Rethinking the Generation of High-Quality CoT Data from the Perspective of LLM-Adaptive Question Difficulty Grading

Anonymous ACL submission

Abstract

Recently, DeepSeek-R1 (671B) (DeepSeek-AI et al., 2025) has demonstrated its excellent reasoning ability in complex tasks and has publicly shared its methodology. This provides potentially high-quality chain-of-thought (CoT) data for stimulating the reasoning abilities of small-sized large language models (LLMs). To generate high-quality CoT data for different LLMs, we seek an efficient method for generating high-quality CoT data with LLM-Adaptive question difficulty levels. First, we grade the difficulty of the questions according to the reasoning ability of the LLMs themselves and construct an LLM-Adaptive question database. Second, we sample the problem database based on a distribution of difficulty levels of the questions and then use DeepSeek-R1 (671B) (DeepSeek-AI et al., 2025) to generate the corresponding high-quality CoT data with correct answers. Thanks to the construction of CoT data with LLM-Adaptive difficulty levels, we have significantly reduced the cost of data generation and enhanced the efficiency of model supervised fine-tuning (SFT). Finally, we have validated the effectiveness and generalizability of the proposed method in the fields of complex mathematical competitions and code generation tasks. Notably, with only 2k high-quality mathematical CoT data, our ZMath-32B surpasses DeepSeek-Distill-32B in math reasoning task. Similarly, with only 2k high-quality code CoT data, our ZCode-32B surpasses DeepSeek-Distill-32B in code reasoning tasks. Our ZMath-32B LLM also outperforms both the DeepSeek-Distill-32B and QwQ-32B models on general evaluation benchmarks. Our data and LLMs will be open-sourced in the future.

1 Introduction

Since the release of DeepSeek-R1 (DeepSeek-AI et al., 2025), long chain-of-thought reasoning has gained widespread popularity in both foundational AI LLMs and a wide range of industrial AI applications. However, the deployment of full-capacity R1-class models (e.g., DeepSeek-R1 with 671B parameters) poses substantial computational challenges, rendering their utilization infeasible for edge devices and real-time systems due to prohibitive resource demands. This limitation has spurred intensive research into developing compact (<70B parameters) models capable of sustaining extended CoT reasoning, which is a core competency requirement for mathematical problem-solving, code generation, and scientific analysis. Thanks to the shared reasoning process of DeepSeek-R1, we can get high-quality CoT data to boost the reasoning abilities of small-parameter LLMs.

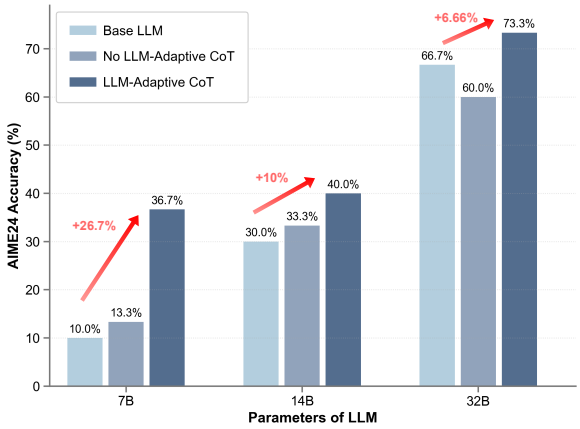

Figure 1: Constrcution of CoT Data with/without LLM-Adaptive question difficulty grading. For LLms of different parameters, the former consistently outperforms the latter in reasoning performance on the mathematical competition dataset AIME24 (of America, 2024).

Recently, many methods for generating CoT data based on DeepSeek-R1 have been widely studied in the community. (Labs, 2025; Team, 2025c) enhance the reasoning capabilities of LLMs by using massive CoT data, enabling their reasoning abilities to reach competitive levels. (Ye et al., 2025; Muennighoff et al., 2025) aim to trigger the reasoning capabilities of large models by constructing a small batch of high-quality CoT data, yet they are unable to achieve further improvements in rea-

soning performance. (Wen et al., 2025a) focuses on refining reasoning abilities through multi-stage curriculum learning and rejection sampling. However, these approaches rarely consider the adaptive relationship between the Base LLM and its training data during data distillation. Therefore, we rethink the question:**"What constitutes high-quality CoT data?"** and try to provide a comprehensive answer **from the perspective of LLM-Adaptive Question Difficulty Grading**.

