Enhancing Prompting with Deep Understanding and Extended Reasoning for Solving Mathematical Problems

Zhiwei Yang[1(✉)], Rongxin Huo[1], Jiahua Yang[1], Longtao Wang[1], and Yuxuan Zhou[2]

[1] Guangdong Institute of Smart Education, Jinan University, Guangzhou, China
yangzw@jnu.edu.cn
[2] College of Computer Engineering, Jiangsu University of Technology, Changzhou, China

Abstract. With the emergence of chain-of-thought (CoT) reasoning, large language models (LLMs) have demonstrated substantial potential in tackling multi-step reasoning tasks. However, existing CoT-based approaches primarily emphasize either semantic understanding or the enhancement of the reasoning process, with limited efforts to address both aspects concurrently. As a result, these methods struggle with suboptimal performances due to semantic misunderstanding, calculation errors, or step-missing errors when solving complex mathematical problems. In this paper, we propose a novel Deep Understanding and Extended Reasoning (DUER) prompting method for solving mathematical problems, which seamlessly integrates semantic comprehension with reasoning enhancement. Specifically, DUER samples the most complex examples from the training set, and utilizes a deep understanding module to extract the core question and key information from mathematical problems and an extended reasoning module to generate more detailed reasoning steps for these examples. Finally, these enhanced examples containing the core question and information are used to prompt LLMs to infer the answer to the test question. Extensive experiments on 9 reasoning benchmarks show that our method achieves competitive performances on all datasets and establishes new state-of-the-art on GSM8K, SVAMP, AQuQ, and MATH benchmarks. The code is available at: https://github.com/Nicozwy/DUER.

Keywords: Math Problem Reasoning · Few-shot Prompting · Large Language Models

1 Introduction

LLMs have exhibited remarkable emerging abilities, revolutionizing the paradigm in tackling complex natural language processing tasks [1]. However, existing studies demonstrate that scaling up the size of LLMs could not further enhance their performance in handling complex tasks, especially those requiring multi-step reasoning and logical inference [20]. To address this issue, [20] introduces a simple

T. Taniguchi et al. (Eds.): ICONIP 2025, LNCS 16309, pp. 336–349, 2026.
https://doi.org/10.1007/978-981-95-4367-0_23

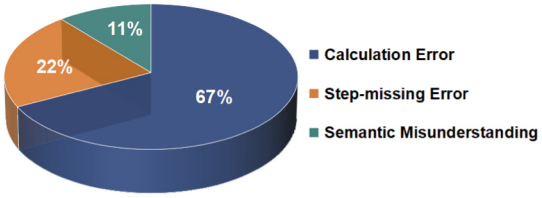

Fig. 1. Error answer analysis of 1,319 GSM8K questions using the Manual-CoT method with GPT-3.5-turbo.

CoT prompting to enable step-by-step reasoning in solving a task, significantly eliciting the reasoning ability of LLMs. Compared with full fine-tuning, CoT-based methods improve performance with higher sample efficiency, thus attracting widespread attention. Although existing CoT-based methods contribute to solving complex mathematical problems, they primarily emphasize either semantic understanding or the enhancement of the reasoning process, leading to suboptimal performances. Therefore, a more effective approach to seamlessly integrating both aspects is urgently needed. For solving mathematical problems, existing methods can be broadly categorized into two types: 1) Fine-tuning-based and 2) Prompt-based. Fine-tuning-based approaches achieve promising results at the cost of large-high-quality annotated datasets and substantial computation [12]. In contrast, prompt-based methods can elicit the inherent reasoning abilities of existing open-source or closed-source LLMs with high sample efficiency. For example, [20] introduces CoT by simply adding intermediate steps to prompt LLMs to improve their performance on reasoning tasks, [4] enhances multi-step reasoning with complexity-based prompting by selecting few-shot examples containing chains of more reasoning steps. However, they only study semantic understanding or reasoning processes, failing to effectively combine their merits for solving problems.

As shown in Fig. 1, GPT-3.5-turbo with Manual-CoT on GSM8k shows error distribution: *calculation* (67%), *step-missing* (22%), and *semantic* (11%). Recent approaches have explored enhancements from the perspectives of semantic understanding and reasoning augmentation, respectively. For example, [24] explores extracting key information to reduce semantic understanding errors. In addition, [9] reveals the strong correlation between the effectiveness of CoT and the length of reasoning chains, demonstrating that extending the reasoning steps for few-shot examples in prompts can significantly enhance the reasoning capabilities of LLMs, thus reducing step-missing and calculation errors. However, a critical question remains: *how to jointly address all error types for better complex reasoning?*

To this end, we propose a prompting method combining Deep Understanding and Extended Reasoning (DUER) to enhance LLMs for solving math problems, consisting of three stages, i.e., complex sampling, demo construction, and inference. Specifically, 1) DUER utilizes complex sampling to select the most complex

examples from training set based on their initial reasoning length. 2) Then DUER leverages deep understanding (D-U) to extract core problem-solving information based on LLMs and utilizes extended reasoning (E-R) to detail reasoning steps based on complex samples through LLMs, thus constructing k enhanced examples for prompting. 3) Finally, the enhanced examples with the extracted core information are used to prompt LLMs to infer the answers to the test questions.

