \ldots, s_k)$. At each step t, the model generates a reasoning step s_t conditioned on the input and the verified context C_{t-1}. Below, we introduce three step-level training objectives for structurally distilling student model.

Objective 1: Learning Acceptable Reasoning Trajectories. Learning from validated reasoning trajectories provides effective supervision for improving the reasoning ability of small language models. In ICCDi, the reasoning steps generated by the student model are first filtered through MACA. The accepted steps, combined with the additional steps introduced during teacher intervention, form the acceptable reasoning trajectories s^+. We fine-tune the student model on these trajectories using a standard next-step prediction objective. The loss is defined as follows:

$$\mathcal{L}_1 = -\sum_{(q,s^+)\in\mathcal{D}^+} \sum_{t=1}^{|s^+|} \log \pi_\theta(s_t^+ | q, C_{t-1}) \quad (2)$$

where $\mathcal{D}^+$ denotes the set of acceptable reasoning trajectories constructed through multi-agent consensus.

Objective 2: Learning Step-Level Reliability. Beyond generating valid reasoning steps, the student model is trained to assess the reliability of each reasoning step. During the MACA process, each generated step s_t is associated with a consensus evaluation signal Φ_t, produced by multiple teacher agents, which reflects the reliability of the step under the context C_{t-1}. To learn this capability, we augment the student model π_θ with a lightweight value head that predicts a reliability score $\hat{\Phi}_t$ for each step. The model is trained to minimize the discrepancy between the predicted score and the consensus evaluation signal:

$$\mathcal{L}_2 = \frac{1}{|\mathcal{S}|} \sum_{(q,t)\in\mathcal{S}} (\hat{\Phi}_t - \Phi_t)^2 \quad (3)$$

where $\mathcal{S}$ denotes the set of all audited reasoning steps with their associated contexts and evaluation signals. The construction of Φ_t is described in Appendix A.2.

Objective 3: Learning Decision Preferences at Branching Points. This objective encourages the model to favor corrected reasoning steps over erroneous alternatives under the same reasoning context, thereby improving local decision quality at critical branching points. When a reasoning step

s_k is rejected by consensus auditing, ICCDi constructs a contrastive decision pair under the same verified prefix $s_{<k} = (s_1, \ldots, s_{k-1})$. Specifically, we define:

- **Preferred step:** $s_{\text{win}} = \hat{s}_k$, the teacher-provided corrective step;
- **Non-preferred step:** $s_{\text{lose}} = s_k$, the student-generated step rejected by consensus auditing.

All such samples form a step-level preference dataset $\mathcal{D}_{\text{step}} = \{(q, s_{<k}, s_{\text{win}}, s_{\text{lose}})\}$. The student model is trained to prefer corrected steps over rejected ones under identical context conditions. Following prior work on preference optimization (Lai et al., 2024; Lu et al., 2024b), we adopt a DPO-style objective to improve local decision quality:

$$\mathcal{L}_3 = -\mathbb{E}_{(q,s_{<k},s_{\text{win}},s_{\text{lose}})\sim\mathcal{D}_{\text{step}}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(s_{\text{win}} \mid q, s_{<k})}{\pi_{\text{ref}}(s_{\text{win}} \mid q, s_{<k})} - \beta \log \frac{\pi_\theta(s_{\text{lose}} \mid q, s_{<k})}{\pi_{\text{ref}}(s_{\text{lose}} \mid q, s_{<k})} \right) \right] \tag{4}$$

where π_{ref} is a reference policy (e.g., a frozen copy of the initial model), and β controls the strength of preference optimization.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on five challenging reasoning benchmarks that require multi-step logical deduction in scientific domains. These include: (i) **MATH** (Hendrycks et al., 2021), a competition-level mathematics dataset covering algebra, geometry, and calculus; and four subsets from MMLU-Pro (Wang et al., 2024b): (ii) **Physics**, (iii) **Chemistry**, (iv) **Biology**, and (v) **Computer Science**. All datasets require not only correct final answers but also coherent intermediate reasoning.