Based on the above discussion, we propose a method for constructing high-quality CoT data based on LLM-Adaptive Question Difficulty Grading, as shown in Figure 1. Our method efficiently creates LLM-Adaptive CoT datasets, significantly enhancing reasoning abilities of LLMs across varying parameters without requiring resource-intensive fine-tuning approaches such as curriculum learning or rejection sampling. In contrast, LLMs trained on data without adaptive difficulty grading struggle to improve or may experience degraded performance under the same cost constraints. First, we evaluate and grade the difficulty levels of the reasoning questions by analyzing the intrinsic reasoning capabilities of the LLMs. Based on this adaptive difficulty grading, we develop an adaptive question database that covers various difficulty levels. Next, we sample questions from this adaptive library, guided by a carefully designed distribution across different levels of difficulty. Finally, utilizing the powerful reasoning capabilities of the DeepSeek-R1 (671B) (DeepSeek-AI et al., 2025), we generate corresponding high-quality CoT data that covers both mathematical reasoning and code generation tasks.

In summary, the main contributions of this work are as follows:

Adaptive Difficulty Evaluation: We analyze the intrinsic reasoning capabilities of LLMs to effectively evaluate and classify reasoning questions into adaptive difficulty levels.

Comprehensive Adaptive Problem Library: Based on the adaptive difficulty levels, we construct an extensive problem library covering diverse difficulty categories and carefully sample questions according to a well-designed difficulty distribution.

High-Quality CoT Data Generation: Leveraging the DeepSeek-R1 model (671B) (DeepSeek-AI et al., 2025), we generate high-quality chain-of-thought (CoT) datasets that cover mathematical reasoning and code generation tasks, ensuring consistent accuracy and detailed reasoning.

Comprehensive Evaluation: We conduct extensive experiments on mathematical reasoning and code generation tasks using LLMs with different parameters, demonstrating the effectiveness and generalization of our proposed method for high-quality chain-of-thought (CoT) data generation.

2 Related Works

2.1 Chain-of-Thought (CoT) Data Generation

Current research focuses on three primary strategies for generating high-quality CoT data: (1) Manual annotation by domain experts to create gold-standard reasoning chains, primarily for benchmarking (Li et al., 2024; Huang et al., 2024; Gao et al., 2024); (2) Prompt engineering leveraging LLMs' in-context learning capacity to elicit step-by-step rationales, though constrained by model biases (Wu et al., 2024; Maiti et al., 2025; Whitney et al., 2024); (3) Automated generation using self-alignment frameworks (Mahene et al., 2024; Liu et al., 2025). While such methods show promise, particularly in boosting small sized LLMs via supervised fine-tuning, key challenges persist in ensuring the diversity, correctness, and coherence of generated reasoning chains (Muennighoff et al., 2025; Ye et al., 2025). To address these limitations, recent advances integrate rejection sampling to filter low-quality reasoning paths and employ iterative refinement of teacher models. For instance, some approaches (Labs, 2025; Team, 2025c) leverage DeepSeek-R1 as the teacher reasoning model to improve step-by-step rationale generation, coupled with GPT-4o-mini for mathematical solution verification. Despite these improvements, scaling high-quality CoT generation across broader domains and difficulty levels remains an open challenge, particularly in maintaining robustness against error propagation in multi-step reasoning.

2.2 LLM-Adaptive Difficulty Grading

Traditional data generation approaches typically rely on static difficulty labels or heuristic rules, which inadequately account for the continuously evolving capabilities of large language models (LLMs). Inspired by adaptive assessment techniques in educational settings, this strategy automatically calibrates training data to align with the model's current competence, thereby optimizing learning efficiency. Prior studies have explored alternative methods, such as employing LLM-generated scoring to adjust difficulty (Team, 2025a;

