Fallback Prompting Guides Large Language Models for Accurate Responses in Complex Reasoning

Jianing Sun[1], Zhichao Zhang[2], Xiaopu Wang[1], Xinyuan Ji[1], Yizhi Zhang[1]
[1]School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, 710062, China
[2]School of Information Science and Technology, Hainan Normal University, Haikou, Hainan, 571158, China

Since the introduction of Chain-of-Thought (CoT), leveraging Large Language Models (LLMs) to solve complex reasoning problems has become possible. While an increasing number of studies focus on improving the accuracy of answers, there still lack of efficient mechanism for errors evaluation and rectification during the reasoning process. To tackle this challenge, we propose a new strategy, fallback prompting, to enable self-refinement of LLMs based on a feedback-driven method. Our main goal is to precisely locate and revise errors through a backward evaluation process. We conducted experiments on seven datasets across three reasoning tasks: arithmetic reasoning, symbolic reasoning, and knowledgeable reasoning. The results demonstrate that fallback prompting achieves state-of-the-art performance across all datasets and models. Notably, it achieves near-perfect accuracy of 99.3% on Chinese-school-Math with Qwen2.5 and delivers outstanding results on symbolic and knowledgeable reasoning tasks, including 91.7% accuracy on HIST and 97.3% on CSQA with GLM4. These findings highlight the effectiveness and robustness of fallback prompting in enhancing LLMs' reasoning capabilities, offering a promising direction for improving reasoning accuracy through self-refinement.

Index Terms—**Chain-of-Thought, Large Language Models, complex reasoning, prompt tuning, error propagation.**

I. Introduction

The emergence and rapid advancement of transformer-based large language models (LLMs) have catalyzed a groundbreaking transformation in the field of natural language processing (NLP) [1]. Founded in the self-attention mechanism [2] introduced by the transformer architecture, LLMs such as GPT [3], BERT [4] and their successors have demonstrated an unprecedented ability to understand [5], generate, and reason with human language [6]. These models are pre-trained on vast amounts of text data and fine-tuned for specific tasks [7], enabling them to perform a wide variety of functions, from language translation and summarization to complex reasoning and creative content generation.

Among the numerous strategies developed to further optimize the performance of LLMs, Chain-of-Thought (CoT) prompting [8] has emerged as a particularly impactful technique. CoT prompting builds on the inherent strengths of LLMs by guiding them to reason through problems step by step, mimicking human logical thought processes. This structured approach not only enhances their ability to solve complex problems, but also ensures greater accuracy and coherence in the generated responses. As a result, CoT prompting has proven highly effective in a diverse range of applications, including code generation [9], task planning, knowledge retrieval [10]–[12], and more [13]–[18], solidifying its role as a cornerstone technique in the continued evolution of LLM capabilities.

In tasks of complex reasoning, CoT prompting offers a novel perspective of model reasoning by guiding large language models through a coherent sequence of intermediate steps. To obtain better results, Zhou et al. [19] introduced a prompting strategy by decomposing a complex problem into a series of simpler sub-problems and solving them in sequence. Zheng et al. [20] proposed an abstraction-and-reasoning framework to handle complex tasks involving intricate low-level details.

Although CoT prompting has demonstrated remarkable ability in step-by-step problem solving for various tasks, its inherent linear and straightforward structure constrains its ability to capture intricate contextual information. To address this, various topological variants have been developed, offering more sophisticated approaches to reasoning. Yao et al. [21] proposed the Tree-of-Thought (ToT) prompting, which views each intermediate step as a node in a tree structure. This approach allows the models to self-evaluate the progress of different intermediate thoughts towards the final answer, translating classical insights about problem-solving into efficient and adaptive methods. Furthermore, Besta et al. [22] utilize the Graph-of-Thoughts (GoT) prompting framework, modeling the reasoning process of an LLM as a graph, allowing the model to combine information from multiple nodes for subsequent reasoning steps, enhancing its alignment with human logical reasoning processes.

However, the additional complexity introduced by structural variations multiplies the computational cost of model's reasoning, limiting their practical application in real-world tasks. More importantly, these existing methods fail to consider the potential errors that may be arisen during the reasoning process [23]. Specifically, errors made in intermediate reasoning steps can result in incorrect final answers. These issues remain significant challenges for LLMs in providing accurate answers to complex problems.