Baselines. We compare ICCDi with three strong baselines: **(i) No Distillation**: the base student model uses standard chain-of-thought prompts (Wu et al., 2023a) for reasoning. **(ii) Single-Teacher Distillation**: the distilled student model using supervision fine-tuning (SFT) with knowledge from a single teacher. **(iii) Multi-Teacher Structured Distillation**: the model distilled using the current advanced methods, including MAGDi (Chen et al., 2024b), which models multi-teacher interactions through graph structures and extracts distillation signals; D&R (Zhou et al., 2025), which constructs contrastive decision signals through debate and reflection among teachers and trains using tree-structured preference optimization.

Evaluation Metrics. We report two standard metrics to evaluate model performance: **(i) Accuracy (Acc)**: the percentage of test instances for which the final answer matches the ground-truth solution, measuring overall reasoning correctness; **(ii) Total Token Usage per Example**: the total number of input and output tokens consumed during inference, used to assess reasoning efficiency and the computational cost of different methods.

Implement Details. We use Qwen2.5-3B-Instruct (Hui et al., 2024) as the base student model. The teacher include three large language models: DeepSeek-V3-0324 (DeepSeek-AI et al., 2025), Qwen2.5-Math-72B-Instruct (Yang et al., 2024), and Kimi-K2 (Team et al., 2026). We also evaluated the performance of different teachers across various tasks (see Appendix A.1 for results). By default, we employed a temperature of 0.0 to ensure deterministic outputs, with a maximum sequence length of 512 tokens. Detailed information on the prompt templates is presented in the extended version (see Appendix B.2).

Configurations. For student reasoning, we employed a temperature of 0.3 to expose diverse failure modes. The maximum number of inference steps was set to 8, with each trajectory permitted up to 4 correct attempts ($K = 4$). Appendix A.3 details the specific configuration of K. All experiments were conducted on a server equipped with 8 NVIDIA RTX 4090 GPUs (24GB memory per GPU).

4.2 Main Results

We evaluated the accuracy of ICCDi on MATH and MMLU-Pro (covering four subfields: physics, chemistry, biology, and computer science), comparing it against three baseline approaches. To mitigate potential bias introduced by any single model instance, all reported results are averages of five independent runs conducted under each experimental setting.

Advantages over no distillation and single-teacher distillation. As shown in table 1, ICCDi

Dataset	*No Distillation*	*Single-Teacher Distillation*			*Multi-Teacher Distillation*		
	Qwen2.5-3B	**DeepSeek-v3**	**Qwen2.5**	**Kimi**	**MAGDi**	**D&R**	**ICCDi**
MATH	27.68	28.47	28.12	27.79	30.41	32.17	**36.43**
MMLUPro-Chem	33.60	34.33	33.21	33.70	36.71	38.24	**42.46**
MMLUPro-Phys	36.49	37.12	36.49	35.41	41.89	44.21	**46.97**
MMLUPro-Comp	43.07	44.89	43.43	44.24	56.71	**62.41**	61.50
MMLUPro-Biol	60.67	61.46	61.40	59.71	62.54	63.43	**68.81**
Average	40.30	41.25	40.53	40.17	45.65	48.09	**51.23**

Table 1: Accuracy comparison of ICCDi with different distillation baselines across various datasets. Based on our teacher settings, we have reimplemented them using the official codes released by MAGDi (Chen et al., 2024b) and D&R (Zhou et al., 2025). Bold figures denote the best results.

achieves an average accuracy of 51.23% across five datasets, significantly outperforming both the no distillation (40.30%) and single-teacher distillation (40.17%-41.25%). It suggests that relying solely on CoT prompting or imitating a single teacher's reasoning is insufficient for improving small model performance on complex reasoning tasks. In particular, the no distillation baseline performs poorly (e.g., 27.68% on MATH), reflecting the intrinsic limit of the base model. While single-teacher distillation provides modest gains (up to 41.25%), performance varies notably across teachers, indicating sensitivity to individual biases and errors. In contrast, ICCDi surpasses the strongest single-teacher baseline by +8.78%, demonstrating that multi-teacher, step-level supervision yields more stable learning signals.