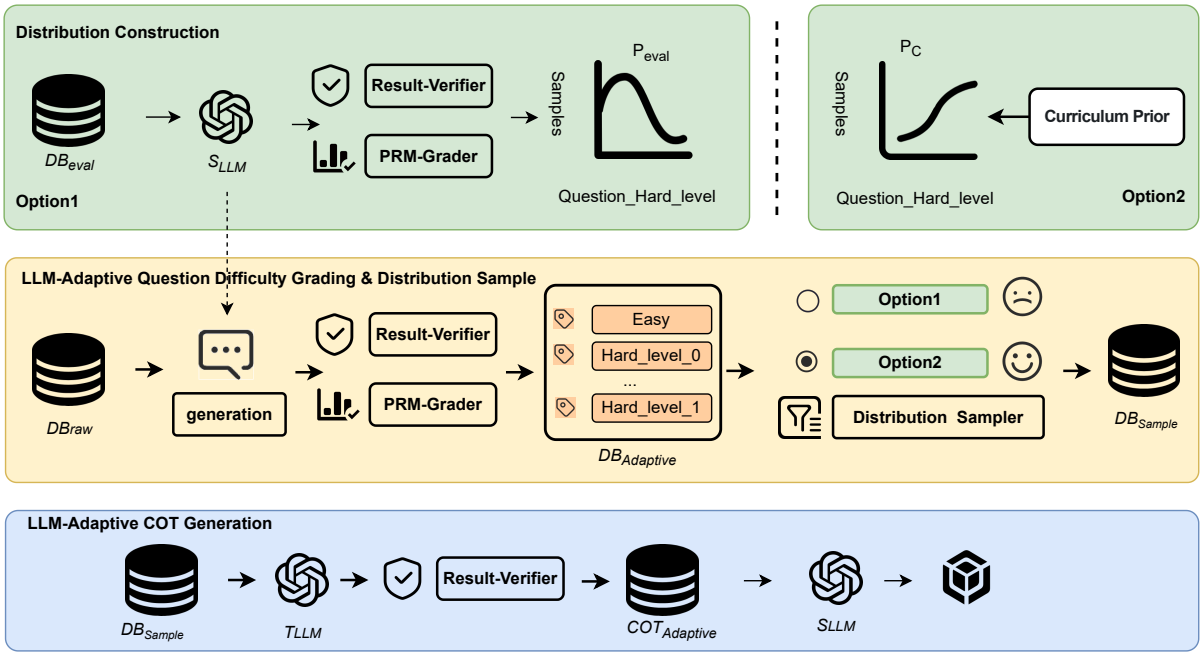

Figure 2: The Framework for CoT Data Generation via LLM-Adaptive Question Difficulty Grading , comprising three core components: **Distribution Construction**, **LLM-Adaptive Question Difficulty Grading & Distribution Sampling**, and **LLM-Adaptive Chain-of-Thought (CoT) Generation**

Xie et al., 2024; Lee and Song, 2024) or adopting curriculum learning frameworks (Wen et al., 2025a; Min et al., 2024; Yuan et al., 2025) that treat long-form QA as inherently challenging tasks. However, these approaches suffer from critical limitations, including inaccurate difficulty categorization and insufficient granularity in difficulty stratification. For instance, coarse-grained curriculum designs often oversimplify difficulty levels (e.g., categorizing questions merely by length), while LLM-based scoring methods struggle to capture nuanced reasoning demands. Such shortcomings highlight the need for more sophisticated, fine-grained adaptive frameworks to bridge the gap between data difficulty and model capability.

3 Methods

In this section, we introduce in detail our method for constructing high-quality CoT data with LLM-Adaptive question difficulty grading. As shown in Figure 2, our approach contains three components, described separately in the following subsections: (1) **Distribution Construction**, (2) **LLM-Adaptive Question Difficulty Grading & Distribution Sampling**, and (3) **LLM-Adaptive CoT Generation**.

3.1 Distribution Construction

To efficiently sample questions with model-adaptive difficulty grading, we require an effective reference distribution. To this end, we propose two alternative approaches for constructing a question-difficulty distribution. **Option1** leverages the Base LLM (defined as S_{LLM}) to characterize the true difficulty-level distribution (defined as P_{eval}) over the evaluation datasets. **Option2** employs a customized distribution (defined as P_C) based on human-defined priors. Detailed descriptions of both methods are presented below.

Option1 To obtain the actual difficulty-level distribution P_{eval} from the S_{LLM} for the evaluation data DB_{eval}, we first perform answer verification through a Result-Verifier. Those questions correctly answered by S_{LLM} are defined as easy-level problems. Then, we utilize a PRM-Grader to grade the difficulty levels of the questions that the S_{LLM} answers incorrectly. The specific grading formulation is shown as follows:

$$P_{eval} = \begin{cases} \text{Easy}, & \text{if answer is True} \\ \text{Grader}(Q, R), & \text{if answer is False} \end{cases} \quad (1)$$

where Grader(Q, R) denotes the Difficulty grading given by the PRM-Grader based on the reponse R of the S_{LLM}. The details of the Result-Verifier and PRM-Grader can be seen in 3.2

Option2 Inspired by the idea of curriculum learning, we hypothesize that during the model fine-

tuning process, the model learns relatively difficult questions more easily compared to very difficult ones. Therefore, we also propose a curriculum-learning-based customized distribution. Specifically, we classify question difficulty into five levels, with the number of samples at each difficulty level decreasing as the difficulty increases. The distribution can be formally defined by the following equation:

$$P_C = \frac{N_i}{N_{\text{total}}} = \frac{w_i}{\sum_{j=1}^{5} w_j}, \quad w_i > w_{i+1}, \quad i = 1, 2, \ldots, 4 \tag{2}$$

where N_i denotes the number of questions at difficulty level i, N_{total} denotes the total number of questions, and w_i represents the weight assigned to difficulty level i. The constraint $w_i > w_{i+1}$ illustrates that the assigned number of samples decreases as question difficulty increases.

