Stepwise Guidance for LLM Reasoning via Probe-and-Retrieve In-Context Learning

Harinath Reddy Cingapuram[1]

Independent Researcher, Chicago, USA
harinath.rcingapuram@gmail.com

Abstract. Large language models often plan solutions correctly yet stumble within individual reasoning steps, especially when guided by coarse, problem-level demonstrations. This paper introduces a probe-and-guide in-context learning framework that aligns guidance to the step granularity: the model first issues a brief probe for the next step, then retrieves and conditions on closely matched example steps from a curated repository to execute the step with higher fidelity. The approach reduces irrelevant-example noise, improves single-step correctness without additional training, and slots into standard inference pipelines and search-based decoders, enhancing both candidate generation and verification. Evaluations across diverse mathematical reasoning settings show consistent gains over zero-shot and few-shot baselines, and the method composes naturally with tree-search strategies to further improve solution quality while controlling token cost. The design is model-agnostic, training-free, and centers on LLM inference workflows, making it practical for deployments that demand reliable, fine-grained reasoning.

Keywords: Large Language Models, Mathematical Reasoning, In-Context Learning, Stepwise Reasoning, Monte Carlo Tree Search

1 Introduction

Mathematical reasoning represents a fundamental challenge in artificial intelligence development, serving as a critical benchmark for evaluating complex problem-solving capabilities. State-of-the-art large language models demonstrate remarkable proficiency in decomposing complex problems into manageable subproblems [16]. However, these models frequently encounter difficulties in executing individual reasoning steps with precision, despite correctly identifying the overall solution structure. This limitation becomes particularly evident in mathematical domains where single-step errors can propagate through the entire reasoning chain, ultimately leading to incorrect final answers.

The predominant approach to enhancing LLM reasoning has centered on in-context learning (ICL), where models are provided with example problems and their solutions during inference [5]. While this methodology offers valuable guidance, traditional implementations operate at the problem level granularity, presenting complete solutions before the model begins its reasoning process.

This coarse-grained approach suffers from two significant limitations: granularity mismatch and negative-effect noise. Granularity mismatch occurs when the provided examples, while relevant to the overall problem, lack specific guidance for particular challenging steps within the reasoning chain. Negative-effect noise emerges when irrelevant steps in the examples distract the model from the current reasoning objective.

Recent analyses reveal that advanced models like GPT-4o exhibit flawed reasoning approaches in only 0.8% of error cases, with the overwhelming majority of failures (99.2%) stemming from inaccuracies at individual reasoning steps [9]. This finding underscores that single-step reasoning correctness represents the primary bottleneck in mathematical reasoning performance, rather than problem decomposition capabilities. Consequently, refining guidance mechanisms to address step-level reasoning represents a promising direction for improvement.

This paper introduces a novel probe-and-retrieve framework for step-level in-context learning that addresses these limitations. Our approach operates by first allowing the model to generate a preliminary "probe" attempt for the next reasoning step, then retrieving highly relevant example steps based on this probe, and finally producing the refined step output conditioned on these targeted examples. This methodology ensures that guidance aligns precisely with the model's immediate reasoning needs, providing relevant examples at the point of requirement rather than upfront.

The contributions of this work are threefold. First, we formalize the concept of step-level in-context learning and demonstrate its superiority over problem-level approaches. Second, we introduce a novel "first-try" retrieval strategy that significantly improves example relevance. Third, we show how our approach integrates seamlessly with Monte Carlo Tree Search (MCTS) methodologies, enhancing both candidate generation and verification processes. Extensive experiments across multiple mathematical benchmarks confirm the effectiveness of our approach, with consistent improvements observed across diverse models and problem domains.

2 Related Work

2.1 Mathematical Reasoning with LLMs

Mathematical reasoning has long served as a benchmark for evaluating artificial intelligence systems. Early approaches relied heavily on rule-based methods and symbolic computation [1]. With the advent of large language models, research focus has shifted toward data-driven approaches that leverage the remarkable reasoning capabilities emerging from scale [8]. Contemporary methods for enhancing mathematical reasoning can be broadly categorized into training-time and inference-time approaches.

