Proceedings of Machine Learning Research vol 278:1–11, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Enhancing Robustness in Multi-Step Reasoning: A Synergistic Approach Combining Planning with Reflective Self-Correction

Xinyuan Wang WANGXINYUAN2024@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences, Beijing, China
the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Danli Wang* DANLI.WANG@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences, Beijing, China
the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Hehao Zhang ZHANGHEHAO2023@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences, Beijing, China
the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

Bo You YOUBO2019@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Xueen Li XUEE.LI@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Yu Gu* GUYUPIG@163.COM
Inner Mongolia Digital Information Co., LTD, Hohhot, China
**Corresponding author*

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

Large Reasoning Models (LRMs) have significantly advanced complex problem-solving capabilities by incorporating extended chain-of-thought (CoT) reasoning. However, managing error propagation over long inference chains remains a critical challenge. In this work, we propose a novel self-supervised framework named **P**lanning and **Re**flective **S**elf-**C**orrection that integrates two complementary mechanisms: planning phase and reflection phase. The planning phase decomposes complex queries into streaming sub-problems and generates detailed reasoning trajectories, while the reflection phase leverages corrective feedback from erroneous outputs to refine these trajectories. The datasets sampled through these two mechanisms are used for self-supervised training, further reinforcing the LLM's reasoning capabilities. Experiments conducted on the multi-hop Question Answering dataset demonstrate that our approach enhances the model's ability to generate coherent and accurate reasoning paths. Ablation studies further reveal the distinct contributions of planning and reflection to the overall performance. Our results suggest that integrating anticipatory planning with reflective self-correction provides a promising avenue for robust long-range inference in LRMs.

Keywords: Large Reasoning Models, Chain-of-Thought Reasoning, Self-Supervised Learning, Reflective Feedback, Planning

1. Introduction

Large Reasoning Models (LRMs) represent a transformative advancement in artificial intelligence, rivaling the impact of ChatGPT's inception (Besta et al., 2025). By leveraging extended chain-of-thought (CoT) reasoning, LRMs have redefined complex problem-solving paradigms (Wei et al., 2022). Pioneering systems, such as OpenAI's o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), exhibit exceptional multi-step reasoning capabilities, achieving unprecedented outcomes in mathematical proofs, algorithmic design, and scientific inquiry (Yu et al., 2023; Zhong et al., 2024). These sophisticated models build upon traditional large language models (LLMs) by integrating enhanced reasoning mechanisms that support nuanced inference, improved contextual understanding, and robust decision-making. The LRMS decomposition strategy iteratively constructs comprehensive reasoning trajectories, typically involving many steps, which has promoted scientific discovery and has become the cornerstone of AGI-oriented research (El Naqa et al., 2020; Kislev, 2022) .

Central to the effectiveness of LRMs is their ability to generate and refine intermediate CoT reasoning steps. Extending the intermediate CoT reasoning mechanism requires not only incorporating additional intermediate steps and enhancing their quality but also integrating robust planning techniques. For example, hierarchical planning enables the model to decompose complex tasks into a series of logically ordered subproblems, with each subtask's solution informing subsequent steps (Wang et al., 2023a). Similarly, Monte Carlo Tree Search techniques allow the model to simulate multiple action sequences before committing to a final decision, while iterative backtracking supports the re-evaluation and adjustment of subgoals when a chosen path is suboptimal (Zhou et al., 2023). Moreover, Tree-of-Thought prompting, in which multiple reasoning paths are generated concurrently and the most promising branch is selected through a voting or filtering mechanism, facilitates dynamic evaluation and continuous refinement of plans (Yao et al., 2023a). Collectively, these planning strategies result in a more structured, adaptive, and reliable CoT, ultimately enhancing the model's performance on complex, multi-step reasoning tasks.

Despite significant advances in CoT reasoning for LRMs, a key challenge remains in managing the propagation of errors over long reasoning horizons. Even minor inaccuracies in early reasoning steps can amplify as the process unfolds, ultimately degrading overall performance (Wang et al., 2025). Various strategies including iterative self-reflection, dynamic re-planning, and reinforcement learning based validators have been explored to address this issue. However, a robust solution that effectively curbs error accumulation is still lacking. To address this challenge, we have enhanced the planning capabilities of LRMs to generate more coherent and consistent inference paths. We have also introduced a reflective mechanism that learns from errors in the reasoning process, thereby producing more robust and efficient reasoning trajectories. We denote our proposed framework as **PReSC** (Planning and Reflective Self-Correction).