We propose two methods for constructing model-adaptive difficulty distributions: Option1 derives difficulty levels from the actual performance of the S_{LLM}, while Option2 uses a curriculum-learning-inspired human-defined distribution. Subsequent experiments in section 4 will analyze and compare these approaches in detail. Next, we will use the distributions to guide the selection of LLM-Adaptive Questions.

3.2 LLM-Adaptive Question Difficulty Grading & Distribution Sampling

After constructing the question difficulty distribution based on either the evaluation-set distribution or curriculum-learning-inspired principles, we further need to build a specialized candidate question database from a large-scale data space using the model-adaptive difficulty grading method. Following such construction, we can perform sampling according to the predefined difficulty distribution to obtain the final high-quality questions. First, we illustrate our LLM-Adaptive Question Difficulty Grading method for building the candidate question database (defined as $DB_{Adaptive}$) in detail. Then, we describe the process of Distribution Sample to acquire the LLM-Adaptive Questions.

LLM-Adaptive Question Difficulty Grading First, we collect original questions from large-scale open-source datasets, each accompanied by standardized answers, thus constructing an initial question-answer database (defined as DB_{raw}). Next, we generate responses for these questions using the S_{LLM} and record their reply trajectories and results. Then, we apply the Result-Verifier customized to specific tasks. For mathematical reasoning, we adopt a Math Verifier, directly comparing the LLM-generated and standard answers; for code generation tasks, correctness is verified by executing the produced code against a suite of test cases, passing all tests indicating correctness. According to equation 1 , verified correct responses are labeled as easy and directly added to the candidate question database. For responses deemed incorrect, we label these questions as difficult and further utilize the PRM-Grader, assigning difficulty levels into five categories. Specifically, the PRM-Grader computes an average score (ranging from 0 to 1) reflecting the response trajectory of S_{LLM}, mapping this score onto five discrete difficulty levels, with lower scores indicating relatively higher difficulty. Let the average score computed by PRM-Grader be denoted as $s \in [0, 1]$, and let the difficulty level be denoted as $L \in \{1, 2, 3, 4, 5\}$, where a lower value of L indicates a higher difficulty. The mapping from s to L is defined as:

$$L = \text{Grader}(Q, R) = f(s) = \lceil 5 \cdot (1 - s) \rceil \tag{3}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Ultimately, the questions categorized by these difficulty levels are collectively included within our candidate question database, thereby completing the construction of $DB_{Adaptive}$ in a LLM-Adaptive manner.

Distribution Sample After constructing the candidate question database, we employ a distribution-based sampler, guided by the question-difficulty distribution established in Section 3.1, to sample high-quality, model-adaptive questions from this database as preliminary inputs for obtaining high-quality CoT data. This procedure is formally defined as follows:

$$DB_{\text{sample}} \sim \text{Sampler}\left(DB_{\text{Adaptive}}, P_{\text{eval}} \vee P_{\text{C}}\right) \tag{4}$$

where DB_{sample} represents the sampled questions, DB_{Adaptive} denotes the candidate question database, and P_{eval}, P_C indicates the difficulty-level distribution determined in Section 3.1.

3.3 LLM-Adaptive CoT Generation

After obtaining the DB_{Sample}, we directly employ the T_{LLM} to generate responses and associated

reasoning processes for these sampled questions. Subsequently, we apply the Result-Verifier to examine and validate the correctness of these generated responses. The implementation of Result-Verifier here is identical to that described in Section 3.2. The T_{LLM} in our experiments is DeepSeek-R1(671B) (DeepSeek-AI et al., 2025). Following the verification process, we select only those questions whose corresponding responses and reasoning processes have been validated as correct, thereby forming a high-quality CoT dataset $COT_{Adaptive}$. Finally, this rigorously-constructed CoT dataset serves as training data for fine-tuning S_{LLM} to get the final reasoning LLM R_{LLM}.

4 Experiment

4.1 Setup

Datasets and Metrics Our training datasets consist of high-quality mathematical reasoning problems sourced from NuminaMath (LI et al., 2024), historical AIME problems (of America, 2024), and OlympicArena (Huang et al., 2024), as well as challenging code generation tasks from TACO (Li et al., 2023) and CodeForces (Penedo et al., 2025).

Benchmarks To give a reasonable result, we evaluate our trained models on the following authoritative benchmarks:

- *AIME24 and AIME25* (of America, 2024) comprise challenging mathematics competition problems from the American Invitational Mathematics Examination of 2024 and 2025.
- *MATH500* (Lightman et al., 2023) is a representative subset of 500 mathematical problems from the comprehensive MATH dataset.
- *GPQA* (Rein et al., 2024) is a dataset focused on graduate-level physics questions designed to evaluate advanced problem-solving skills.
- *LiveCodeBench* (Jain et al., 2024) contains competitive coding problems sourced from various platforms categorized by three difficulty levels.