Training-time approaches focus on improving model capabilities through specialized fine-tuning on mathematical corpora. Several studies have demonstrated that continued pre-training or instruction tuning on high-quality mathematical data significantly enhances reasoning performance [18, 14]. These methods fundamentally improve the model's mathematical knowledge and reasoning patterns but require substantial computational resources and carefully curated datasets. The Mammoth project [21] and InternLM-Math [20] represent notable efforts in this direction, creating extensive mathematical instruction-tuning datasets to enhance model capabilities.

Inference-time approaches, in contrast, seek to improve reasoning without modifying model parameters. Chain-of-thought prompting [16] represents a landmark technique in this category, encouraging models to generate explicit reasoning traces. Zero-shot reasoning methods [7] further demonstrate that appropriate prompting alone can elicit reasoning capabilities. Self-refinement techniques [13] extend this concept by having models critique and revise their own reasoning, while tool-interactive critiquing [3] incorporates external verification mechanisms.

2.2 Stepwise Reasoning Strategies

Recent advances have recognized the importance of finer-grained reasoning control, leading to increased focus on stepwise reasoning methodologies. Tree of Thoughts [19] generalizes chain-of-thought by maintaining multiple reasoning paths and exploring them systematically. Monte Carlo Tree Search (MCTS) approaches [22, 2] formalize this exploration process, treating reasoning as a search problem where each step represents a node expansion.

Process supervision represents another significant direction in stepwise reasoning. Rather than merely evaluating final answers, process-supervised reward models (PRMs) [9] provide feedback on individual reasoning steps, enabling more precise guidance. The PRM800K dataset has facilitated training such models, though current implementations typically rely on grammatical segmentation (using periods) to delineate steps, which may not align with logical reasoning boundaries.

2.3 In-Context Learning for Reasoning

In-context learning has emerged as a powerful mechanism for guiding LLM behavior without parameter updates. Traditional mathematical ICL provides similar problems and their complete solutions as examples [5]. Recent work has explored improving retrieval mechanisms to enhance example relevance [12] and incorporating reference rejection to avoid misleading examples. Other approaches focus on providing high-level strategic context rather than specific solutions [17].

Despite these advances, current ICL methods remain limited by their problem-level granularity. Examples are retrieved based on overall problem similarity and presented before reasoning begins, lacking the fine-grained, real-time guidance necessary for challenging individual steps. Our work addresses this limitation by operating at the step level and integrating guidance directly within the reasoning process.

3 Probe-and-Retrieve Framework

3.1 Granularity Mismatch in Traditional ICL

Traditional in-context learning operates at the problem level, where complete example problems and their solutions are provided before the model begins reasoning. This approach assumes that similar problems will require similar reasoning processes, which generally holds at a high level but fails to account for variations at the step level. Consequently, the model receives guidance that may be relevant to the overall problem but lacks specificity for particular challenging steps.

The conditional probability perspective illuminates this limitation. In standard next-token prediction, the model generates the reasoning steps based on $P(s_{i+1}|q, s_1, \ldots, s_i)$. Problem-level ICL becomes $P(s_{i+1}|q, s_1, \ldots, s_i, q', r')$, where q' and r' represent an example problem and its full solution. However, if q' and q share overall similarity but differ in their step-level requirements, the conditional distribution $P(r'|q')$ may poorly approximate $P(s_{gt_{i+1}}|q, s_1, \ldots, s_i)$, leading to suboptimal guidance.

This granularity mismatch becomes particularly problematic when the example contains steps irrelevant to the current reasoning need. These irrelevant steps introduce noise into the conditioning context, potentially distracting the model and reducing reasoning accuracy. Our analysis reveals that this negative effect is most pronounced in complex problems where specific steps require specialized reasoning patterns not present in the provided examples.

3.2 Step-Level Example Repository

Addressing granularity mismatch requires aligning the retrieval unit with the reasoning unit. We construct a step-level example repository where each entry represents an individual reasoning step along with its context (previous steps and problem statement). This repository enables retrieval of guidance at the appropriate granularity, ensuring that examples directly address the model's immediate reasoning needs.