Taking the mathematical questions as an example, when confronted with incorrect results, humans instinctively adopt a "backward thinking" strategy, which guiding them meticulously check each deduction step from the end to the beginning, to identify the errors in reasoning process. Inspired by this intuitive traceability mechanism, we propose Fallback Prompting to precisely locate the erroneous reasoning step and subsequently correct it by leveraging a systematic backward

Manuscript received October 17, 2024; revised December 23, 2024. Corresponding author: Yizhi Zhang (yzzhang@snnu.edu.cn).

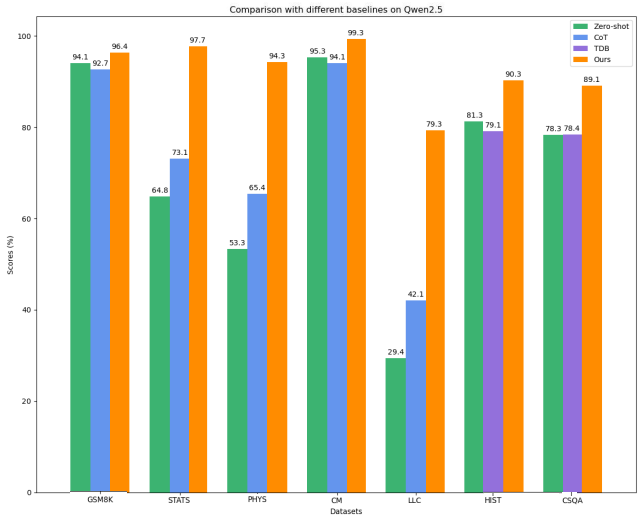


Fig. 1. Comparison on different baselines on Qwen2.5, where we present results of Fallback Prompting (orange) compared with Zero-shot (green), CoT (blue) and TDB (purple) baselines.

tracing mechanism.

Our main contributions are listed as follows:

- Motivated by human activities in error checking, we proposed a new strategy, fallback prompting, including problem evaluation and rectification, to guide LLMs to provide accurate answers in multi-step reasoning tasks.
- Specifically, we conceptualize it as a bottom-up decomposition strategy. Initially, each original question will be broken down into a series of smaller sub-problems and we solve them successively. It' s different when the answer is incorrect. When this occurs, the fallback prompting will guide LLMs to retrace every reasoning step from the last one and halt at the first correct step to commence the rectification process.
- We evaluate our proposed fallback prompting on five datasets across three categories of reasoning tasks, including: (a) Arithmetic reasoning: GSM8K [24], Chinese-school-Math (CM), and MMLU, (b) Symbolic reasoning: Last-letter-concatenation (LLC) [8] and (c) knowledgeable reasoning: Commonsense QA (CSQA) [25]. Experimental results (Fig 1) show that the fallback prompting effectively alleviates the challenges posed by errors in intermediate reasoning steps and ensuring the accuracy and robustness of the overall multi-step reasoning framework. We believe that this technique has the potential to significantly enhance the reliability and efficiency of computational systems engaged in complex reasoning tasks.

The rest of the paper is organized as follows. Section II reviews the current state of research. Section III specifically describes the fallback prompting from both a mathematical and a practical perspective. Section IV presents the experimental results and analysis. Finally, we present the conclusion and outlook of this paper in Section V.

II. RELATED WORK

A. LLMs and emergent ability

The rapid development of Large Language Models (LLMs) and their extensive application across various industries and domains [28]–[30] have significantly showcased their transformative potential, reflecting a shift from natural language understanding to creative content generation and even complex reasoning. One of the most notable aspects of LLMs is their emergent abilities [31]. These abilities refer to the unexpected and often impressive cognitive capabilities exhibited by LLMs when prompted in a specific way, enabling them to solve tasks without the need for model fine-tuning [32]. A key breakthrough in harnessing these emergent abilities is the Chain-of-Thought (CoT) prompting method, introduced by Wei et al. [8]. CoT prompting guides the model through a series of intermediate reasoning steps, mirroring human-like problem-solving strategies. This step-by-step reasoning process [33] not only facilitates the tackling of complex problems that require multi-step thinking but also enhances the transparency and interpretability of the model reasoning. CoT prompting has been shown to significantly improve LLMs' performance across various benchmarks, especially in tasks that demand strong and coherent logical reasoning.

Recently, researchers have expanded on the concept of emergent abilities by exploring how these models can adapt to new tasks with minimal or no retraining [34]–[36]. For example, models such as GPT-4 [3] have demonstrated the ability to perform tasks ranging from arithmetic reasoning to symbolic manipulation, simply by being provided with the correct prompt. This capability is a direct result of the large-scale training these models undergo, which enables them to generalize across a wide range of tasks, often outperforming more specialized systems. The ability of LLMs to demonstrate emergent reasoning abilities is closely related to the concept of in-context learning [37], where the model can use the context of a given task to generate answers without need for task-specific training.