Advantages over Multi-Teacher Structured Distillation. ICCDi also outperforms existing multi-teacher methods, exceeding MAGDi and D&R by +5.58% and +3.14% on average, respectively. The advantage is consistent across domains, including Physics (46.97%), Chemistry (42.46%, +5.75% over MAGDi), and Biology (68.81%). These improvements not only come from the introduction of multiple teachers, but also depend on ICCDi's incremental, step-level reasoning supervision. Unlike MAGDi and D&R, which learn complete trajectories or interactions, ICCDi performs real-time consensus auditing and correction during generation, effectively preventing error accumulation in long reasoning chainsespecially critical for weaker student models.

4.3 Detailed Analysis

4.3.1 Impact of Different Distillation Objectives

We analyze the contribution of each objective in ICCDi. As shown in table 2, supervised fine-tuning on validated step-level trajectories (SFT) already provides a strong foundation, improving performance by +5.13% over the base model on MATH. It demonstrates that learning from consensus-validated reasoning trajectories is essential for establishing basic reasoning structure.

Setting	MATH	MMLUPro (Avg)
Qwen2.5-3B	27.68	43.46
w/o R, S-DPO	32.81[↑ **5.13**]	51.23[↑ **7.78**]
w/o S-DPO	33.87[↑ 1.06]	52.08[↑ 0.85]
ICCDi	**36.43**[↑ 2.56]	**54.94**[↑ 2.86]

Table 2: Accuracy of ICCDi on MATH and average accuracy on MMLUPro under different distillation objectives. R and S-DPO correspond to our second and third distillation objectives, respectively.

Building on it, incorporating step-level reliability learning further improves performance to 33.87% on MATH and 52.08% on MMLU-Pro, indicating that explicit modeling of reasoning state confidence provides additional supervision beyond trajectory imitation.

We then introduce step-level preference alignment (S-DPO), which leverages contrastive decision pairs constructed at critical branching points. This objective yields consistent gains (reaching 36.43% on MATH and 54.94% on MMLU-Pro), particularly on tasks with frequent local decision errors, highlighting its effectiveness in correcting fine-grained reasoning mistakes that trajectory-level supervision cannot fully address.

Overall, the best performance is achieved when all three objectives are jointly optimized. Each objective targets distinct aspects of reasoning, and their combination is crucial for fully leveraging these step-level supervision signals.

Setting	MATH	MMLUPro (Avg)
w/o Qwen, Kimi	34.43	52.42
w/o Deepseek, Kimi	32.06	51.19
w/o Deepseek, Qwen	33.97	49.86
ICCDi	**36.43**	**54.94**

Table 3: Performance of ICCDi (*Isolated Multi-Teacher Consensus Audit*) using a single teacher agent across different datasets.

Method	MATH		MMLUPro	
	Acc(%)	#Token	Acc(%)	#Token
No Distillation				
Qwen2.5-3B	27.68	1013.77	43.46	822.89
Single-Teacher Distillation				
DeepSeek	28.47	997.43	44.45	776.66
Multi-Teacher Distillation				
MAGDi	30.41	877.51	49.46	707.15
D&R	32.17	804.31	52.07	707.89
ICCDi	36.43	835.52	54.94	692.54

Table 4: Comparison of total tokens consumed per sample (*#Token*) during reasoning between ICCDi and different distillation baseline methods across various test sets.

4.3.2 Multi-teacher Enhances Distillation Effectiveness

We construct a single-teacher ICCDi variant that retains incremental generation, truncation, and repair, but replaces multi-teacher consensus with a single teacher agent for step auditing and correction (using DeepSeek, Qwen, or Kimi). This setting isolates the contribution of multi-agent consensus. As shown in table 3, multi-teacher ICCDi consistently outperforms its single-teacher counterparts across all benchmarks, achieving +2.00% on MATH and +2.52% on MMLU-Pro on average.