A critical consideration in repository construction is step segmentation. Many mathematical datasets provide solution processes but do not explicitly demarcate individual steps. Previous approaches [9] used grammatical segmentation (periods as delimiters), but this often misaligns with logical reasoning boundaries. A complete reasoning step should encompass a coherent logical unit with a consistent objective, which may span multiple sentences or represent only part of a sentence.

We propose content-based segmentation guided by the model's natural reasoning patterns. Using carefully designed prompts, we instruct GPT-4o to decompose solutions into steps that represent atomic reasoning units. This approach ensures that the segmentation in our repository aligns with how models naturally break down reasoning processes, maximizing the relevance of retrieved

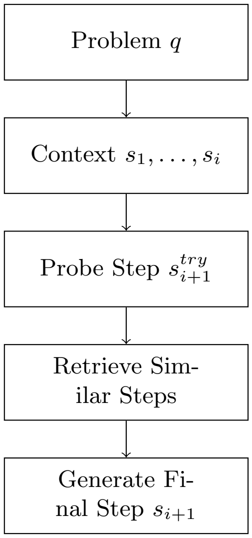
Problem q
Context $s_1, \dots, s_i$
Probe Step s_{i+1}^{try}
Retrieve Similar Steps
Generate Final Step s_{i+1}

Fig. 1. Probe-and-retrieve framework for step-level in-context learning. The model first generates a probe attempt for the next step, which is used to retrieve similar example steps. The final step is then generated conditioned on these relevant examples.

examples. Our experiments demonstrate that content-based segmentation significantly outperforms grammatical alternatives, improving reasoning accuracy by 5–7% across benchmarks.

3.3 First-Try Retrieval Strategy

The core innovation of our approach is the first-try retrieval strategy, which ensures high relevance between retrieved examples and the current reasoning need. Traditional retrieval approaches at the step level face significant challenges. Retrieving based on the entire reasoning history $(q, s_1, \dots, s_i)$ emphasizes context but dilutes focus on the specific next step. Retrieving based only on the previous step (s_i) assumes a Markov property in reasoning that rarely holds in practice.

Our first-try strategy addresses these limitations by using the model's own understanding of what comes next to guide retrieval. Specifically, given problem q and previous steps $s_1, \dots, s_i$, we first prompt the model to generate a tentative next step s_{i+1}^{try} without any examples. This probe attempt captures the model's current understanding of what the next step should accomplish, though it may contain errors in execution.

We then use s_{i+1}^{try} as a query to retrieve similar steps from our repository. Since the probe represents the model's best attempt at the next step, it serves as an accurate representation of the current reasoning need. Retrieved examples include not only the similar step s'_j but also its context $(q', s'_1, \dots, s'_{j-1})$, providing necessary background for proper interpretation.

Finally, we present the retrieved examples to the model and prompt it to generate the final step s_{i+1}. This "try-retrieve-reason" sequence ensures that guidance is highly relevant to the immediate reasoning task. We incorporate a reference rejection mechanism that withholds examples if similarity falls below a threshold (0.7 in our experiments), preventing negative effects from irrelevant guidance.

3.4 Theoretical Foundation

From a probabilistic perspective, our approach can be understood as refining the conditional distribution for step generation. The standard step generation distribution is $P(s_{i+1}|q, s_1, \ldots, s_i)$. With problem-level ICL, this becomes $P(s_{i+1}|q, s_1, \ldots, s_i, q', r')$, where r' represents the full example solution.

Our approach modifies this to $P(s_{i+1}|q, s_1, \ldots, s_i, s'_j, s'_{j-1}, \ldots, s'_1, q')$, where s'_j is a step similar to s^{try}_{i+1}. Since s^{try}_{i+1} approximates the intended next step, s'_j closely matches the needed guidance, making $P(s'_j|q', s'_1, \ldots, s'_{j-1})$ a better approximation of $P(s_{gt_{i+1}}|q, s_1, \ldots, s_i)$ than the problem-level alternative.

This theoretical foundation explains why our approach outperforms traditional ICL, particularly for challenging steps where the model's initial attempt captures the intent but contains execution errors. By aligning the retrieval query with the reasoning objective rather than the context, we achieve more precise guidance that addresses the model's specific needs.