Despite these advances, a critical challenge remains in ensuring the reliability and consistency of LLMs when applied to more complex tasks. Although CoT prompting and other prompting techniques [38] can improve performance, errors still occur, especially in tasks that require high-level reasoning or domain-specific knowledge. This is where our proposed fallback prompting strategy aims to make a significant contribution by enabling LLMs to self-correct and refine their responses iteratively, enhancing their overall performance, and reducing error rates during complex reasoning tasks.

B. Multi-step reasoning

Multi-step reasoning tasks, which involve solving complex problems through a series of interconnected logical steps, present challenges for traditional reasoning methods like Chain-of-Thought (CoT) prompting. Although CoT prompting has been widely recognized for its ability to guide models through sequential reasoning steps, its linear and straightforward structure often struggles to capture the complexity of reasoning needed for multi-step tasks. CoT prompting tends

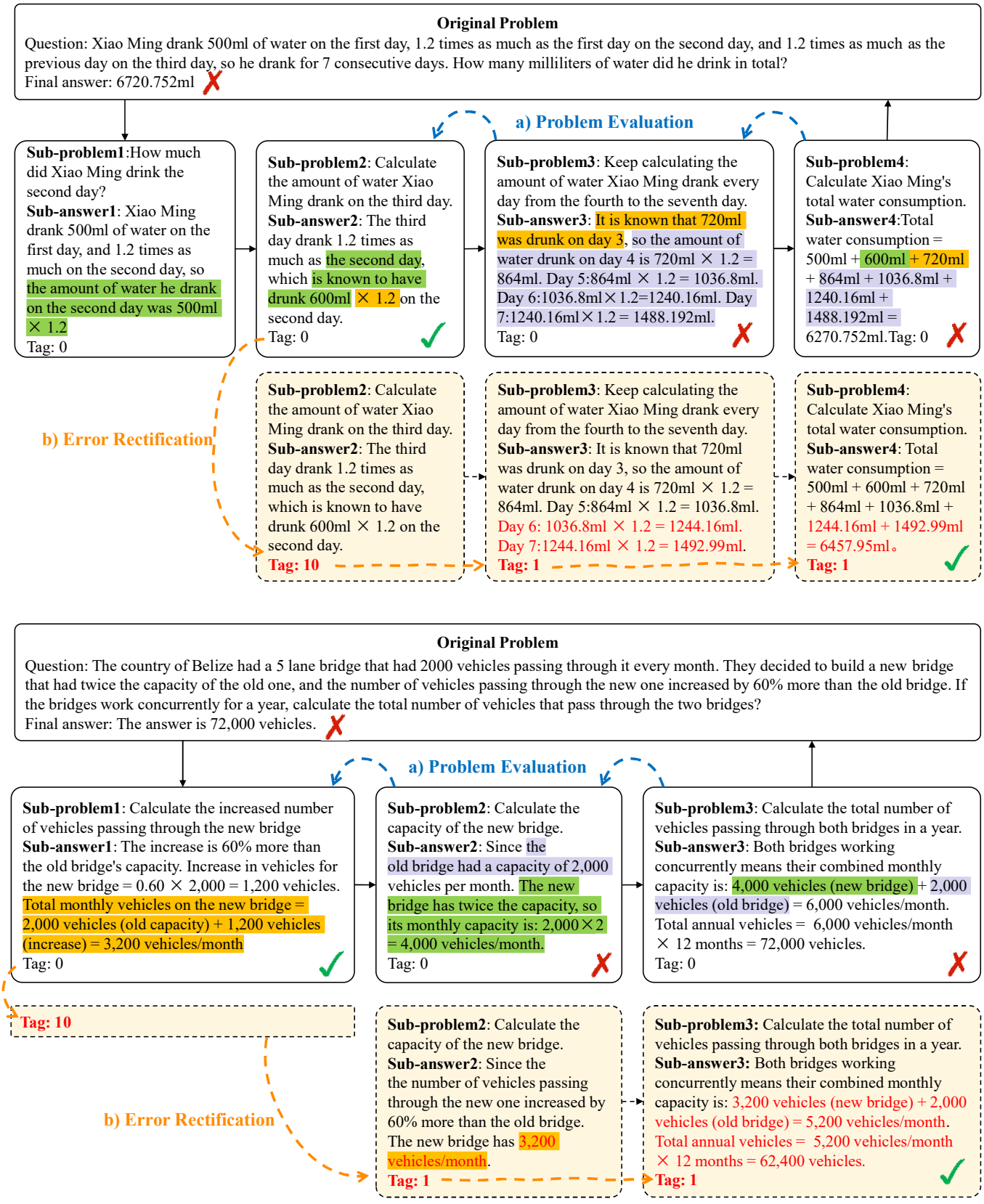


Fig. 2. Examples of Fallback Prompting in arithmetic reasoning tasks in both Chinese and English. We apply this backward-tracing method in two stages, including 1) Problem Evaluation and 2) Error Rectification. *Top*: an example (translate to English) of Chinese School Math [26], where we locate the error at **Sub-problem 3** as a calculation mistake. *Bottom*: an example of GSM8K [27] where the error is identified at **Sub-problem 2**, as it fails to correctly use the result of Sub-problem 1.

to be limited when problems require intricate or nonlinear connections between steps, and its effectiveness diminishes as the complexity of the task increases.

To address these challenges, several strategies have been proposed. Some research address this through task decomposition. the Least-to-Most (L2M) prompting [19] enables the model to break down a complex problem into simpler sub-problems and solve them sequentially. By dividing the problem into smaller tasks, the models can reduce the overall complexity of the original problem, making it easier to reason through each step. However, it may face the challenge of lacking the coherence needed to ensure that the steps are logically connected and that the final solution fully captures the nuances of the original problem. Take-a-Step-Back (TSB) prompting [20] starts by looking for high-level concepts or first principles that underlie the problem, takes a step back before solving the problem and then works downward to solve it. This strategy aims to ensure that the reasoning process remains logically grounded from the beginning, addressing the problem in a more structured and foundational way before diving into specific details. Other studies focus on the selection of demonstrations. Self-Consistency prompting [39] employs a voting mechanism to choose the final reasoning path. By predicting multiple reasoning paths and selecting the one that is most consistent, the model can improve its chances of producing the correct answer. Complex prompting [40], an extension of this approach, enhances the voting system by favoring the most complex reasoning paths among the sampled ones. This helps ensure that the model reasoning process is sufficiently detailed and thorough, especially for complex tasks that require deeper insights.