These results indicate that the gains of ICCDi are not merely due to stronger individual teacher agent, but arise from the consensus-based auditing mechanism itself. By improving the reliability of step-level supervision while maintaining sufficient coverage, multi-agent consensus enables the construction of higher-quality distilled signals, leading to more robust student reasoning performance.

4.3.3 Token Efficiency Analysis

We analyze the token utilization efficiency of ICCDi to evaluate the trade-off between performance and inference cost. Table 4 reports average accuracy and tokens per sample on MATH and MMLUPro.

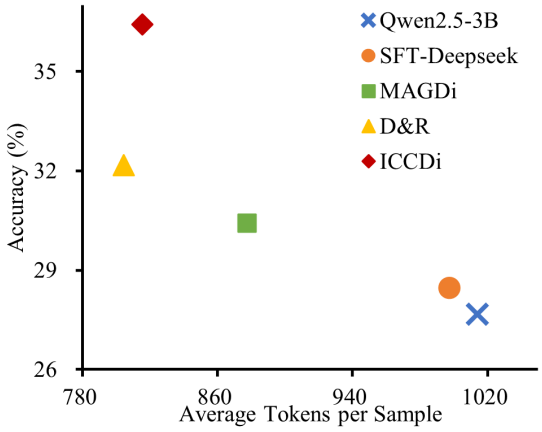


Figure 3: Trade-off between Performance and Token Consumption on MATH, ICCDi achieves nearly equivalent token consumption to the strongest baseline D&R while maintaining the highest accuracy.

On MATH, the no distillation baseline consumes an average of 1013.77 tokens per sample with 27.68% accuracy, indicating inefficient long reasoning chains without effective supervision. Single-teacher distillation provides only marginal gains, improving accuracy to 28.47% while slightly reducing token usage. However, due to the lack of fine-grained control over intermediate errors, reasoning remains verbose and inefficient. A similar trend is observed on MMLUPro.

In contrast, multi-teacher methods achieve better performance-efficiency trade-offs. MAGDi reaches 30.41% accuracy on MATH with 877.51 tokens, while D&R further improves to 32.17% with 804.31 tokens, demonstrating the benefit of structured multi-teacher interactions in reducing redundant reasoning. ICCDi achieves the best balance: on MATH, it reaches 36.43% accuracy with 835.52 tokensonly 31 tokens more than D&R, yet delivering a +4.26% gain. As illustrated in fig. 3, ICCDi lies on the *Pareto frontier* of the performance-efficiency trade-off: no method achieves higher accuracy with equal or fewer tokens.

5 Conclusion

In this work, we presented ICCDi, an incremental distillation framework that transfers reasoning capabilities to small language models through multi-teacher process supervision. Experimental results on mathematical and scientific reasoning benchmarks demonstrate that ICCDi consistently outperforms both single-teacher and exist-

ing multi-teacher distillation methods, while maintaining strong token efficiency. Overall, our results suggest that effective reasoning distillation benefits from aligning supervision with the student model's reasoning process, rather than relying solely on final outputs or static reasoning traces. This process-centric perspective provides a promising direction for improving the scalability and reliability of reasoning in language models.

Limitations

There are important limitations that warrant consideration. First, our study primarily focuses on mathematics and other logic-intensive reasoning benchmarks, and the generalizability of ICCDi to broader NLP tasks or real-world knowledge-intensive applications remains an open question. Second, although our framework emphasizes process-level reasoning by shifting supervision from final answers to step-by-step reasoning paths, the validity of these paths is entirely dependent on the subjective judgment of large teacher models, and the process itself has not been independently verified. These limitations highlight the need for further research to extend ICCDi's applicability and to establish more reliable validation mechanisms for step-level supervision.

References

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

Justin Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. 2024b. Magdi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *International Conference on Machine Learning*, pages 7220–7235. PMLR.

