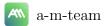# Leveraging Reasoning Model Answers to Enhance Non-Reasoning Model Capability

Haotian Wang, Han Zhao, Shuaiting Chen, Xiaoyu Tian, Sitong Zhao, Yunjie Ji, Yiping Peng, and Xiangang Li

a-m-team

Abstract

Recent advancements in large language models (LLMs), such as DeepSeek-R1 and OpenAI-o1, have demonstrated the significant effectiveness of test-time scaling, achieving substantial performance gains across various benchmarks. These advanced models utilize deliberate "thinking" steps to systematically enhance answer quality. In this paper, we propose leveraging these high-quality outputs generated by reasoning-intensive models to improve less computationally demanding, non-reasoning models. We explore and compare methodologies for utilizing the answers produced by reasoning models to train and improve non-reasoning models. Through straightforward Supervised Fine-Tuning (SFT) experiments on established benchmarks, we demonstrate consistent improvements across various benchmarks, underscoring the potential of this approach for advancing the ability of models to answer questions directly.

1 Introduction

The field of Natural Language Processing has experienced remarkable advancements with the development of large language models (LLMs) (Min et al., 2021; Kaplan et al., 2020). A significant trend in enhancing the capabilities of these models is test-time scaling (Yang et al., 2025; Wu et al., 2025), where increasing computational resources allocated during inference leads to notable performance improvements. Models such as OpenAI's o1 series (OpenAI, 2024) and DeepSeek-R1 (DeepSeek-AI, 2025) have demonstrated the effectiveness of this approach across various tasks and benchmarks (Lightman et al., 2023; Huang et al., 2024). The capability of these models to achieve superior results by allocating additional computational resources during inference indicates an important shift in optimizing performance for LLMs. Specifically, dedicating more computation to the answer-generation process, rather than solely relying on scaling training data and model parameters, can lead to significant improvements, particularly in tasks that require complex reasoning (Snell et al., 2024). The success of test-time scaling thus emphasizes the crucial role of computation during the answer-generation phase.

A key factor in the success of advanced models lies in their inherent ability to perform explicit reasoning before arriving at a final answer (Wei et al., 2023; Snell et al., 2024; Wu et al., 2025). This deliberate "think" step enables models to evaluate multiple potential solutions, resulting in more accurate and nuanced answers. Techniques like test-time scaling have facilitated the generation of these intermediate reasoning steps, significantly boosting performance on challenging tasks (Huang and Chang, 2022). Given the enhanced quality of answers produced by such reasoning models, a relevant question emerges: Can these high-quality answers be effectively utilized to enhance the performance of less computationally intensive, non-reasoning models?

This paper investigates this question by exploring strategies for leveraging the outputs of reasoning models to enhance the capabilities of non-reasoning models. Our central hypothesis is that training non-reasoning models using improved answers derived from reasoning models can lead to superior performance. To validate
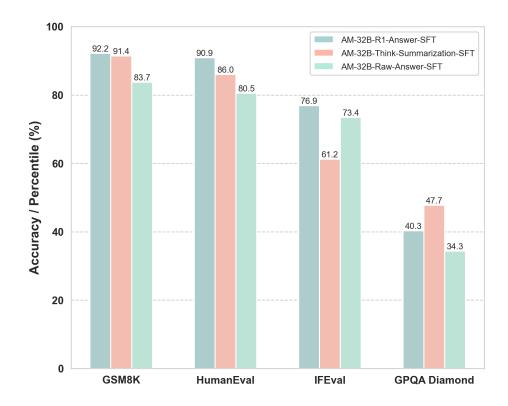
1

Figure 1: Benchmark performance of different answer utillize method

this hypothesis, we conduct a series of experiments on several benchmark datasets, comparing different approaches to incorporating reasoning-model outputs into the training of non-reasoning models. Specifically, the contributions of this paper are: (1) we provide empirical evidence demonstrating that utilizing high-quality answers from reasoning models for training significantly enhances the performance of non-reasoning models. (2) we systematically investigate various methods for integrating outputs generated by reasoning models to improve the capabilities of non-reasoning models.