We position our work in the field of multi-step reasoning [41] and propose a bottom-up decomposition strategy based on reverse reasoning, resulting in more accurate and coherent problem solving.

C. Verify and refine

One of the most significant challenges in multi-step reasoning tasks is the generation of errors and the subsequent propagation of these errors throughout the reasoning process. Errors in the initial reasoning steps can have a cascading effect, causing the model to produce incorrect results in later stages. This issue is particularly problematic for complex tasks where accuracy at each step is crucial for reaching the correct final answer. The inability to trace and correct these errors effectively can result in the model failing to deliver reliable solutions.

In response to this challenge, Sun et al. proposed Iterative-CoT (Iter-CoT) prompting [42]. Iter-CoT aims to address errors generated by zero-shot CoT by introducing a revise prompt. This mechanism allows the model to self-correct its reasoning process by iterating over the reasoning steps, refining the output until the correct result is produced. Although Iter-CoT improves accuracy by enabling self-refinement, it has a potential drawback that during the correction process the model may lose important contextual information from the original problem, which could degrade the quality of the reasoning at later stages. To mitigate the loss of context during error correction, Residual-Connection (ResPrompt) prompting [43] was introduced. ResPrompt enhances the revision process by incorporating the necessary dependencies into the prompt. This approach ensures that the model maintains the relevant contextual information while revising its reasoning path, reducing errors due to missing context. Reconstructing the reasoning process, ResPrompt provides a more reliable method to refine the output and address errors effectively.

Our approach builds upon these methods but focuses on reducing the cost of self-refinement. Instead of allowing the model to iterate through the entire reasoning process to correct errors, our method uses a more targeted strategy. When an error is identified, we backtrack only to the reasoning step where the first error occurred, thus limiting the scope of correction. This approach not only reduces the cost of error correction, but also minimizes the risk of losing context or introducing new errors during the refinement process. By focusing on more efficient and context-preserving error correction, our method aims to offer a more practical solution for improving multi-step reasoning in large language models.

III. Fallback Prompting

As shown in Figure 2, when the LLM is faced with a complex reasoning problem that requires coherent logic (Figure 2, top, *"The total amount of water for seven consecutive days"*) or contains confusing information in the description of the problem (Figure 2, bottom, *"the capacity of the bridge"* and *"the number of vehicles passing through the bridge"*), it may make errors during intermediate steps and Inspired by the human-logical approach of conducting error checking in a backward way, we propose the FALLBACK PROMPTING method.