Pei Chen, Shuai Zhang, and Boran Han. 2024c. Comm: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1720–1738.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2025. Capture the key in reasoning to enhance CoT distillation generalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 441–465, Vienna, Austria. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *arXiv*, arXiv:2412.19437. Version 1.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2026. Minillm: On-policy distillation of large language models. *arXiv*, cs.CL/2306.08543. Version 5.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: a survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv*, cs.LG/2103.03874. Version 1.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv*, cs.LG/1503.02531. Version 1.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Chan-Jan Hsu, Davide Buffelli, Jamie McGowan, Feng-Ting Liao, Yi-Chang Chen, Sattar Vakili, and Da shan Shiu. 2025. Group think: Multiple concurrent reasoning agents collaborating at token level granularity. *arXiv*, cs.AI/2505.11107. Version 1.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. *arXiv*, cs.CL/2409.12186. Version 1.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv*, cs.LG/2406.18629. Version 1.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and

Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv*, cs.LG/2305.20050. Version 1.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024a. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico. Association for Computational Linguistics.

Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024b. Step-controlled dpo: Leveraging stepwise error for enhanced mathematical reasoning. *arXiv*, cs.CL/2407.00782. Version 1.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *arXiv*, cs.CL/2311.11045. Version 1.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *arXiv*, cs.AI/2407.18219. Version 1.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, cs.CL/2402.03300. Version 1.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2024. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv*, cs.AI/2310.00280. Version 1.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, and 181 others. 2026. Kimi k2: Open agentic intelligence. *arXiv*, cs.AI/2507.20534. Version 1.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv*, cs.LG/2211.14275. Version 1.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024c. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 257–279, Mexico City, Mexico. Association for Computational Linguistics.

Dingjun Wu, Jing Zhang, and Xinmei Huang. 2023a. Chain of thought prompting elicits knowledge augmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6519–6534, Toronto, Canada. Association for Computational Linguistics.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun

Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023b. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv*, cs.AI/2308.08155. Version 1.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590, Singapore. Association for Computational Linguistics.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv*, cs.CL/2402.13116. Version 1.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv*, cs.CL/2505.09388. Version 1.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv*, cs.CL/2409.12122. Version 1.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. 2024b. Exploring collaboration mechanisms for LLM agents: A social psychology view. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14544–14607, Bangkok, Thailand. Association for Computational Linguistics.

Xiaofeng Zhou, Heyan Huang, and Lizi Liao. 2025. Debate, reflect, and distill: Multi-agent feedback with tree-structured preference optimization for efficient language model enhancement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9122–9137, Vienna, Austria. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Additional Analysis

A.1 Analysis of Teacher Agents Performance

Within the ICCDi framework, multi-agent consensus auditing serves as the core mechanism for process supervision. A natural concern is whether the teacher agents themselves provide sufficiently stable and reliable reasoning to support step-level judgments. To address this, we evaluate the foundational reasoning capability of the teacher agents.

Dataset	Deepseek	Qwen	Kimi
	Avg(1/2/3)	Avg(1/2/3)	Avg(1/2/3)
MATH	70.41	71.46	67.84
MMLUPro-Chem	79.47	78.59	75.43
MMLUPro-Phys	74.57	75.91	75.60
MMLUPro-Comp	76.43	77.89	76.58
MMLUPro-Biol	86.13	84.17	85.43

Table 5: Results from different teacher agent using the same settings as the student model across various datasets.

Specifically, each teacher agent is evaluated over three independent runs on all datasets, using the same input format and prompting strategy as the student. As shown in table 5, all teacher agents achieve over 70% average final-answer accuracy across datasets, with minimal variance across runs, indicating stable reasoning performance.

These results confirm that the selected teacher agents provide sufficiently reliable supervision signals, making them suitable as reference sources for step-level auditing in ICCDi.

A.2 Robustness under Data Scaling and Student Variants

To further validate the stability and generalizability of ICCDi under varying training conditions, we examine the impact of training data scale and student model choice on distillation effectiveness using the MATH dataset. Specifically, we vary the training set size (10%, 25%, 50%, 100%) and apply ICCDi to additional student models, including Qwen2.5-7B and LLaMA-3.1-8B. As shown in the fig. 4, ICCDi exhibits consistent performance gains with increasing data scale and maintains stable improvements across different student models, demonstrating its robustness under varying training conditions.