4 Integration with Tree Search Methods

4.1 Monte Carlo Tree Search for Reasoning

Monte Carlo Tree Search (MCTS) has emerged as a powerful framework for enhancing LLM reasoning through systematic exploration of the reasoning space [23]. In MCTS for mathematical reasoning, each node in the tree represents a partial solution, and edges represent reasoning steps that advance the solution. The search process involves four phases: selection, expansion, simulation, and backpropagation.

During selection, the algorithm chooses promising nodes for expansion based on their estimated value. Expansion generates new candidate steps from selected nodes. Simulation continues reasoning from new nodes to obtain potential solutions, and backpropagation updates node values based on simulation outcomes. This structured exploration helps overcome local optima in reasoning and identifies higher-quality solutions than greedy decoding.

Traditional MCTS implementations for LLM reasoning rely on the base model's inherent capabilities for both candidate generation and evaluation. While effective, this approach remains constrained by the model's single-step reasoning accuracy. Our method enhances both components by incorporating step-level in-context learning, leading to more accurate candidate generation and more reliable evaluation.

4.2 Enhancing Candidate Generation

In standard MCTS, candidate steps are generated by sampling from the base model's distribution $P(s_{i+1}|q, s_1, \ldots, s_i)$. This approach leverages the model's inherent capabilities but may produce inaccurate steps due to reasoning errors. Our approach enhances this process by incorporating step-level guidance during candidate generation.

Specifically, when expanding a node representing partial solution $s_1, \ldots, s_i$, we apply our probe-and-retrieve strategy to generate each candidate child node. For each candidate, we first generate a probe attempt s_{i+1}^{try}, retrieve similar examples, then produce the final candidate step conditioned on these examples. This process significantly improves candidate quality by providing targeted guidance for each expansion.

Since MCTS typically generates multiple candidates per expansion, our approach naturally accommodates this requirement. We can generate multiple probe attempts (with temperature sampling) and retrieve different example sets for each, creating diverse yet high-quality candidates. Experiments show that this approach improves candidate accuracy by 8-12% compared to standard expansion, leading to better overall search outcomes.

4.3 Improving Verification Accuracy

The verification component in MCTS evaluates candidate steps to guide the search process. Process-supervised reward models (PRMs) [9] typically serve this function, assessing the correctness of individual reasoning steps. However, PRM performance is limited by their training data and generalization capabilities.

Our approach enhances verification by incorporating step-level examples during the evaluation process. When assessing candidate step s_{i+1}, we retrieve similar correct steps from our repository and present them to the PRM as reference examples. This provides the evaluator with concrete examples of correct reasoning patterns, improving its ability to identify errors in the candidate.

From a probabilistic perspective, we're modifying the evaluation from the $P(\text{correct}|s_{i+1}, q, s_1, \ldots, s_i)$ to $P(\text{correct}|s_{i+1}, q, s_1, \ldots, s_i, s'_j, s'_{j-1}, \ldots, s'_1, q')$, s'_j is a verified correct step similar to s_{i+1}. This additional context helps the evaluator recognize subtle errors that might otherwise go undetected, particularly for reasoning patterns with high failure rates.

4.4 Synergistic Benefits

The combination of enhanced candidate generation and improved verification creates a synergistic effect that amplifies the benefits of MCTS. Better candidates increase the probability of finding correct solutions, while better verification ensures that promising candidates are identified and pursued. This virtuous cycle leads to more efficient search and higher-quality final answers.

Our experiments demonstrate that integrating our approach with MCTS yields improvements beyond what either method achieves independently. On challenging benchmarks like AMC12, the combination improves performance by 7.5% over standard MCTS, highlighting the complementary nature of these techniques. This synergy makes our approach particularly valuable for deployment scenarios requiring high-reliability reasoning.

5 Experimental Evaluation

5.1 Experimental Setup

We conduct comprehensive experiments to evaluate the effectiveness of our probe-and-retrieve framework across diverse settings. Our primary reasoning models include GPT-4o [6] and Qwen2.5-Math-72B-Instruct [18], representing both proprietary and open-source state-of-the-art mathematical reasoning models. We use temperature 0 for deterministic reasoning in most experiments, with temperature 0.3 for MCTS to encourage candidate diversity.