In our research, we extend this CoT paradigm by integrating a systematic planning phase prior to the reasoning process. Specifically, our methodology first leverages an LRM to generate a detailed plan for each input question, effectively decomposing complex queries into streaming sub-problems. This planning not only clarifies the overall problem structure but also identifies critical subgoals and outlines potential solution pathways, thereby guiding the subsequent multi-step reasoning process. Once the plan is established, the LRM proceeds with an in-depth reasoning phase to produce extensive chains-of-thought that detail intermediate inferences. To ensure reliability, we filter and collect only those reasoning trajectories that yield correct solutions. Recognizing that errors in reasoning can offer valuable insights, our framework incorporates a reflective mechanism that re-plans and re-reasons

on the erroneous paths. This self-reflection enables the model to diagnose its mistakes and generate corrected reasoning sequences. Finally, the model is fine-tuned using both the verified correct reasoning paths and the refined outputs from the reflection mechanism. This comprehensive training approach reinforces the LRM's ability to generate coherent, accurate, and adaptive reasoning chains in future tasks.

Our main contributions are summarized as follows:

- We introduce **PReSC**, a framework reinforces the reasoning capabilities of the model. By leveraging a detailed, stream-generated plan as a starting point, it ensures clear and coherent guidance throughout the reasoning process.
- The **PReSC** introduces a reflection mechanism that enables the model to analyze and correct errors during reasoning, thereby enhancing its capacity to learn from mistakes and self-correct.
- We demonstrate that targeted fine-tuning on the **PReSC** framework significantly improves the model's reasoning performance by reinforcing correct inference trajectories and incorporating reflective error-correction to optimize decision-making on complex problems.

2. Related Work

2.1. Large Reasoning Models

Unlike traditional LLMs that scale by increasing model parameters or enlarging training corpora, LRMs boost problem-solving abilities by executing extended reasoning steps during inference. This development has led to two distinct yet complementary research streams: implicit and explicit reasoning mechanisms.

Implicit reasoning models, exemplified by QwQ (Qwen Team, 2024), fully internalize reasoning structures within neural weights through black-box computation. While effective for pattern-driven domains like machine translation (Wang et al., 2024), their opacity limits error detection and external knowledge integration. Explicit reasoning architectures decouple the reasoning process from the model's intrinsic parameters by incorporating standalone modules dedicated exclusively to inference. For example, systems such as LLaMA-Berry (Zhang et al., 2024) and marco-01 (Zhao et al., 2024) integrate algorithmic components, including Monte Carlo Tree Search planners and validators that use reinforcement learning, into modular inference pipelines. These frameworks support iterative hypothesis generation and facilitate solution refinement through explicit state tracking. Consequently, explicit model introduces additional computational overhead, resulting in more latency, compared to the implicit model.

2.2. Reasoning Schemes of LRMs

In recent work, reasoning in language models has been formalized as the decomposition of complex inputs into a series of intermediate steps that cumulatively lead to the final answer. CoT reasoning decomposes intricate problems into sequential, verifiable intermediate steps, thereby enhancing both computational accuracy and interpretability, which is critical for effective multi-step inference (Wei et al., 2022). Building on this idea, several extensions have emerged. For instance, Tree-of-Thought prompting enables the model to explore multiple reasoning trajectories by evaluating diverse paths and performing self-assessment to select the most promising option (Yao et al., 2023a). Self-Consistency extends the reasoning process by generating multiple reasoning chains and subsequently employing a consensus mechanism (e.g., majority voting) to select the most reliable solution, thereby

mitigating the compounding of errors that may occur in a single chain (Wang et al., 2023b). Graph-of-Thought employs an explicit graph-based representation to capture non-linear and interdependent relationships among reasoning steps (Besta et al., 2024). The above works expands the reasoning process of LRMs by evolving from simple sequential decompositions to more structured.

3. Method

As shown in Fig.1, Our methodology unfolds in two main stages: (1) data samplings and (2) self-supervised training. In the following, we will provide detailed descriptions of each stage.