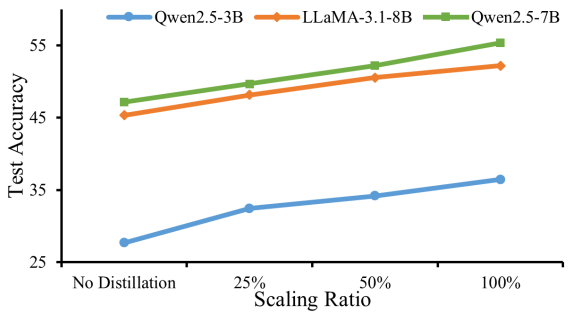


Figure 4: Distillation Performance of Different Student Model Variants at Different Proportions of Training Data Size (On MATH).

A.3 The Balance between *K* and Distillation Effectiveness

A key design in ICCDi is the parameter K, which controls the maximum number of errors the student model can make before teacher agent takeover. Results on MATH are shown in table 6.

Value K	$K=1$	$K=2$	$K=4$	$K=6$
Qwen2.5-3B	-	-	27.68	-
ICCDi	29.14	31.27	**36.43**	35.34

Table 6: Effect of *K* Value Settings on Distillation Efficiency.

When K is small, the student is terminated early after only a few errors. Under this setting, supervision signals are dominated by teacher-generated steps, leading to stable but limited learning. The performance remains relatively low (e.g., 29.14 at $K = 1$, 31.27 at $K = 2$), indicating insufficient exposure to diverse student error states.

As K increases, the student is allowed to explore deeper reasoning trajectories, generating more step-level contrastive samples through error-correction pairs. This significantly improves performance, peaking at 36.43 when $K = 4$. The results suggest that richer exposure to intermediate error states enhances the effectiveness of step-level supervision. However, when K becomes too large, performance slightly declines (35.34 at $K = 6$). In this setting, the student may remain in unstable reasoning states for extended steps, producing low-quality or misaligned trajectories.

Overall, an appropriate K can achieve the best balance between student exploration and supervision quality. It enables sufficient generation of informative error-correction pairs while avoiding excessive noise and computational overhead, leading to optimal distillation performance.

B More implementation details of ICCDi

B.1 Details of Multi-Agent Consensus Auditing

This section details the multi-agent consensus auditing protocol in ICCDi.

Input and Evaluation. At step t, the student generates a candidate step s_t given context C_{t-1}. Each agent receives the same input (Q, C_{t-1}, s_t) and outputs a discrete judgment judgment: $v_i \in \{P, N, Neutral\}$, where:

- P (Positive): The step is acceptable;
- N (Negative): The step is incorrect or degenerate and should be rejected;
- $Neutral$: The step provides limited progress (e.g., redundant or cyclic) and is not recommended.

Neutral does not indicate a logical error but lowers the likelihood of acceptance.

Consensus Score and Passing Rule. Agent's judgments are aggregated into a consensus score:

$$\Phi_t = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}(v_i = P) \tag{5}$$

where $\Phi_t \in [0, 1]$ represents the proportion of teacher agents supporting the step. Only Positive votes contribute to Φ_t. A step is accepted if $\Phi_t \geq \tau$; otherwise, it is rejected and triggers teacher agents correction. In our default setting, $n = 3$ and $\tau = 0.66$, requiring at least two teachers to approve the step.

Dual Purpose of Φ_t. The consensus score serves two purposes: **(1)** Process control during inference, determining whether a step is accepted or corrected; **(2)** Supervision signal during distillation, where Φ_t is used to train the student's value head for step-level reliability estimation.

Boundary clarification. Φ_t is not treated as an absolute measure of correctness, but as a consensus-based approximation of step acceptability under the current context.

B.2 Prompt Templates for Different Tasks

Figure 5 shows the design of prompt templates used for different stages in the ICCDi framework.

B.3 Detailed Examples of MATH dataset

We present detailed example from MATH in fig. 6.