Settings Our training framework builds on previous advancements in s1-1k(Muennighoff et al., 2025), LIMO(Ye et al., 2025), and Light-R1(Wen et al., 2025b), implemented through the LLama-Factory(Zheng et al., 2024) to leverage its proven scalability. The framework incorporates the Deepseek-R1 template, flash-attention2(Dao, 2024) and Liger-Kernel(Hsu et al., 2024) to improve computational efficiency while minimizing memory requirements. All experiments are conducted on a 2×8 H800 GPU cluster, with performance evaluations executed using the Skythought benchmarking suite(Team, 2025a). The core hyperparameters for the initial experiments included a context length of 16384, a learning rate of 5e-6, a batch size of 128, and 10 training epochs.

Model	MATH			
	MATH 500	*AIME 24*	*AIME 25*	*GPQA*
DS-distill-7B	89.4	56.67	33.3	49.49
ZMath-7B	**93.2**	**60**	**43.33**	49.49
phi4-14B	79.2	30	16.67	54.55
ZMath-14B	**89.4**	**50.0**	**36.67**	**63.13**
DS-distill-32B	89.8	66.67	50.0	59.6
Sky-32B-Preview	90	43.33	23.33	50.0
ZMath-32B	**94.6**	**73.33**	**56.67**	**63.13**

Table 1: Comparison of LLMs with different parameters on Math Reasoning Benchmarks

Model	*LiveCodeBench*		
	EASY	*MEDIUM*	*HARD*
DS-distill-7B	79.21	**41.09**	**11.11**
ZCode-7B	**81.0**	39.58	10.11
phi4-14B	72.4	29.91	5.19
ZCode-14B	**89.96**	**41.99**	**8.89**
DS-distill-32B	92.11	74.92	30
Sky-32B-Preview	84.23	46.53	8.89
ZCode-32B	**96.06**	**75.53**	**31.85**

Table 2: Comparison of LLMs with different parameters on Code Generation Benchmarks

Baselines We take the three representative baselines below for comparison:

- *phi-4*: phi-4(Abdin et al., 2024) is a 14B LLM developed by Microsoft. Phi-4 demonstrates exceptional performance in complex reasoning tasks, particularly in mathematics, where it achieves 80.4% on the MATH benchmark and 80.6% on MGSM.
- *DeepSeek-Distill-R1*: A series of LLMs developed by Deepseek, distilled from the Deepseek-R1 using 800k training instances(DeepSeek-AI et al., 2025). Our implementation primarily utilizes the 7B and 32B variants distilled on the Qwen architecture.

- *Sky-T1-32B-Preview*: This model exhibits characteristics similar to Distill-R1-32B, but was developed by the Sky-T1 team using 10,000 distillation samples that specifically target mathematical and coding capabilities(Team, 2025b).

4.2 Results and Analysis

It should be noted that, due to resource constraints, our evaluation process uses pass@1 measured over 16 runs, and we report the average result.

Main Results Table 1 and Table 2 present the results of controlled experiments evaluating LLM performance on mathematical and coding benchmark datasets, respectively. Synthetic mathematical and coding data were generated using three base LLMs, DS-distill-7B, phi4-14B, and DS-distill-32B, and subjected to supervised fine tuning (SFT) to produce a series of LLMs at 7B, 14B, and 32B parameter scales. These LLMs were grouped into two categories: Zmath for mathematical tasks and Zcode for coding tasks.

The Zmath-14B LLM was trained on approximately 16,000 distilled data points derived from Deepseek-R1. The distilled data set was manually verified to ensure the accuracy of the answers. After training, the LLM output was aligned with the Deepseek-r1 format. Compared to the baseline phi4-14B LLM, Zmath-14B demonstrated substantial improvements: an average gain of 20 points on the AIME (2024, 2025) and Livecode-easy benchmarks; an approximate increase of 10 points on the GPQA and Livecode-medium datasets.Notably, Zmath-14B outperformed the Sky-32B-Preview LLM on mathematical tasks, highlighting its superior capability in this domain.

To explore the upper limit of our adaptive data synthesis approach, we trained LLMs using only 2,000 Chain of Thought (CoT) mathematical and coding data points on the DS-distill-32B. The results were as follows: Zmath-32B achieved an average improvement of 5 points across all mathematical benchmarks, significantly exceeding the performance of DS-distill-32B. On the Livecode Bench, Zcode-32B recorded an average gain of 2.14 points over DS-distill-32B. We hypothesize that further enhancements in coding performance, relative to mathematical gains, may necessitate a larger training corpus. Comparable performance improvements were observed with the smaller Zmath-7B and Zcode-7B models, consistent with the trends observed in their 32B counterparts. These findings underscore the efficacy of our data synthesis method across varying model scales and task domains.