2 Approach

This section details our proposed approach, beginning with the generation of a well-distributed and diverse Supervised Fine-Tuning (SFT) dataset. Following this, we utilize DeepSeek-R1(DeepSeek-AI, 2025), yielding high-quality think and answer. We also utilize DeepSeek-v3-0324(DeepSeek-AI, 2024) to generate non-resoning response for comparison. Crucially we introduce and rigorously compare distinct methods for exploiting these distilled answers. The core of our technical contribution lies in proposing these varied utilization strategies and evaluating their effectiveness for improving non-reasoning models.

2.1 Data Description

Our dataset was constructed in two main stages: first, the collection of input prompts and second, the sampling of the corresponding responses.

- **Prompt Collection:** To ensure broad coverage and data diversity, we curated our dataset from open-source communities. It incorporates data derived from several established collections, including Infinity Instruct, OpenCoder(Huang et al., 2024),PRIME(Cui et al., 2025),NuminaMath(LI et al.,

2024),CodeContests(Li et al., 2022), FLAN(Wei et al.), Orca(Mukherjee et al., 2023), AM-DeepSeek-R1-Distilled-1.4M(Zhao et al., 2025),tuluv3(Lambert et al., 2024) and more. This aggregation spans multiple critical domains such as mathematics, code, science, general question answering, instruction following, tooluse and more.In total, our data collection efforts resulted in approximately 1.3 million instances.

- **Response Collection:** Following the curation of our diverse query set, we utilized DeepSeek-R1 to generate reasoning process and answers.In addition to generating data via direct distillation from the DeepSeek-R1 model, we supplemented our dataset by directly selecting a subset of instances from the AM-DeepSeek-R1-Distilled-1.4M(Zhao et al., 2025), which represents one of the current state-of-the-art resources in this domain. To elicit high-quality and domain-appropriate outputs, we implemented a tailored prompting strategy. Specifically, distinct user prompts were designed and applied for queries falling within the mathematics and code domains, acknowledging their unique structural and logical demands compared to general tasks. Furthermore, to capture the reasoning process, the generated outputs were structured to distinguish between intermediate thinking steps (enclosed in <think></think>tags) and the final conclusive answer (enclosed in <answer></answer>tags). Detailed system prompts, domain-specific user prompts can be found in Appendix.
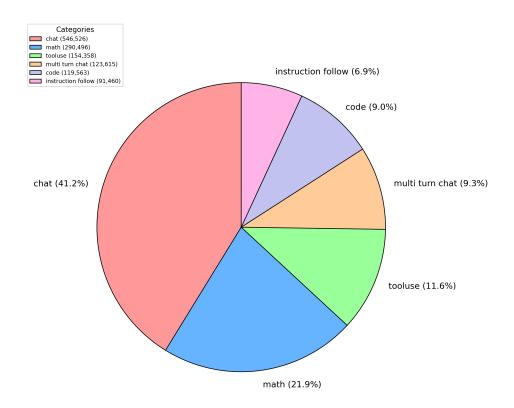

Figure 2: Category Distribution of dataset

2.2 Methods for Utilizing Reasoning Response

To evaluate different strategies for leveraging reasoning models to generate informative responses, we explored three distinct methods. Let Q represent the input query. We define the following components:

- R_{orig}: The original response for query Q obtained from the baseline open-source community dataset.

- M_{reason}: The reasoning large language model.

- $(T_{reason}, A_{reason})=M_{reason}(Q)$: The output of the reasoning model M_{reason} for query Q, consisting of an intermediate thinking component T_{reason} and a final answer component A_{reason}.

- M_{sum}: A separate summarization model (specifically, Qwen2.5-7B-Instruct) used in Method 4.

- $\oplus$: An operator representing the concatenation of two text strings.

The three methods generate final responses, denoted R_1,R_2,R_3 respectively, as follows:

1. **Original Response:** This method utilizes the raw response directly from the community dataset, serving as our baseline for comparison. The final response R_1 is simply:

$$R_1 = R_{orig} \tag{1}$$

   This represents the unprocessed data from the source dataset.