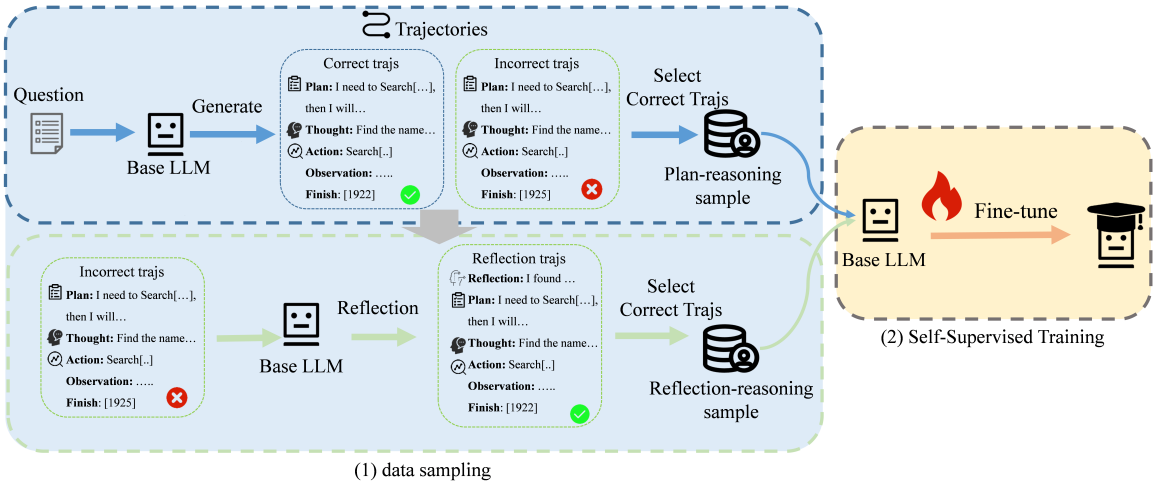

Figure 1: The overview of **PReSC**. The framework is composed of two main stages. In the data sampling stage, the base model generates a detailed plan and subsequently produces reasoning trajectories. Erroneous trajectories are identified and corrected via a reflection process, yielding refined reasoning outputs. In the self-supervised training stage, these outputs are used to fine-tune the model, enhancing its overall reasoning performance.

3.1. Data Samplings

During the data sampling stage, the base LLM generates a detailed reasoning plan for each query. Based on this plan, the model then carries out a series of reasoning processes and filtering out correct execution paths. For incorrect paths, the model triggers a reflection mechanism to re-infer reasoning and ultimately integrates all corrected trajectories as the output basis.

In the planning phase, for a given query q, the base LLM constructs a detailed plan. This plan delineates which [entities] information should be retrieved and specifies the procedures for processing it. For instance, for the query *Who is older, Annie Morton or Terry Richardson?*, the plan is formulated as follows: *I need to Search[Annie Morton] and Search[Terry Richardson] to obtain their birth dates, and then I will compare the dates to Finish[older one].* Mathematically, the plan Mathematically, the plan p can be represented as:

$$p \sim \pi_\theta \left(\cdot \mid \rho_{\text{Plan}}, q \right), \tag{1}$$

where π_θ represents the base model with parameters θ, and ρ_{Plan} denotes the few-shot prompt used for planning.

In the reasoning phase, the base LLM employs the constructed plan p to generate reasoning trajectories following the ReAct (Reason+Act) framework (Yao et al., 2023b). In this approach, the model alternates between generating internal thoughts and executing corresponding actions. Each trajectory comprises a sequence of thoughts, which capture the internal reasoning and deliberation, and actions, which represent concrete steps such as retrieving information, searching for keywords, and answering queries. Each thought guides the subsequent action, and each action produces an observation that offers feedback on information retrieval and keyword searches. Mathematically, the reasoning trajectory R is generated as:

$$R(t_0, a_0, o_0, ..., t_n, a_n) \sim \pi_\theta \left(\cdot \mid \rho_{\text{ReAct}}, p \right), \tag{2}$$

where π_θ denotes the few-shot prompt used for the ReAct reasoning process. t_i denotes the internal thought generated at step i. a_i denotes the corresponding action executed at step i. o_i denotes the observation obtained as a result of executing a_i.

In the reflection phase, the model identifies the erroneous trajectories based on the ground-truth feedback and initiates a reflection process. During this process, the base model generates a reflection $C_{reflcetion}$ that examines the errors in previous trajectories and formulates new guidance for subsequent reasoning. This reflection is then used to refine the reasoning trajectory R'. Mathematically, the reflection process can be expressed as:

$$C_{reflcetion} \sim \pi_\theta \left(\cdot \mid \rho_{\text{reflection}}, p_{incorrect}, R_{incorrect}, q \right), \tag{3}$$

$$R' \sim \pi_\theta \left(\cdot \mid \rho_{\text{reflection}}, p_{incorrect}, R_{incorrect}, q, C_{reflection} \right), \tag{4}$$

where π_θ denotes the few-shot prompt used for reflection, $p_{incorrect}$ represents the plan associated with the incorrect reasoning, $R_{incorrect}$ denotes the erroneous trajectories.