Grading Methods	*LiveCodeBench*		
	EASY	*MEDIUM*	*HARD*
No-Grading	92.11	74.92	30.00
UT-Grading	93.19	71.60	28.15
PRM-Grading	**96.06**	**75.53**	**31.85**

Table 3: Performance (%) of DS-distill-32B trained on 2K data using different difficulty grading methods (None, UT-based, and PRM-based), evaluated on LiveCode-Bench.

Ablation Studies We conducted four ablation studies using the code datasets to comprehensively validate the effectiveness of our proposed LLM-Adaptive difficulty grading approach from various perspectives.

(1) Comparison of Difficulty Grading Methods (PRM vs. UT) This experiment verifies the effectiveness of different difficulty grading methods. We compared two difficulty grading methods: Process Reward Model (PRM)-based grading, which assigns 0-1 scores divided into five levels (lower scores indicate higher difficulty), and Unit Test (UT)-based grading, which categorizes problems into five levels based on the percentage of passed test cases (lower pass rates indicate higher difficulty). As shown in Table 3, we compare the results of DeepSeek-Distill-32B trained with UT-graded 2K data, PRM-graded 2K data, and the baseline evaluated on LiveCodeBench (easy-medium-hard), demonstrating the superior effectiveness of the PRM-based difficulty grading method.

(2) Distribution Transfer Experiment To examine whether different models exhibit unique preferences for difficulty distributions, we perform a distribution transfer experiment. Specifically, we transfer the PRM-based difficulty distribution derived from the DeepSeek-Distill-32B model to train the DeepSeek-Distill-7B model. We then compare its performance with a counterpart trained using the 7B model's own PRM-based distribution. As shown in Table 4, the 7B model benefits more from training on its self-derived difficulty distribution, suggesting that model-specific difficulty adaptation plays a critical role in performance optimization.

CoT Source	MATH			
	MATH 500	*AIME 24*	*AIME 25*	*GPQA*
Baseline-7B	89.40	56.67	33.30	49.49
+CoT from 32B	92.00	56.67	40.00	45.96
+CoT from 7B	**93.20**	**60.00**	**43.33**	**49.49**

Table 4: Performance (%) of 7B LLM with training data distributions derived from 32B and 7B LLMs on math benchmarks. The 7B/32B LLM is DS-distill-7B/32B.

Training Setup	MATH			
	MATH 500	*AIME 24*	*AIME 25*	*GPQA*
No PRM fine-tuning	89.80	66.67	50.00	59.60
+1K PRM-graded data	**95.50**	73.33	53.33	60.61
+2K PRM-graded data	94.60	**73.33**	**56.67**	**63.13**

Table 5: Performance (%) of DeepSeek-Distill-32B trained on varying sizes of PRM-graded math data.

(3) Influence of Training Data Size To investigate the influence of the size of the training data on reasoning performance, we trained the DeepSeek-Distill-32B model using 1K and 2K PRM-graded math examples. As shown in Table 5, increasing the training data size from 1K to 2K led to consistent performance improvements across all four math benchmarks.

(4) Impact of Training Data Distribution Standard We compare two strategies for defining training data distributions: **Option 1** leverages the base LLM to infer the true difficulty distribution from the evaluation dataset, while **Option 2** relies on human-defined prior distributions. As shown in Tab. 6, using the model-inferred evaluation distribution (Option 1) achieves stronger results on most benchmarks, indicating that aligning training distribution with evaluation difficulty leads to better generalization.

Sample Distribution	MATH			
	MATH 500	*AIME 24*	*AIME 25*	*GPQA*
Baseline-7B	89.40	56.67	33.30	49.49
+Option 1 (LLM)	**93.20**	60.00	**43.33**	**49.49**
+Option 2 (Human)	90.80	**63.33**	33.33	48.99

Table 6: Performance (%) of DS-distill-7B trained on 2K PRM-graded data using different training distribution strategies. Option 1 uses LLM-inferred evaluation-set distribution; Option 2 adopts human-defined priors.

(5) General Capability Evaluation on Core Benchmarks To assess the general reasoning and problem-solving capabilities of models trained with our adaptive difficulty synthesis pipeline, we conduct evaluations on a set of representative core benchmarks beyond just math and code. These include standard language understanding (MMLU, CMMLU), commonsense reasoning (BBH), and structured evaluation datasets (CEVAL, GSM8K, MBPP, HumanEval, MATH500). Specifically, we compare ZMath-32B and ZCode-32B against several competitive 32B open-source models, including Distill-32B, QwQ-32B, Qwen2.5-32B-Instruct.