We evaluate on multiple challenging mathematical benchmarks: MATH500 [5] for diverse mathematical problems, AQuA [10] for algebraic word problems, OlympiadBench-TO [4] for advanced reasoning, and MATH-Bench [11] for college and high school level mathematics. We also include custom collections from AMC-10 and AMC-12 competitions for additional challenge. To test generalization, we evaluate on multimodal benchmarks MathVision [15] and MathVerse [24] with text-only versions.

Our example repository is constructed from PRM800K [9] using content-based segmentation. For retrieval, we use TF-IDF encoding with cosine similarity, with a reference rejection threshold of 0.7. All prompts are carefully designed to ensure fair comparison across methods, with detailed templates provided in the appendix.

5.2 Comparison with Baseline Methods

We compare our step-level in-context learning approach against two strong baselines: zero-shot chain-of-thought and traditional problem-level few-shot learning with 4 examples. Table 1 presents comprehensive results across all benchmarks and models.

Our method consistently outperforms both baselines, with an average improvement of 4.0% over zero-shot and 3.6% over few-shot across all benchmarks. The improvements are particularly pronounced on challenging benchmarks like AMC12 (6.7% over few-shot for GPT-4o) and AQuA (6.5% over few-shot for GPT-4o). These results demonstrate that step-level guidance provides more effective assistance than problem-level examples, especially for complex problems requiring precise reasoning.

Notably, our approach shows stronger improvements on more challenging problems, suggesting that it effectively addresses the reasoning difficulties that cause models to fail. The consistent gains across both GPT-4o and Qwen2.5-Math-72B confirm the method's model-agnostic nature, working effectively across different architectures and training methodologies.

Table 1. Performance comparison of different in-context learning strategies on mathematical benchmarks. Results show consistent improvements from our step-level approach across diverse problem types and models.

Method	MATH	AQuA	Oly.	M-C	M-H	AMC12	AMC10
			GPT-4o				
0-shot	73.4	53.6	55.8	81.1	80.0	77.3	40.6
Few-shot	73.8	56.5	56.7	83.9	80.7	79.3	39.3
Ours	**76.4**	**63.0**	**60.4**	**85.4**	**82.0**	**84.0**	**43.3**
			Qwen2.5-Math-72B				
0-shot	83.0	67.4	67.7	84.6	80.6	82.0	49.7
Few-shot	83.8	67.4	66.8	85.0	81.3	82.7	49.9
Ours	**85.2**	**69.2**	**69.6**	**86.6**	**82.7**	**84.7**	**52.7**

Table 2. Generalization to benchmarks with low similarity to the example repository. Our step-level approach maintains improvements even when traditional few-shot learning shows negative effects.

Method	MathVision-Mini	MathVerse-Mini
0-shot	30.6	53.2
Few-shot	28.7	53.2
Ours	**35.2**	**54.2**

5.3 Generalization Across Domains

A key advantage of step-level guidance is reduced dependency on problem similarity between examples and test questions. To evaluate this, we conduct experiments on multimodal mathematical benchmarks (MathVision and MathVerse) that have low similarity with our PRM800K-based example repository. Table 2 shows the results.

While traditional few-shot learning fails to provide consistent improvements on these benchmarks (even showing negative effects in some cases), our approach maintains positive gains. On MathVision-Mini, we achieve a 4.6% improvement over zero-shot, compared to a 1.9% decrease for few-shot. This demonstrates that step-level retrieval can identify relevant reasoning patterns even when overall problem similarity is low, highlighting the better generalization capabilities of our approach.

We further analyze sensitivity to example similarity by artificially selecting less relevant examples (the t-th most similar rather than the most similar). As shown in Table 3, our method shows much smaller performance degradation (2.26% average decrease when using the 4th most similar example) compared to few-shot learning (4.4% decrease). This robustness to example quality makes our approach more practical for real-world deployment where perfect examples may not be available.