Student – Initial Step

You are a student reasoning model. Your task is to solve the following problem step by step. You must begin by producing ONLY the first reasoning step.
Problem: {Q}
Instructions:
- Generate exactly ONE initial reasoning step.
- Do NOT generate multiple steps.
- Do NOT include explanations outside the step.
Output format:<step>...</step>

Student Step-wise Generation

You are a student reasoning model. Continue solving the problem step by step.
Problem: {Q}
Verified reasoning so far: {C_{t-1}}
Instructions:
- Generate exactly ONE next reasoning step.
- Do NOT restate previous steps.
- Do NOT anticipate future steps.
- If you have the final answer, you can output the answer inside <answer>...</answer>.
Output format:<step>...</step>

Teacher – Step Audit

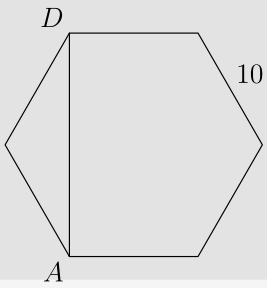
You are an expert auditor evaluating a student's reasoning step.
Problem: {Q}
Verified reasoning so far: {C_{t-1}}
Student step:<step>{s_t}</step>
Your tasks:
- Judge whether the step is logically valid and helpful.
  Choose ONE label: [Positive] or [Negative].
- If you choose [Negative], you MUST provide a corrected version of this step that would be valid under the same context.
Output format:<label>...</label>
<repair>...</repair>

Teacher Takeover

You are an expert reasoning model. The student model is unable to complete the reasoning reliably. You are asked to take over and finish the solution.
Problem: {Q}
Verified reasoning so far: {C_t}
Instructions:
- Continue the reasoning from the given prefix.
- Provide a complete and coherent reasoning chain.
- End with the final answer.
Output format:<step>...</step>
<answer>...</answer>

Figure 5: The Prompt Templates for Different Tasks.

Question： Find the number of units in the length of diagonal DA of the regular hexagon shown. Express your answer in simplest radical form. [asy]
size(120);
draw((1,0)--(3,0)--(4,1.732)--(3,3.464)--(1,3.464)--(0,1.732)--cycle);
draw((1,0)--(1,3.464));
label("10",3.5,2.598),NE);
label("A",(1,0),SW);
label("D",(1,3.464),NW); [/asy]



gold_answer "$10\sqrt{3}$"

Student Output：<step>Because the hexagon is regular, each side length is 10.</step>
Teacher 1：<label>Positive</label>
Teacher 2：<label>Positive</label>
Teacher 3：<label>Positive</label>
Step1:Because the hexagon is regular, each side length is 10.
Student Output：<step>Since A and D are opposite vertices of a regular hexagon, DA equals 2 times the side length, so $DA = 2 \times 10 = 20$.</step>
Teacher 1：<label>Negative</label><repair>A and D in the diagram form the short diagonal (skipping one vertex), which can be obtained as the altitude of an equilateral triangle with side 10. </repair>
Teacher 2：<label>Negative</label><repair>In a regular hexagon, connecting vertices with one vertex between yields a short diagonal of length side $\cdot \sqrt{3}$; here DA is such a diagonal. </repair>
Teacher 3：<label>Negative</label><repair>DA is not the longest diagonal; it corresponds to a two-side turn and has length $10\sqrt{3}$.</repair>
Step2:Diagonal DA connects opposite vertices of a regular hexagon, so $DA = 2 \times 10 = 20$.
Step2:In a regular hexagon, connecting vertices with one vertex between yields a short diagonal of length side $\cdot \sqrt{3}$.
Student Output：<step>Thus $DA = 10 \cdot \sqrt{3} = 10\sqrt{3}$.</step><answer>$10\sqrt{3}$</answer>
Teacher 1：<label>Positive</label>
Teacher 2：<label>Positive</label>
Teacher 3：<label>Positive</label>
Step3:Thus $DA = 10 \cdot \sqrt{3} = 10\sqrt{3}$.
Finaly answer: $10\sqrt{3}$

Figure 6: A detailed example from the MATH dataset.