Data sampling stage procedures yield two datasets: D_{plan} and $D_{reflection}$. The $D_{plan} = \{q_i,\, p_i,\, R_i\}_{i=1}^{N}$ dataset encompasses the planning data and reasoning data, while the $D_{reflection} = \{q_i,\, C_i,\, R_i'\}_{i=1}^{N}$ dataset comprises the refined reasoning trajectories produced via the reflection process.

3.2. Self-Supervised Training

In the self-supervised training stage, the model is further fine-tuned using the datasets generated during the data sampling phase. Specifically, the planning dataset D_{plan} and the reflection dataset $D_{reflection}$ are leveraged to optimize the model's reasoning capabilities by self-supervised training.

For each query $q_i \in D_{plan}$ in two dataset, the model is trained to generate the corresponding plan p_i and reasoning trajectory R_i from D_{plan}, thereby learning to map input queries to coherent reasoning chains. Formally, the training objective for the planning stage can be expressed as:

$$\mathcal{L}_{plan} = -\sum_{i=1}^{N} \log P(p_i, R_i \mid q_i; \theta), \tag{5}$$

where θ denotes the model parameters.

Similarly, the reflection dataset $D_{reflection}$ is utilized to refine the reasoning process by incorporating corrective feedback. Here, the model is trained to generate both the reflection C_i and the

refined reasoning trajectory R_i' based on the corrective signals. The corresponding training objective is defined as:

$$\mathcal{L}_{reflection} = -\sum_{i=1}^{N} \log P(C_i, R_i' \mid q_i; \theta). \tag{6}$$

Through iterative fine-tuning on these self-supervised objectives, the model progressively improves its ability to generate accurate and coherent reasoning trajectories.

4. Experiments

4.1. Datasets and Setup

To evaluate our model's reasoning capabilities, we employ the HotpotQA dataset (Yang et al., 2018), a benchmark for multi-hop question answering. This dataset offers a platform for evaluating the model's abilities in multi-step planning, comprehension, and information retrieval, testing model's capacity to effectively extract accurate information from a wide range of online resources. Within the HotpotQA environment, our model utilizes the Wikipedia API to obtain relevant information for multi-hop question answering task inference. In this context, the model interacts through three primary actions: Search[entity] to initiate searches, Lookup[string] to identify specific details, and Finish[answer] to deliver the final response. Wikipedia passages are retrieved using a public API.

4.2. Implementation Details

Our base model using the LLAMA-3-8B-Instruct model (Dubey et al., 2024). The model is trained for 3 epochs with a learning rate of 3×10^{-5}. For efficient fine-tuning, we employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) with hyperparameters set to $lora\_r = 32$, $lora\_alpha = 32$, and a dropout rate of 0.05. The LoRA adaptation targets the following modules: `"q_proj"`, `"v_proj"`, `"k_proj"`, and `"o_proj"`. In addition, we use a cosine learning rate scheduler with a warmup ratio of 0.1, a weight decay of 0.01, and the AdamW optimizer. From the training set, we sampled 1,500 data instances.

For validation, we sample 500 instances from the development set for evaluation. We employ exact match (EM) and F1 metrics. The EM metric is binary, taking a value of 1 if the model-generated answer exactly matches the reference answer in content, order, punctuation, and spelling, and 0 otherwise. In contrast, the F1 score, calculated as the harmonic mean of precision and recall, provides a more nuanced assessment by capturing partial correctness, ranging continuously from 0 to 1.

4.3. Baselines

In our study, the following baseline methods were used for comparison: FireAct is a method for fine-tuning a language model through agent tracks generated from multiple tasks and prompt methods (Chen et al., 2023). CoT is a prompt-based method designed to enhance a model's performance in complex reasoning tasks by guiding it through a step-by-step reasoning process (Wei et al., 2022). ReAct is a framework for collaborative reasoning and action in large language models. This approach improves the model's performance and interpretability in complex tasks by guiding the model to generate alternating "think," "act," and "observe" steps (Yao et al., 2023b).