We evaluate our method on 11 benchmarks and experimental results demonstrate that DUER significantly outperforms other baselines on GSM8K, SVAMP, AQuQ, and MATH benchmarks. The contributions of this paper are as follows:

- We propose a novel deep understanding and extended reasoning (DUER) prompting method, which enhances LLMs' ability to solve complex reasoning problems and effectively reduces three main errors in reasoning.
- DUER's deep understanding and extended reasoning modules are plug-and-play, allowing seamless adaptation to various models and datasets.
- Extensive experimental results show that DUER surpasses other counterparts by a large margin, achieving new state-of-the-art (SOTA) results on GSM8K, SVAMP, and AQuA.

2 Related Works

2.1 In-Context Learning

LLMs have seen significant improvements in reasoning performance by combining in-context learning (ICL) with CoT. ICL enables LLMs to perform target tasks by incorporating a few example prompts as part of the input. Without gradient updates, ICL allows a single model to universally perform a variety of tasks. Several research directions have been explored to enhance the performance of ICL: (1) obtaining the most effective prompts [26], such as dynamically retrieving related examples for a given test input, [23] proposes a solution: by leveraging LLMs to generate reasoning chains autonomously, and then applying the K-means algorithm to cluster complex problems, [23] selects the most representative samples to construct Few-Shot demonstrations; (2) augmenting with fine-grained information, such as incorporating task instructions [10]. However, How to construct higher-quality examples remains a challenge. There remains substantial potential for improving LLMs' reasoning abilities in complex tasks.

2.2 Semantic Understanding Enhancement

Large models have achieved remarkable success in natural language understanding and generation, but there has been limited research on how to better understand mathematical problems and improve problem-solving accuracy. In a simple manner, [21] shifts the focus to the input by re-reading the question, processing it twice, thereby enhancing the understanding process. This approach is highly versatile and produces significant results. [25] proposed a method to decompose a problem into subproblems, thereby reducing the complexity of language understanding. While effective, these semantic enhancement methods struggle with complex reasoning tasks, where enhanced semantic understanding is inadequate.

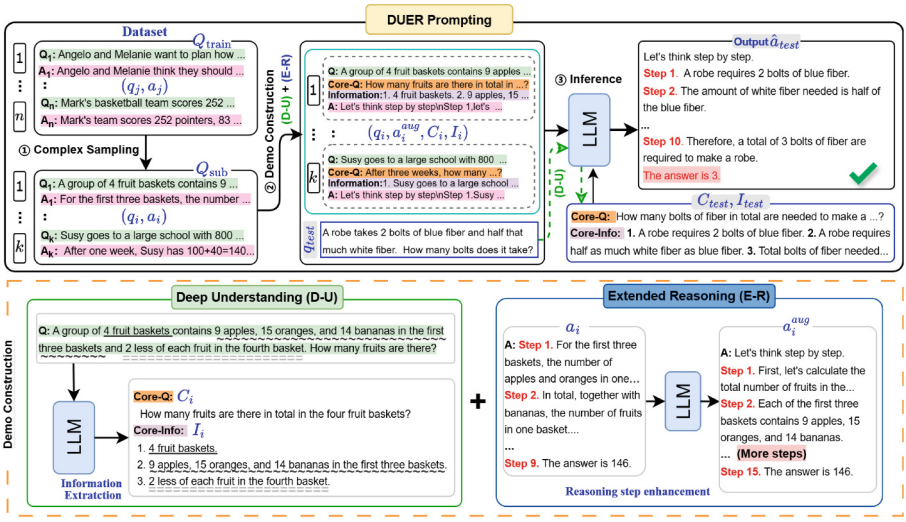

Fig. 2. Overview of our DUER prompting consisting of three stages. 1) **Complex Sampling**: Selecting complex examples containing more reasoning steps from the dataset. 2) **Demo Construction**: Constructing k enhanced demos on complex samples using the D-U and E-R modules. 3) **Inference**: Using the D-U module to extract the core question and problem-solving information of the test question and prompt LLM with the k enhanced demos for reasoning. Core-Q denotes the core question, and Core-Info denotes the core information.