2. **Direct Reasoning Model Output (Answer Component):** This approach uses only the answer component A_{reason} generated directly by reasoning model M_{reason}.The final response R_2 is:

$$R_2 = A_{reason} \tag{2}$$

   This method isolates the direct answer produced by the reasoning model, potentially lacking the intermediate steps outlined in T_{reason}.

3. **Think Summarization:** To create a response that includes the reasoning process without excessive length, this method first summarizes the thinking component T_{reason} using the summarization model M_{sum}. Let the generated summary be S_{think}, where:

$$S_{think} = M_{sum}(T_{reason}) \tag{3}$$

   This summary S_{think} , capturing the essential problem-solving steps, is then prepended to the reasoning model's original answer component A_{reason}. The final response R_3 is constructed as :

$$R_3 = S_{think} \oplus A_{reason} \tag{4}$$

   The goal is to integrate the core reasoning path with the final answer, yielding a more comprehensive and explanatory output.

In addition to the datasets generated by the three approaches outlined above, we utilized the DeepSeek-v3-0324(DeepSeek-AI, 2024) model to produce a fourth set of responses. This dataset serves as a control group for our analysis, bringing the total number of datasets evaluated in this paper to four.

3   Experiments

3.1   Evaluation

3.1.1   Benchmark

We evaluated non-reasoning model's ability using GPQA-Diamond (Rein et al., 2023), GSM8K(Cobbe et al., 2021),MMLU(Hendrycks et al., 2021),HumanEval(Chen et al., 2021),IFEval(Zhou et al., 2023),Align-Bench(Liu et al., 2023), MTBench(Zheng et al., 2023). These benchmarks span multiple fields and difficulty levels, enabling a thorough assessment of the model's performance across diverse scenarios.

3.1.2 Evaluation Methodology

To ensure consistent and comparable results across all evaluated models, we standardized the generation parameters. The maximum generation length was uniformly set to 16,384 tokens for all tasks.We employed two distinct decoding strategies based on the specific requirements of the evaluation benchmarks:

1. **Stochastic Sampling:** For benchmarks requiring probabilistic generation, GPQA-Diamond (Rein et al., 2023), we utilized a temperature of 0.6 and a top-p value of 0.95. To estimate the pass@1 metric for these benchmarks, we generated 8 candidate samples per input instance.

2. **Greedy Decoding:** For all other benchmarks, we employed greedy decoding by setting the temperature to 0.0 and generating a single sample per input instance. This deterministic approach was applied to GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), IFEval (Zhou et al., 2023), MMLU (Hendrycks et al., 2021), AlignBench (Liu et al., 2023), and MTBench (Zheng et al., 2023).

Within the greedy decoding setting, specific configurations were adopted for certain benchmarks:

- IFEval(Zhou et al., 2023), we reported the prompt-strict score.

- For MMLU(Hendrycks et al., 2021), evaluations were conducted using a 5-shot prompting methodology.

- For AlignBench(Liu et al., 2023) and MTBench(Zheng et al., 2023), the generated responses were subsequently evaluated using the OpenAI GPT-4 model (OpenAI, 2024) as the judge.

3.2 Experiment Setup

We conducted SFT on Qwen2.5-32B(Qwen, 2024) using datasets mentioned in Section 2.2. We employed a cosine Learning Rate Scheduler, set the Learning Rate to 8e-6, the Warm Up Ratio to 0.05, the Batch Size to 64, and the Max Token Length to 16,384. The training process consisted of three epochs.