5. Results

5.1. Main Results

Table 1 presents a performance comparison of various reasoning methods on the HotpotQA dataset. Notably, when comparing our method to fine-tuning methods that utilize the same model size and training data, such as FireAct, our method demonstrates effectiveness. Specifically, FireAct achieves an EM score of 0.262, while our method achieves a score of 0.304, marking an improvement of 16%. In addition, the CoT approach obtains an EM score of 0.294. In contrast, our method, using a 7B model, delivers an EM score of 0.304, outperforming the CoT approach and even ReAct (Palm-540B), which scores 0.274. This demonstrates that our approach offers a competitive performance in multi-step reasoning tasks.

Notably, our result slightly underperforms ReAct (GPT-3) in EM scores (0.304 vs. 0.308) primarily due to the substantial parametric knowledge gap caused by GPT-3's significantly larger model size (175B parameters) and training data scale. This data advantage proves critical in knowledge-intensive QA tasks. While our model (Llama3-8B) exhibits marginally lower performance than ReAct (GPT-3), it demonstrates unique advantages in computational efficiency and practical deployability. These efficiency gains position our approach as a highly deployable lightweight solution for resource-constrained environments requiring adaptive multi-step reasoning.

Table 1: Performance comparison on the HotpotQA dataset.

Method	**EM**
FireAct (Chen et al., 2023)	0.262
ReAct (Palm-540B) (Yao et al., 2023b)	0.274
CoT (Wei et al., 2022)	0.294
ReAct (GPT-3) (Yao et al., 2023b)	0.308
Ours	0.304

5.2. Ablation Study Results

Our ablation experiments reveal the distinct contributions of the planning and reflection trajectories to the overall performance of the model, as shown in Table 2. Without feedback, the few-shot approach achieves an F1 score of 0.307 and an EM of 0.234. Fine-tuning using only the planning dataset (D_{plan}) improves these scores to 0.324 (F1) and 0.242 (EM), while further incorporating the reflection dataset ($D_{reflection}$) yields scores of 0.345 (F1) and 0.254 (EM). When the feedback mechanism is enabled, combining both D_{plan} and $D_{reflection}$ in the fine-tuning process under feedback conditions results in the highest performance, with an F1 of 0.387 and an EM of 0.304. These results underscore the complementary roles of planning and reflection trajectories and demonstrate that our feedback training mechanism significantly enhances the model's reasoning capabilities.

Notably, after incorporating external feedback, the model trained on D_{plan} did not better than the performance of the few-shot approach. A possible explanation is that the fine-tuning process may lead to some loss of generalization ability. Nonetheless, its performance still exceeds that of the best model without feedback.

Table 2: Ablation study results on the HotpotQA dataset.

Feedback	Method	Training Dataset		Metrics	
		D_{plan}	$D_{reflection}$	F1	EM
w/o	Few-shot			0.307	0.234
	Fine-tune	✓		0.324	0.242
	Fine-tune	✓	✓	0.345	0.254
w/	Few-shot			0.362	0.294
	Fine-tune	✓		0.297	0.264
	Fine-tune	✓	✓	0.387	0.304

5.3. Impact of PPL-Based Reflection Triggering

To improve the reliability of LRMs, we introduce a reflection mechanism that is conditionally activated based on the model's output perplexity (PPL). Specifically, when the PPL of a generated answer exceeds a predefined threshold, the system considers the answer uncertain and triggers a reflection process in which the model re-generates the output with revised reasoning. Otherwise, the original answer is preserved. We selected the model fine-tuned on the D_{plan} and $D_{reflection}$ dataset.

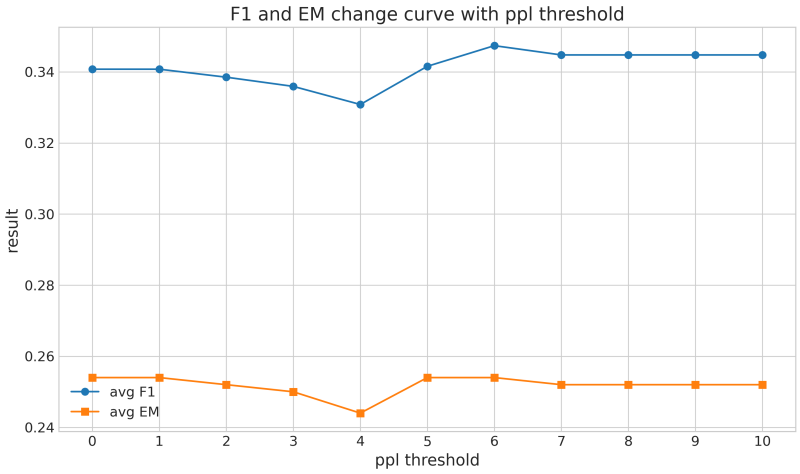

Figure 2: Average F1 and EM scores under different perplexity (PPL) thresholds.