As shown in Table 7, ZMath-32B achieves the highest overall average score (**89.98%**), setting new state-of-the-art results on MATH, HumanEval, and MBPP. Notably, ZMath-32B surpasses even models tailored for code generation in MBPP and HumanEval, demonstrating its strong generalization capability from math to code. While ZCode-32B also shows strong performance, especially on HumanEval and GSM8K, it is slightly behind ZMath-32B in average core capability. These results validate that our training strategy not only enhances performance in specialized domains but also strengthens general language model competencies across a wide range of tasks.

Visualization As shown in Figure 3, we compare the output differences between the DeepSeek-Distill-32B model and the ZMath-32B model based on samples from the AIME25 (of America, 2024) test set. Our model gives the correct answer, while the former produces an incorrect response. For ease of presentation, we only visualize the inference results of the two models without displaying the entire reasoning process.

5 Conclusion

In this paper, we propose a general and efficient method for constructing high-quality Chain-of-Thought (CoT) datasets. Firstly, we build a question database more aligned with the Base LLM itself by leveraging a method that adaptively grade question difficulty. This database possess a potential source of high-quality questions. Next, we use the difficulty distribution from either LLM performance on evaluation datasets or curriculum-learning-inspired difficulty levels, to sample crucial questions for improving the reasoning capability. Finally, these selected questions are employed to generate Chain-of-Thought data through

Model	*AVG_CORE*	*MATH*	*GSM8K*	*HUMANEVAL*	*MBPP*	*CEVAL*	*MMLU*	*CMMLU*	*BBH*
DeepSeek-Distill-32B	87.48	88.28	92.57	96	89.60	87.98	87.79	84.88	72.74
QwQ-32B	86.66	90.90	**96.21**	94	71.20	**89.25**	**89.34**	**87.12**	**75.27**
Qwen2.5-32B-instruct	80.61	77.74	94.54	84	76.60	83.53	79.27	81.66	67.57
ZCode-32B	87.33	89.68	95.98	96	84.60	87.54	87.84	85.62	71.41
ZMath-32B	**89.98**	**93.20**	96.06	**99**	**92.40**	88.94	88.82	86.78	74.66

Table 7: Evaluation results (%) of 32B-scale models on a diverse suite of core reasoning benchmarks. ZMath-32B and ZCode-32B show strong performance gains across both math and general tasks, outperforming prior state-of-the-art open-source models.