2.3 Complex-Based Prompting

The CoT-based few-shot method has achieved remarkable results on current popular LLMs. However, selecting appropriate samples as prompts remains a challenging issue. One effective approach is complexity-based prompting. [4] explored which reasoning examples make the most effective prompts ICL. By selecting more complex reasoning chains (those with more reasoning steps) within the chain of thought, the model's performance gain a significant improvement. [9] investigated the correlation between the effectiveness of CoT and the length of reasoning steps in prompts, concluding that the performance of LLMs is positively correlated with the length of the reasoning chain within a certain range. Complex tasks benefit significantly from longer reasoning sequences. Still, There is no unified standard for constructing complex samples that provide rich information for contextual learning in LLMs. But this points the way for our research, emphasizing the importance of constructing complex examples with longer reasoning chains to better leverage contextual information and enhance the model's reasoning capabilities.

Therefore, our work is inspired to develop a novel few-shot prompting method that leverages deep semantic understanding and reasoning path enhancement to effectively improve the reasoning ability of LLMs in solving math problems.

3 Proposed Method

Overview. Given a training dataset $Q_{\text{train}} = \{(q_j, a_j)\}_{j=1}^{n}$ where q_j denotes the question and a_j denotes the answer with intermediate steps, we first select a subset from Q_{train}, i.e., $Q_{\text{sub}} = \{(q_i, a_i)\}_{i=1}^{k}$, where $|a_i| \geq |a_m|, \forall (q_m, a_m) \in \{Q_{\text{train}} - Q_{\text{sub}}\}$, and $|\cdot|$ denotes the number of reasoning steps. Then each sample is processed by deep understanding $f_{\text{D-U}}$ and reasoning $f_{\text{E-R}}$, i.e., $f_{\text{D-U}} : f(q_i) \rightarrow \{q_i, C_i, I_i\}_{i=1}^{k}$, where C_i represents the core question of q_i, I_i represents the key information of q_i, and $f_{\text{E-R}} : f(a_i) \rightarrow \{a_i^{\text{aug}}\}_{i=1}^{k}$. Formally, the test question $f_{\text{D-U}}(q_{\text{test}}) \rightarrow \{q_{\text{test}}, C_{\text{test}}, I_{\text{test}}\}$ and combine it with the enhanced samples to obtain the prompt: $Prompt = \{(q_i, a_i^{\text{aug}}, C_i, I_i)\}_{i=1}^{k} + \{q_{\text{test}}, C_{\text{test}}, I_{\text{test}}\}$, aiming to guide LLMs reasoning towards the final answer $\hat{a}_{test}$.

3.1 Deep Understanding (D-U)

Complex reasoning problems often contain noisy or redundant information that can obscure the key elements necessary for accurate reasoning based on LLMs. Obviously, misunderstanding the question or failing to obtain key information will lead to incorrect results. To address this issue, we argue that two critical aspects should be considered, i.e., 1) Core question to be solved and 2) Problem-solving information.

Inspired by [24], we adopt a two-step prompting for information extraction. Specifically, we first append the prompt *"Please extract the core question, only extract the most comprehensive and detailed one !"* at the end of the question to guide LLMs to extract the core question. Then we use the prompt *"Note: Please extract the problem-solving information related to the core question [Core Question], only extract the most useful information, list them one by one !"* to generate a list of key information that aids in problem-solving, where '[Core Question]' refers to the identified question in the first step. Thus, we can capture all relevant problem-solving information while filtering out the irrelevant.

3.2 Extended Reasoning (E-R)

In few-shot context learning, constructing high-quality samples is crucial to enhance the performance of LLMs [2]. Inspired by [4], we select complex samples from the training set according to the length of the reasoning chain and then using the prompt *"Break down the reasoning process into smaller steps, adding as many reasoning steps as possible while ensuring correctness"* to extend the reasoning chain. Based on the original reasoning chain, LLMs generate a more complex and detailed counterpart. According to human problem-solving principles, skipping steps often leads to overlooked errors [17]. Only by providing the most detailed reasoning steps can we minimize the step-missing and calculation errors during the process.

3.3 Demo Construction and Inference

We apply both D-U and E-R modules to enhance complex samples, constructing a few-shot prompt enriched with problem-related information and more detailed reasoning steps. For a given multi-step reasoning problem (i.e., Test-Q in Fig. 2), we apply the D-U technique to extract the relevant key information (i.e., Core-Q and Core-Info in Fig. 2) and integrate it with the previously enhanced few-shot samples, thus constructing a complete prompt. Thus, the prompts enhanced by DUER can ultimately improve the final performance in complex reasoning.