Before directly addressing the original problem, following Zhou et al. [19], we guide the LLM to decompose the problem into a series of sub-problems and solve them sequentially. For instance, in Figure 2, we decompose *"The total amount of water over seven consecutive days"* into a series of sub-problems for determining the *"accurate amount of water in each day"*. For each sub-problem, the answer to the previous one is added to the current question as context information. This is also the primary cause of error propagation. Building on this process, we additionally record the answer to each sub-problem at every stage and initialize a tag of '0' for each sub-problem, which served as a reference for fallback prompting.

Then, we summarize our fallback prompting into two stages:

1. Problem evaluation. When confronted with incorrect results, we use a backward prompt to guide the LLM in reviewing each deduction step, starting from the last sub-problem and sequentially evaluating the correctness of each answer. A tag of '1' will be assigned if the answer is incorrect, with all sub-problems initially having a tag value of '0'. This process continues until the first correctly answered sub-problem is found, where we update its tag to '10' (here, '10' represents the binary value 2, corresponding to the decimal value).

2. Error rectification. In this stage, we rectify the errors based on the tag values in stage 1. Specifically, starting from

the sub-problem with a tag of '10', we sequentially revise the answers for each subsequent.

Under this backward-tracing mechanism, we can accurately locate errors while alleviating the cost of model reasoning. For we only need to focus on the parts where errors occur, which also aligns with the logic strategy of human thinking. When an error occurs again, our proposed fallback prompting will further narrow the search space for error localization by focusing on sub-problems that are tagged with values other than '0'.

We designed fallback prompting as a plug-and-play strategy that, in theory, can be integrated with any existing CoT strategies to enhance the accuracy of reasoning results.

In the following sections, we present empirical study results of fallback prompting on a range of reasoning tasks covering arithmetic, symbolic and knowledgeable reasoning tasks.

IV. Experiments and results

A. Experimental settings

Here we define the tasks and models we experiment with. We also describe our evaluation metric and the baseline methods we consider.

1) Tasks and datasets

We evaluate our proposal on 7 datasets from three categories of reasoning tasks including arithmetic, symbolic and knowledgeable reasoning tasks.

TABLE I

Datasets used in this paper

Domain	Dataset	Numbers
Arithmetic reasoning	GSM8K	100
	Stats	100
	Phys	100
	CM	100
Symbolic reasoning	LLC	100
knowledgeable reasoning	HIST	100
	CSQA	100

For arithmetic reasoning, we evaluate on problems that need multi-step to solve.

- GSM8K [27]: Grade school math word problems which take between 2 and 8 steps to solve using basic arithmetic operations. We randomly selected 100 instances as our testbed and compared their results with CoT prompting.
- MMLU [44]: Multi-choice questions collected from various examinations of 57 different subjects. Here we select high-school statistic (Stats) and high-school physic (Phys) as they need coherent and logical reasoning.
- CM [26]: Chinese school math quizzes and answers generated by BELLE [45], with multi-steps to solve. As the prior works merely focused on evaluations using English datasets, results on Chinese datasets remain largely under-explored.

For symbolic reasoning, we use Last-letter-concatenation (LLC) [8], that asks the model to concatenate the last letters of each word (e.g., the input is "Elon Musk" and the output should be "nk").

For knowledgeable reasoning, we evaluate on MMLU high-school European history (HIST), where the question contains rich and lengthy background information, and Commonsense QA (CSQA) [46] that asking questions with complex semantics that often require reasoning based on prior knowledge.

2) Model selection

We selected three open-source state-of-the-art bilingual large language models:

- Llama3 [47]: the most capable openly available LLM to date of Meta. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) [48] to align with human preferences for helpfulness and safety.
- Qwen2.5 [49]: A collection of Alibaba Group's latest LLM family, including specialized models for coding, Qwen2.5-Coder [50], and mathematics, Qwen2.5-Math [51]. There is extra support for Chinese of Qwen2.5-Math by endowing it with the abilities to perform Chain-of-Thought(CoT), Program-of-Thoughts(PoT) [52], and Tool-integrated-Reasoning(TIR) [53].
- GLM4 [54]: We choose a chat-optimized version of GLM-4-9B, that trained on a multilingual corpus with a context length of 8K tokens, and capable of extended text reasoning up to 128K tokens.

For all LLMs, we set a unified generation configuration for fair comparison: temperature is set to 0.8 and the top-k is set to 5. All experiments are done in the same computation environment with 1 NVIDIA 24GB RTX3090 GPU.