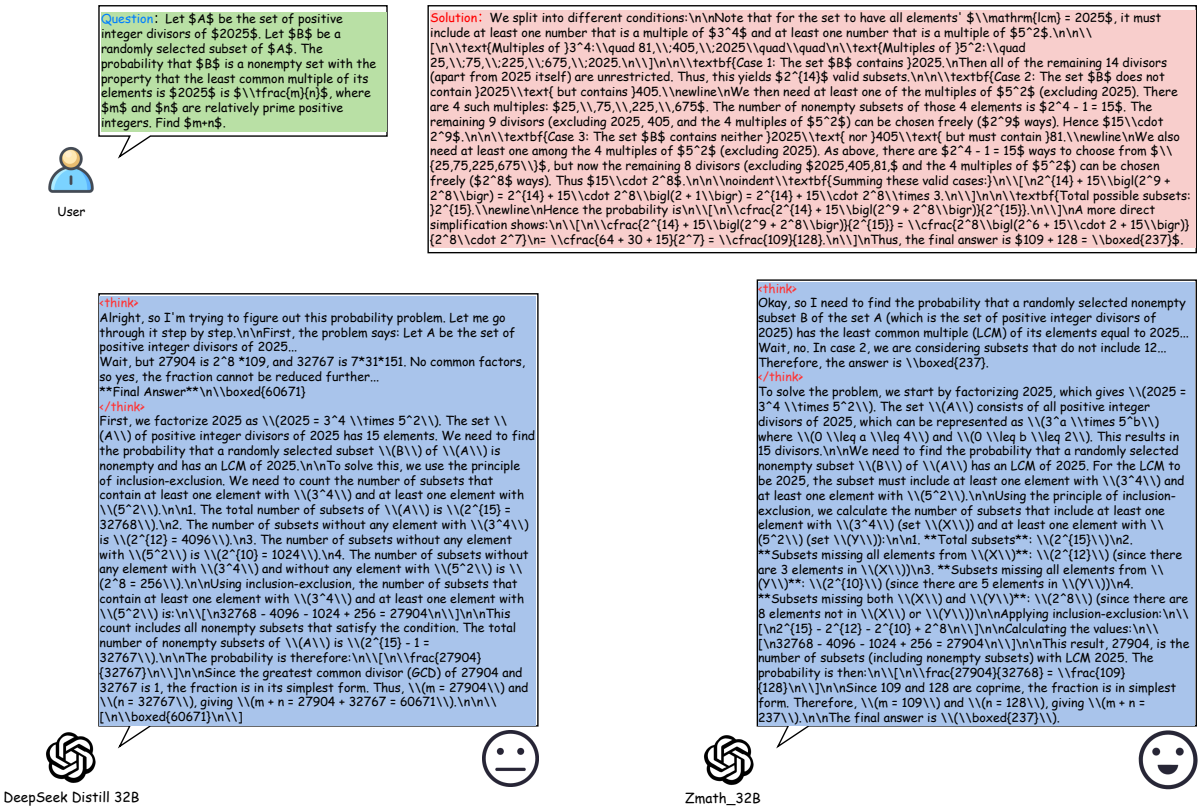

Figure 3: Comparison of DeepSeek-Distill-32B and ZMath-32B Model Outputs on AIME25 Test Samples.The output of our ZMath-32B is shown to be correct, while the DeepSeek-Distill-32B model gives an incorrect answer. For clarity, only the final inference results from each model are visualized, without the full reasoning process.

the teacher LLM (DeepSeek-R1), forming a COT dataset that is adaptively graded according to question difficulty aligned with the Base LLM. Benefiting from our constructed COT data, we effectively refine LLMs through supervised fine-tuning (SFT), achieving improved reasoning abilities across LLMs of different parameter scales. In the future, we plan to integrate our approach for constructing high-quality COT data with reinforcement learning or reject sampling, further enhancing the reasoning abilities of the models. We will also explore high-quality generation methods for tool-integrated reasoning (TIR) data, with the aim of further enhancing the reasoning abilities of our model.

6 Limitations

Our work has certain limitations. First, we have not conducted experiments on tasks beyond mathematics and code generation, so the generalizability of our model to other domains remains unverified. Second, we have not explored reinforcement learning-based approaches to further improve the reasoning capabilities of our model, primarily due to constraints in computing resources. These limitations suggest promising directions for future research.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael

Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. 2024. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, and 1 others. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *Advances in Neural Information Processing Systems*, 37:19209–19253.

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

Bespoke Labs. 2025. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation. https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation. Accessed: 2025-01-22.

Jung X Lee and Yeong-Tae Song. 2024. College exam grader using llm ai models. In *2024 IEEE/ACIS 27th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 282–289. IEEE.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. 2023. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*.

Lightman, Hunter, 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 preprint arXiv:2305.20050*.

Beiming Liu, Zhizhuo Cui, Siteng Hu, Xiaohua Li, Haifeng Lin, and Zhengxin Zhang. 2025. Llm evaluation based on aerospace manufacturing expertise: Automated generation and multi-model question answering. *arXiv preprint arXiv:2501.17183*.

Anthony Mahene, Daniel Pereira, Vincent Kowalski, Elizabeth Novak, Catherine Moretti, and Josephine Laurent. 2024. Automated dynamic data generation for safety alignment in large language models. *Authorea Preprints*.

Aniruddha Maiti, Samuel Adewumi, Temesgen Alemayehu Tikure, Zichun Wang, Niladri Sengupta, Anastasiia Sukhanova, and Ananya Jana. 2025. Comparative analysis of openai gpt-4o and deepseek r1 for scientific text categorization using prompt engineering. *arXiv preprint arXiv:2503.02032*.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, and 1 others. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Mathematical Association of America. 2024. Aime.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. 2025. Codeforces. https://huggingface.co/datasets/open-r1/codeforces.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

NovaSky Team. 2025a. Sky-t1: Train your own o1 preview model within 450. https://novasky-ai.github.io/posts/sky-t1. Accessed: 2025-01-09.

NovaSky Team. 2025b. Think less, achieve more: Cut reasoning costs by 50 https://novasky-ai.github.io/posts/reduce-overthinking. Accessed: 2025-01-23.

OpenThoughts Team. 2025c. Open Thoughts. https://open-thoughts.ai.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025a. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025b. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.

Claire Whitney, Edward Jansen, Victor Laskowski, and Charles Barbieri. 2024. Adaptive prompt regeneration and dynamic response structuring in large language models using the dynamic query-response calibration protocol. *Authorea Preprints*.

Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, and 1 others. 2024. A comparative study on reasoning patterns of openai's o1 model. *arXiv preprint arXiv:2410.13639*.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. 2024. Grade like a human: Rethinking automated assessment with large language models. *arXiv preprint arXiv:2405.19694*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. 2025. Agent-r: Training language model agents to reflect via iterative self-training. *arXiv preprint arXiv:2501.11425*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.