3.3 Results and Analysis

Table 1: SFT-Model Performance

Model	GPQA-Diamond pass@1	GSM8K	MMLU	HumanEval	IFEval (prompt strict)	AlignBench	MTBench
OLMo-2-32B-0325-SFT	-	78.4	76.1	-	72.4	-	-
AM-32B-DeepSeek-V3-Answer-SFT	**47.9**	90.2	70.1	90.9	76.6	**7.6**	**8.2**
AM-32B-R1-Answer-SFT	40.3	**92.2**	**82.5**	**90.9**	76.9	6.9	7.6
AM-32B-Think-Summarization-SFT	47.7	91.4	81.0	86.0	61.2	7.2	7.9
AM-32B-Raw-Answer-SFT(Baseline)	34.3	83.7	82.5	80.5	73.4	6.2	7.7

Our experimental results demonstrate that directly using the answer portion from response of reasoning model for model training via Supervised Fine-Tuning (SFT) leads to significant improvements on several key benchmarks, particularly HumanEval(Chen et al., 2021), GSM8K(Cobbe et al., 2021), and GPQA(Rein et al., 2023). However, we observed a slight decrease in performance on chat-oriented metrics such as AlignBench(Liu et al., 2023) and MT Bench(Zheng et al., 2023). Our analysis focuses on elucidating the impact of different data generation methods derived from a reasoning model's output on the fine-tuned non-reasoning model's capabilities across diverse benchmarks. **Baseline Performance:** The AM-32B-Raw-Answer-SFT model, fine-tuned directly on the original community dataset responses (R_{orig}), serves as our primary baseline. It establishes a reference point for evaluating the efficacy of incorporating reasoning model outputs.

Impact of Direct Reasoning Answers: Utilizing only the direct answer component (A_{reason}) from the reasoning model for SFT (AM-32B-R1-Answer-SFT) yielded substantial performance improvements on

several key reasoning and coding benchmarks. As indicated in Table 1, this model achieved the highest scores among our SFT variants on GSM8K(Cobbe et al., 2021) (92.2) and HumanEval(Chen et al., 2021) (90.9), and demonstrated significant gains on GPQA-Diamond (Rein et al., 2023) (40.3) compared to the baseline (34.3). However, this approach resulted in marginally lower performance on chat-oriented benchmarks, namely AlignBench(Liu et al., 2023) and MTBench(Zheng et al., 2023), compared to the baseline. We attribute this phenomenon to the structure of the reasoning model's output; the detailed procedural explanations often reside within the reasoning trace (T_{reason}), leaving the answer component (A_{reason}) overly concise and potentially lacking the conversational context preferred in chat interactions.

Benefits and Trade-offs of Think Summarization: The AM-32B-Think-Summary-SFT model, trained using the summarized thinking process concatenated with the answer ($S_{think} \oplus A_{reason}$), was designed to mitigate the conversational shortcomings observed with the Think Summarization approach. This method achieved the highest GPQA(Rein et al., 2023) score (47.7) among our models, approaching the performance of the distilled external model, and improved performance on MTBench(Zheng et al., 2023) (7.9). These results suggest that explicitly incorporating a summary of the reasoning process enhances understanding and potentially improves alignment in conversational contexts. However, this structural modification introduced a notable trade-off, evidenced by a significant decrease in performance on IFEval(Zhou et al., 2023) (61.2). We find that the substantial alteration of the original answer format may interfere with the model's ability to adhere strictly to instructions as defined by the IFEval(Zhou et al., 2023) benchmark. Despite this specific deficit, the Think Summarization method generally provided a more balanced performance profile across reasoning and chat-related tasks compared to using the direct answer alone.

Implications for Utilize Reasoning Model's Power: Comparing our results, particularly the performance of AM-32B-R1-Answer-SFT, with the externally sourced AM-32B-DeepSeek-V3-Answer-SFT model highlights a critical insight. Simply fine-tuning on the final answers (A_{reason}) extracted from a capable reasoning model, while beneficial for certain tasks, does not automatically transfer the full spectrum of the source model's capabilities, especially concerning conversational abilities or potentially nuanced reasoning reflected in the thought process. This underscores the necessity of developing more sophisticated methods that effectively structure and integrate both the reasoning process and the final answer when the objective is knowledge distillation or holistic capability transfer via SFT. The effectiveness of the Think Summarization approach, despite its IFEval limitation, points towards the importance of response content organization in this process.