3) Baseline methods

We compare with zero-shot prompting and standard Chain-of-Thought prompting on arithmetic reasoning tasks, where we directly ask the LLM to answer the question or give the simple *"Let's think step by step"* instruction [33]. For more challenging reasoning tasks, we selected the In-context learning (ICL) prompting [55], [56], using a few (k-shots) QA pairs as demonstrations to guide the LLM without additional training. We also compare our method with TDB prompting [57], an enhanced zero-shot prompting with *"Take a deep breath and work on this problem step by step."* added at the beginning of the question. All methods in our experiments rely on greedy decoding for inference.

4) Evaluation metrics

We randomly sample 100 instances for each task and compare them with different baselines. We consider two metrics for evaluation: (1) Traditional text evaluation metrics: where we directly compare completeness and consistency of targets and model predictions, and (2) Model evaluation metrics: where we guide LLM to identify equivalence between targets and model predictions with given evaluation prompts.

We use the value of accuracy as evaluation indicator, with the formula:

TABLE II
COMPARISON WITH DIFFERENT BASELINE METHODS ON ARITHMETIC REASONING TASKS

Baseline	Model	GSM8K	Stats	Phys	CM	LLC
Zero-shot	GLM4	86.5	40.5	53.4	87.1	38.7
	Llama3	76.8	38.4	38.6	85.8	65.6
	Qwen2.5	94.1	64.8	53.3	95.3	29.4
CoT	GLM4	91.2	58.1	48.7	91.7	36.2
	Llama3	77.9	39.1	38.2	87.1	66.3
	Qwen2.5	92.7	73.1	65.4	94.1	42.1
Ours	GLM4	**98.1**	92.7	92.5	99.1	76.3
	Llama3	86.3	83.4	91.1	97.7	87.7
	Qwen2.5	96.4	**97.7**	**94.3**	**99.3**	**79.3**

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (1)$$

where TP (True Positives) represents samples that are correctly predicted as positive, FP (False Positives) represents samples that are incorrectly predicted as positive, FN (False Negatives) represents samples that are incorrectly predicted as negative, and TN (True Negatives) represents samples that are correctly predicted as negative.

B. Results analysis

Table II and III provide a comprehensive comparison between our proposed method and baseline approaches across various reasoning tasks. In the tables, the **bolded values** represent the best results achieved by our method on each dataset, while the underlined values indicate the best results achieved by other each baselines in each dataset. The results clearly show that the proposed method consistently outperforms the baselines, indicating its effectiveness in handling complex reasoning tasks.

For **Arithmetic Reasoning** and **Symbolic Reasoning** tasks (Table II), the results demonstrate that fallback prompting achieves the highest accuracy across all datasets and models, highlighting its robustness in handling arithmetic reasoning tasks. Specifically, for English mathematical reasoning datasets, GSM8K and Stats, fallback prompting consistently outperforms the Zero-shot and CoT baselines, with particularly significant improvements observed on Stats. In the Chinese mathematical reasoning dataset, fallback prompting achieves near-perfect accuracy, reaching a maximum of 99.3% with Qwen2.5. Although the results on Phys and LLC are less satisfactory under several baselines, fallback prompting still significantly improves their accuracy, demonstrating its effectiveness even in challenging scenarios.

For **Knowledge Reasoning** tasks (Table III), our proposed method demonstrates exceptional performance, particularly with GLM4 and Qwen2.5. For instance, with GLM4, it achieves 91.7% on HIST, surpassing the next-best baseline, ICL (80.1%), by more than 11 percentage points. Notably, the highest accuracy is achieved on CSQA with GLM4, reaching 97.3%.

TABLE III
COMPARISON WITH DIFFERENT BASELINE METHODS ON OTHER REASONING TASKS

Baseline	Model	HIST	CSQA
Zero-shot	GLM4	73.5	88.1
	Llama3	62.6	74.8
	Qwen2.5	81.3	78.3
ICL	GLM4	80.1	88.6
	Llama3	64.6	74.1
	Qwen2.5	72.8	72.7
TDB	GLM4	80.3	89.2
	Llama3	63.2	62.6
	Qwen2.5	79.1	78.4
Ours	GLM4	**91.7**	**97.3**
	Llama3	88.2	87.5
	Qwen2.5	90.3	89.1

Moreover, we are surprised to observe the powerful capability of Qwen2.5 in handling complex mathematical problems. When comparing the results across both tables, under identical experimental settings, Qwen2.5 consistently outperforms GLM4 and Llama3. Notably, its performance on arithmetic reasoning datasets is significantly better than on symbolic reasoning and knowledgeable reasoning datasets. Furthermore, the results of Llama3 in different four baselines are relatively similar, suggesting a potential limitation in its performance variability in our selected baselines.