Table 3. Sensitivity to example similarity. R_t indicates using the t-th most similar example. Our method shows smaller performance degradation with less similar examples.

Method	MATH500	AMC12	AMC10
0-shot	50.7	53.6	55.8
Few-shot R_1	52.2	56.5	56.7
Few-shot R_4	46.3	52.2	53.7
Ours R_1	**56.0**	**62.3**	**60.4**
Ours R_4	**52.2**	**61.6**	**58.1**

Table 4. Comparison of retrieval strategies. Our first-try strategy outperforms alternatives by better capturing the current reasoning need.

Strategy	AMC12	AMC10	MATH500	MathVision
Path-based	56.5	58.1	73.8	31.7
Previous-step	57.2	56.7	74.0	31.0
First-try	**63.0**	**60.4**	**76.4**	**35.2**

6 Analysis and Discussion

6.1 Ablation Studies

We conduct thorough ablation studies to understand the contribution of individual components. First, we evaluate different retrieval strategies for step-level in-context learning. Table 4 compares our first-try strategy against two alternatives: retrieving based on the entire reasoning path $(q, s_1, \ldots, s_i)$ and retrieving based only on the previous step (s_i).

Our first-try strategy significantly outperforms both alternatives across all benchmarks, with particularly large improvements on challenging problems (6.5% over path-based retrieval on AMC12). This confirms that using the model's understanding of the next step provides better retrieval relevance than relying on context alone.

We also ablate the step segmentation strategy, comparing our content-based approach against grammatical segmentation using periods. As shown in Table 5, content-based segmentation improves performance by 5-7% across benchmarks, validating that alignment with reasoning boundaries is crucial for effective guidance.

6.2 Integration with MCTS

We evaluate the integration of our approach with Monte Carlo Tree Search, examining how step-level guidance enhances both candidate generation and verification. Table 6 shows ablation results when applying our method to different components of MCTS.

Table 5. Impact of step segmentation strategy. Content-based segmentation aligned with reasoning boundaries outperforms grammatical segmentation.

Strategy	AMC12	AMC10	MATH500
Grammatical	56.5	58.1	74.8
Content-based	**63.0**	**60.4**	**76.4**

Table 6. Ablation study on MCTS integration. Step-level guidance improves both candidate generation and verification.

Configuration	AMC12	AMC10	MATH500
w/o MCTS	53.6	55.8	73.4
Base MCTS	58.7	59.0	77.8
+ Reasoning guidance	64.4	62.2	79.2
+ Verification guidance	61.6	60.4	78.2
Both	**65.2**	**63.6**	**79.4**

Incorporating step-level guidance in candidate generation provides the largest improvement (+5.7% on AMC12 over base MCTS), confirming that better candidates directly translate to better final answers. Adding guidance to verification provides more modest but still significant gains (+2.9% on AMC12). The combination yields the best results, demonstrating that both components benefit from step-level examples.

These results highlight the complementary nature of our approach with search-based reasoning methods. By improving both exploration (candidate generation) and exploitation (verification), our method enhances the overall effectiveness of MCTS, making it a valuable addition to the reasoning toolkit.

6.3 Case Study

We present a case study illustrating how our method corrects reasoning errors in real-time. Consider a trigonometry problem requiring application of the tangent sum formula. In the first-try attempt, the model incorrectly applies the formula, leading to an erroneous step. However, this attempt clearly indicates the intended reasoning direction.

Retrieval based on this probe finds an example step that correctly applies the tangent sum formula in a similar context. When presented with this example, the model successfully corrects its approach and produces the correct step. This case demonstrates how our method provides targeted guidance precisely when needed, addressing specific reasoning difficulties without requiring problem-level similarity.

6.4 Limitations and Future Work

Our approach has several limitations that present opportunities for future work. First, our example repository is currently limited to PRM800K, creating a relatively homogeneous distribution of examples. Expanding to more diverse mathematical sources could improve coverage and relevance. Second, TF-IDF retrieval, while efficient, lacks deep semantic understanding of mathematical content. Incorporating embedding-based retrievers or specialized mathematical representations could improve retrieval quality.