4   Discussion and Conclusion

The results presented in this paper affirm that supervised fine-tuning (SFT) using response data derived from reasoning models can significantly enhance the performance of target language models. Our investigation systematically evaluated three distinct methodologies for utilizing these reasoning-derived outputs, revealing that the effectiveness of knowledge transfer is critically dependent on the specific strategy employed for structuring the SFT data.

Crucially, our findings demonstrate that simply leveraging the final answer component (A_{reason}) from a reasoning process, while boosting performance on certain reasoning and coding benchmarks, may not yield holistic improvements and can even slightly degrade performance in conversational alignment metrics. This highlights the importance of the information's structure; methods incorporating summarized reasoning steps (S_{think}) offered alternative performance profiles, often achieving better balance across diverse tasks or excelling in specific areas like instruction following, albeit sometimes involving trade-offs (e.g., the observed IFEval performance reduction with the Think Summarization method).

These results underscore the potential of leveraging reasoning outputs as a potent form of data augmentation for SFT, offering a viable pathway towards enhancing the capabilities of large language models. The variations in performance across methods emphasize that the manner in which reasoning-derived knowledge is structured and presented during fine-tuning is a key determinant of the resulting model's strengths and weaknesses. This work contributes practical strategies for such capability transfer, demonstrating tangible improvements across multiple standard benchmarks.

Building on these findings, future research should explore more sophisticated techniques for extracting, representing, and integrating the knowledge embedded within the reasoning process (T_{reason}). Investigating alternative summarization strategies, methods for dynamically combining reasoning steps with final answers, or techniques for explicitly modeling the reasoning structure could potentially unlock further performance gains and lead to the development of more robust and versatile models through optimized knowledge distillation.

5   Limitation

A notable limitation of the present study pertains to the Think Summarization methodology employed for generating training data. While this approach successfully integrates aspects of the reasoning process by summarizing the thinking trace (T_{reason}), the resulting summarized steps (S_{think}) may not fully capture the original fidelity or granularity inherent in the reasoning model's detailed thought process. Consequently, the SFT data generated via this method might represent a less precise approximation of the underlying reasoning compared to the original trace itself.

An alternative strategy, warranting further investigation, involves leveraging prompt engineering. Specifically, one could potentially refine the prompts used to elicit responses from the source reasoning model, instructing it to directly incorporate essential, concise reasoning steps within the structure of its final answer. This approach could obviate the need for post-hoc summarization and potentially yield training data that more faithfully represents integrated reasoning and conclusions.

However, the systematic exploration and optimization of such prompt engineering techniques fell outside the defined scope of the current project due to practical constraints on time and resources. Nevertheless, developing methods to elicit more integrated reasoning directly via prompting remains a promising avenue for future research, potentially offering a more direct and higher-fidelity approach to creating SFT data that effectively transfers reasoning capabilities.

References

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.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Siming Huang, Tianhao Cheng, Jason Klein Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J. Yang, J. H. Liu, Chenchen Zhang, Linzheng Chai, Ruifeng Yuan, Zhaoxiang Zhang, Jie Fu, Qian Liu, Ge Zhang, Zili Wang, Yuan Qi, Yinghui Xu, and Wei Chu. Opencoder: The open cookbook for top-tier code large language models. 2024. URL https://arxiv.org/pdf/2411.04905.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. 2024.

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. Numinamath. [https://hf-mirror.com/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL https://www.science.org/doi/abs/10.1126/science.abq1158.

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.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. Alignbench: Benchmarking chinese alignment of large language models, 2023.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey, 2021. URL https://arxiv.org/abs/2111.01243.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Qwen. Team qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2025. URL https://arxiv.org/abs/2408.00724.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for llm reasoning, 2025. URL https://arxiv.org/abs/2502.18080.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model traning, 2025. URL https://github.com/a-m-team/a-m-models/blob/main/docs/AM-DeepSeek-R1-Distilled-Dataset.pdf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A   Prompt

System Prompt

You are a helpful assistant. To answer the user's question, you first think about the reasoning process and then provide the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.

User prompt (CODE)

Let's think step by step and output the final answer within
```python
your code
```

User prompt (MATH)

Let's think step by step and output the final answer within \boxed{}.

Figure 3: prompt used in inference