V. CONCLUSION

In this study, we introduced a novel strategy, fallback prompting, to alleviate the challenge of error evaluation and

rectification in the reasoning processes of Large Language Models (LLMs). By leveraging a feedback-driven backward evaluation mechanism, our method enables LLMs to refine their reasoning capabilities iteratively.

Through extensive experiments on seven data sets for three reasoning tasks: arithmetic reasoning, symbolic reasoning, and knowledgeable reasoning, our approach demonstrated state-of-the-art performance. In particular, fallback prompting achieved a near perfect accuracy of 99. 3% on the Chinese school Math dataset using Qwen2.5, and significantly improved results on symbolic reasoning (e.g., 91. 7% on HIST) and knowledgeable reasoning (e.g., 97. 3% on CSQA) with GLM4. These improvements were consistently observed across multiple datasets and baseline comparisons, highlighting the robustness and adaptability of the proposed method.

Furthermore, our analysis revealed the exceptional performance of Qwen2.5 in handling complex mathematical tasks, significantly outperforming GLM4 and Llama3 under identical experimental conditions. However, the relatively stable performance of Llama3 at various baselines indicates potential limitations in its capacity to vary performance.

In summary, fallback prompting offers an efficient and effective solution for improving the reasoning accuracy of LLMs by dynamically addressing errors. This work lays the foundation for future research in improving the reasoning capabilities of LLMs, with potential applications in education, problem-solving, and other domains requiring reliable multi-step reasoning.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities(6142103190207), and the Digital Education Research and Practice Project of Shaanxi Normal University (JYSZH201301).

REFERENCES

[1] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, vol. 3781, 2013.

[2] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[5] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *Advances in neural information processing systems*, vol. 32, 2019.

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[7] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *ACL 2018-56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1. Association for Computational Linguistics, 2018, pp. 328–339.

[8] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[9] J. Li, G. Li, Y. Li, and Z. Jin, "Structured chain-of-thought prompting for code generation," *ACM Transactions on Software Engineering and Methodology*, 2023.

[10] K. Stechly, K. Valmeekam, and S. Kambhampati, "Chain of thoughtlessness: An analysis of cot in planning," *arXiv preprint arXiv:2405.04776*, 2024.

[11] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *arXiv preprint arXiv:2305.04091*, 2023.

[12] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, and Z. Hu, "Reasoning with language model is planning with world model," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 8154–8173.

[13] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 10 014–10 037.

[14] O. Yoran, T. Wolfson, B. Bogin, U. Katz, D. Deutch, and J. Berant, "Answering questions by meta-reasoning over multiple chains of thought," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 5942–5966.

[15] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[16] A. Song, J. Fu, X. Mu, X. Zhu, and K. Cheng, "L-secnet: Towards secure and lightweight deep neural network inference," *Journal of Networking and Network Applications*, vol. 3, no. 4, pp. 171–181, 2024.

[17] W. He, P.-H. Ho, D. Wang, and L. Xiao, "Efficient beacon deployment for large-scale positioning," *Journal of Networking and Network Applications*, vol. 1, no. 2, pp. 40–48, 2021.

[18] N. Ho, L. Schmid, and S. Yun, "Large language models are reasoning teachers," in *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*. Association for Computational Linguistics (ACL), 2023, pp. 14 852–14 882.

[19] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[20] H. S. Zheng, S. Mishra, X. Chen, H.-T. Cheng, E. H. Chi, Q. V. Le, and D. Zhou, "Take a step back: Evoking reasoning via abstraction in large language models," *arXiv preprint arXiv:2310.06117*, 2023.

[21] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[22] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk *et al.*, "Graph of thoughts: Solving elaborate problems with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 682–17 690.

[23] Z. Chu, J. Chen, Q. Chen, W. Yu, T. He, H. Wang, W. Peng, M. Liu, B. Qin, and T. Liu, "Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 1173–1203.

[24] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[25] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," in *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, 2019, p. 4149.

[26] Y. Ji, Y. Deng, Y. Gong, Y. Peng, Q. Niu, L. Zhang, B. Ma, and X. Li, "Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases," *arXiv preprint arXiv:2303.14742*, 2023.