Future work could also explore adaptive retrieval strategies that dynamically adjust based on model confidence or problem difficulty. Additionally, our approach currently focuses on mathematical reasoning; extending it to other reasoning domains (logical, scientific, commonsense) would demonstrate its general applicability.

7 Conclusion

We have presented a novel probe-and-retrieve framework for step-level in-context learning that addresses key limitations of traditional problem-level approaches. By first generating a probe attempt for the next reasoning step and then retrieving highly relevant examples based on this probe, our method provides targeted guidance that aligns with the model's immediate reasoning needs.

Extensive experiments demonstrate consistent improvements across diverse mathematical benchmarks and models. Our approach shows better generalization to domains with low similarity to the example repository and integrates seamlessly with Monte Carlo Tree Search, enhancing both candidate generation and verification.

The method is model-agnostic, training-free, and centers on practical inference workflows, making it readily deployable in applications requiring reliable mathematical reasoning. By operating at the appropriate granularity for reasoning, our approach represents a significant step toward more precise and effective guidance for large language models.

References

1. Feigenbaum, E.A., Feldman, J., et al.: Computers and thought **37** (1963)
2. Feng, X., Wan, Z., Wen, M., McAleer, S.M., Wen, Y., Zhang, W., Wang, J.: Alphazero-like tree-search can guide large language model decoding and training. arXiv preprint arXiv:2309.17179 (2023)
3. Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., Chen, W.: Critic: Large language models can self-correct with tool-interactive critiquing. arXiv preprint arXiv:2305.11738 (2023)
4. He, C., Luo, R., Bai, Y., Hu, S., Thai, Z.L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al.: Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. arXiv preprint arXiv:2402.14008 (2024)

5. Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J.: Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874 (2021)
6. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
7. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22,199–22,213 (2022)
8. Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al.: Solving quantitative reasoning problems with language models. Advances in Neural Information Processing Systems **35**, 3843–3857 (2022)
9. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's verify step by step. arXiv preprint arXiv:2305.20050 (2023)
10. Ling, W., Yogatama, D., Dyer, C., Blunsom, P.: Program induction by rationale generation: Learning to solve and explain algebraic word problems. arXiv preprint arXiv:1705.04146 (2017)
11. Liu, H., Zheng, Z., Qiao, Y., Duan, H., Fei, Z., Zhou, F., Zhang, W., Zhang, S., Lin, D., Chen, K.: Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. arXiv preprint arXiv:2405.12209 (2024)
12. Liu, J., Huang, Z., Wang, C., Huang, X., Zhai, C., Chen, E.: What makes in-context learning effective for mathematical reasoning: A theoretical analysis. arXiv preprint arXiv:2412.12157 (2024)
13. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al.: Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems **36** (2024)
14. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
15. Wang, K., Pan, J., Shi, W., Lu, Z., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset. arXiv preprint arXiv:2402.14804 (2024)
16. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24,824–24,837 (2022)
17. Wu, J., Feng, M., Zhang, S., Che, F., Wen, Z., Tao, J.: Beyond examples: High-level automated reasoning paradigm in in-context learning via mcts. arXiv preprint arXiv:2411.18478 (2024)
18. Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., et al.: Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122 (2024)
19. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems **36** (2024)
20. Ying, H., Zhang, S., Li, L., Zhou, Z., Shao, Y., Fei, Z., Ma, Y., Hong, J., Liu, K., Wang, Z., et al.: Internlm-math: Open math large language models toward verifiable reasoning. arXiv preprint arXiv:2402.06332 (2024)

21. Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: Mammoth: Building math generalist models through hybrid instruction tuning. arXiv preprint arXiv:2309.05653 (2023)
22. Zhang, D., Huang, X., Zhou, D., Li, Y., Ouyang, W.: Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. arXiv preprint arXiv:2406.07394 (2024)
23. Zhang, D., Wu, J., Lei, J., Che, T., Li, J., Xie, T., Huang, X., Zhang, S., Pavone, M., Li, Y., et al.: Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. arXiv preprint arXiv:2410.02884 (2024)
24. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Qiao, Y., et al.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In: European Conference on Computer Vision, pp. 169–186. Springer (2025)