We evaluate this mechanism by varying the PPL threshold from 0 to 10 and measuring its effect on QA accuracy, as captured by average F1 and Exact Match (EM) scores. As shown in the Fig.2, both metrics show relatively minor fluctuations across the range. At a threshold of 0 (i.e., always using the original response), the average F1 is 0.3407 and EM is 0.2540. The best F1 score (0.3474) is achieved when the threshold is set to 6, suggesting that selectively triggering the reflection mechanism at moderate PPL levels can improve answer quality. However, as the threshold increases beyond this point, both F1 and EM exhibit slight declines, stabilizing at 0.3448 and 0.2520 respectively at the threshold of 10. These findings indicate that PPL can serve as a lightweight yet effective indicator for initiating self-reflection in LLMs. When appropriately calibrated, this approach enhances answer precision while avoiding unnecessary re-computation on confident predictions.

6. Conclusions and Feature Work

In this paper, we introduced **P**lanning and **Re**flective **S**elf-**C**orrectiona, a framework that enhances the reasoning capabilities of large language models by integrating streaming planning and reflective feedback. Our approach integrates two complementary mechanisms: the planning phase, which decomposes complex queries into manageable subproblems, and the reflection phase, which refines reasoning trajectories based on corrective feedback. The datasets sampled through these mechanisms are further used for self-supervised training, reinforcing the model's reasoning capabilities. Experimental results on the HotpotQA dataset demonstrate that our method improved performance as measured by F1 and EM metrics. Ablation studies confirm the critical roles of both the planning and reflection components, underscoring their complementary contributions. Overall, our work provides a robust strategy for enhancing chain-of-thought reasoning in LRMs and opens up new directions for future research in error management and dynamic self-correction within complex inference tasks.

Despite the results, our approach still faces challenges in the trigger of the reflection mechanism. In future work, we plan to refine the reflection trigger process by implementing a more granular reflection mechanism. Additionally, we aim to extend our evaluation to a wider range of datasets, further improving the generalizability and robustness of the framework.

Statement on AI-Assisted Writing

This paper incorporates text generated with the assistance of OpenAI's GPT-4 and ChatGPT. The AI system was used to refine language clarity and improve readability in the introduction and discussion sections. All content has been reviewed and validated by the authors to ensure accuracy and alignment with the research objectives.

Acknowledgments

This research is supported by the Beijing Natural Science Foundation under Grant No. L232098, the National Natural Science Foundation of China under Grant No. 61872363.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690, Mar 2024. doi: 10.1609/aaai.v38i16.29720. URL https://ojs.aaai.org/index.php/AAAI/article/view/29720.

Maciej Besta, Julia Barth, Eric Schreiber, Ales Kubicek, Afonso Catarino, Robert Gerstenberger, Piotr Nyczyk, Patrick Iff, Yueling Li, Sam Houliston, Tomasz Sternal, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Łukasz Flis, Hannes Eberhard, Hubert Niewiadomski, and Torsten Hoefler. Reasoning language models: A blueprint, 2025.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024.

Issam El Naqa, Masoom A Haider, Maryellen L Giger, and Randall K Ten Haken. Artificial intelligence: reshaping the practice of radiological sciences in the 21st century. *The British journal of radiology*, 93(1106):20190855, 2020.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card, 2024.

Elyakim Kislev. *Relationships 5.0: How AI, VR, and robots will reshape our emotional lives*. Oxford University Press, 2022.

Qwen Team. Qwq-32b-preview. https://qwenlm.github.io/zh/blog/qwq-32b-preview/, 2024.

Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. Steca: Step-level trajectory calibration for llm agent learning, 2025.

Jiaan Wang, Fandong Meng, Yunlong Liang, and Jie Zhou. Drt: Deep reasoning translation via long chain-of-thought, 2024.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistic. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2023.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning, 2024.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1: Towards open reasoning models for open-ended solutions, 2024.

Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2023.