[27] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

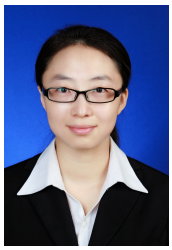
[28] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Transactions on Knowledge and Data Engineering*, 2024.
[29] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[31] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.
[32] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.
[33] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.
[34] T. Wu, W. Yuan, O. Golovneva, J. Xu, Y. Tian, J. Jiao, J. Weston, and S. Sukhbaatar, "Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge," *arXiv preprint arXiv:2407.19594*, 2024.
[35] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3045–3059.
[36] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," *arXiv preprint arXiv:2110.07602*, 2021.
[37] R. Hendel, M. Geva, and A. Globerson, "In-context learning creates task vectors," *arXiv preprint arXiv:2310.15916*, 2023.
[38] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff *et al.*, "The prompt report: A systematic survey of prompting techniques," *arXiv preprint arXiv:2406.06608*, 2024.
[39] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *The Eleventh International Conference on Learning Representations*.
[40] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity-based prompting for multi-step reasoning," in *The Eleventh International Conference on Learning Representations*, 2022.
[41] C. Zhou, W. You, J. Li, J. Ye, K. Chen, and M. Zhang, "Inform: Information entropy based multi-step reasoning for large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 3565–3576.
[42] J. Sun, Y. Luo, Y. Gong, C. Lin, Y. Shen, J. Guo, and N. Duan, "Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 4074–4101.
[43] S. Jiang, Z. Shakeri, A. Chan, M. Sanjabi, H. Firooz, Y. Xia, B. Akyildiz, Y. Sun, J. Li, Q. Wang *et al.*, "Resprompt: Residual connection prompting advances multi-step reasoning in large language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 5784–5809.
[44] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.
[45] BELLEGroup, "Belle: Be everyone's large language model engine," https://github.com/LianjiaTech/BELLE, 2023.
[46] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "Commonsenseqa: A question answering challenge targeting commonsense knowledge," *arXiv preprint arXiv:1811.00937*, 2018.
[47] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.
[48] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.
[49] Q. Team, "Qwen2. 5: A party of foundation models," *Qwen (Sept. 2024). url: https://qwenlm. github. io/blog/qwen2*, vol. 5, 2024.
[50] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Dang *et al.*, "Qwen2. 5-coder technical report," *arXiv preprint arXiv:2409.12186*, 2024.
[51] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin *et al.*, "Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement," *arXiv preprint arXiv:2409.12122*, 2024.
[52] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *arXiv preprint arXiv:2211.12588*, 2022.
[53] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, M. Huang, N. Duan, and W. Chen, "Tora: A tool-integrated reasoning agent for mathematical problem solving," *arXiv preprint arXiv:2309.17452*, 2023.
[54] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Rojas, G. Feng, H. Zhao, H. Lai *et al.*, "Chatglm: A family of large language models from glm-130b to glm-4 all tools," *arXiv preprint arXiv:2406.12793*, 2024.
[55] A. Cattan, A. Jacovi, A. Fabrikant, J. Herzig, R. Aharoni, H. Rashkin, D. Marcus, A. Hassidim, Y. Matias, I. Szpektor *et al.*, "Can few-shot work in long-context? recycling the context to generate demonstrations," *arXiv preprint arXiv:2406.13632*, 2024.
[56] D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei, "Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers," in *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.
[57] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=Bb4VGOWELI

Jianing Sun received her B.Sc degree in School of Computer Science from Shaanxi Normal University, Xian, China, in 2022. She is currently pursuing the Master degree in school of Computer Science with Shaanxi Normal University, Xian, China. Her research focus on artificial intelligence in Education.

Zhichao Zhang received his B.S. and M.S degrees from Harbin Institute of Technology, Heilongjiang China in 2006 and 2011 respectively. Now he is presently working as a lecturer at the Hainan Normal University, Haikou, China. He is currently working toward the D.S. degree with the School of Computer Science, Shaanxi Normal University, Shaanxi, China. His research interests include big data, machine learning, and artificial intelligence.

Xiaopu Wang received his B.Sc. degree in Computer Science from the School of Computer Science, Shaanxi Normal University, Xi'an, China, in 2021. He is currently pursuing a Master's degree in Computer Science at Shaanxi Normal University, Xi'an, China. His research interests include Visual Question Answering and Knowledge Graph Reasoning.

Xinyuan Ji received her B.Sc degree from Luoyang Normal University, Luoyang, China, in 2024. She is currently pursuing the Master degree in school of Computer Science with Shaanxi Normal University, Xian, China. Her research focus on scene computing in Education.



Yizhi Zhang received her M.S degree from Xi'an Jiaotong University, Xian, china, in 2015. Now she is a senior engineer at Shaanxi Normal University, Xian, China. Her research interests include educational digitization and artificial intelligence.