4 Experiments

4.1 Experimental Setup

Datasets. To verify the effectiveness of the proposed method, we conducted experiments across 9 reasoning benchmarks.

- **Arithmetic reasoning.** (1) **GSM8K** [3] contains grade-school math problems requiring 2-8 reasoning steps, each with step-by-step solutions. (2) **SVAMP** [14] provides math word problems with 1-2 expressions and one unknown variable, generated by modifying questions from ASDiv-A [13]. (3) **MultiArith** [15] features multi-step math problems requiring 3-5 operations, covering addition, multiplication, and grouping. (4) **AddSub** [7] contains addition and subtraction problems, solvable in 1-2 steps with clear problem-to-equation mappings. (5) **AQuA** [14] presents challenging arithmetic word problems with multiple-choice options. (6) **SingleEq** [11] is a dataset of algebra word problems solvable by a single linear equation in 1-3 steps, with solutions detailing variable assignment and equation derivation. (7) **MATH** [6] benchmarks advanced problem-solving in subjects like algebra, calculus, geometry, and number theory, with detailed solutions.
- **Commonsense reasoning.** (8) **CommonsenseQA** [16] is a multiple-choice dataset testing commonsense reasoning, covering causality, functionality, and real-world implications, with annotated answer explanations. (9) **StrategyQA** [5] is an open-domain QA benchmark requiring implicit reasoning, where answers are derived through strategic inference. Each example includes a question, reasoning steps, and supporting evidence.

Baselines. We compare our DUER with the following baselines, including zero-shot and few-shot approaches:

- **Zero-shot CoT** [10] enhance LLMs by adding "Let's think step by step" after each question.
- **Zero-shot PS+** [18] comprises devising a plan to divide the task into smaller subtasks and then carrying out the subtasks according to the plan.
- **RCoT** [22] detects factual errors by reconstructing problems from LLM-generated solutions and comparing them with originals, then provides corrective feedback to improve responses.

Table 1. Accuracy (%) on Arithmetic Reasoning benchmarks with GPT-3.5-turbo. The best results are in bold. Underline denotes the second best. Results marked with "*" are taken from the original publications. "–" indicates that the result was not reported in the original publication. SV denotes SVAMP, GS denotes GSM8K, Ad denotes AddSub, Mu denotes MultiArith, AQ denotes AQuA, Si denotes SingleEq, Δ denotes the average performance improvement or decline regarding Zero-shot CoT.

Setting	Method	Arithmetic Reasoning						Score	
		SV	GS	Ad	Mu	AQ	Si	Avg.	Δ
Zero-shot	CoT	79.7	78.0	85.3	95.7	63.0	94.1	82.6	0
	PS+	84.5	81.6	91.6	96.3	64.6	96.4	85.8	+3.2
	RCoT*	79.6	82.0	87.1	–	55.5	91.4	79.1	-3.5
	DUP	82.8	82.8	91.9	96.5	66.1	96.8	86.2	+3.6
Few-shot	Manual-CoT	82.3	81.4	92.6	**99.1**	61.8	97.4	85.8	+3.2
	Auto-CoT	84.7	82.2	**94.4**	**99.1**	59.0	**97.4**	86.1	+3.5
	RCoT*	84.9	84.6	88.2	–	57.1	93.0	81.6	−1.0
	DUER (ours)	**87.5**	**85.1**	91.6	98.0	**66.5**	97.0	**87.6**	+5.0

- **Zero-shot DUP** [24] enhances LLMs' problem comprehension and key information extraction, reducing semantic errors.
- **Manual-CoT** [20] employs eight CoT exemplars as prompts, with each exemplar incorporating a series of intermediate reasoning steps.
- **Auto-CoT** [23] uses k-means to cluster questions, sorts them by distance to cluster centers, and builds a demonstration per group.

Implementations. We evaluate our method based on three LLMs, i.e., GPT-3.5-turbo (0125), GPT-4o-mini (0718), and GPT-4o (0806), over 11 benchmarks. The experimental setup of DUER consists of three stages, where complex samples with longer reasoning chains are sampled heuristically in the first stage, GPT-4o is employed for D-U and E-R modules in the second stage, and the same LLM is used for both information extraction and model inference in the third stage for simplicity. For better performance, we recommend setting up GPT-4o-mini for information extraction, GPT-4o for reasoning, and self-consistency for augmentation. All models are evaluated through their official API [2,8] with *accuracy* as the evaluation metric. We apply the greedy decoding strategy with a temperature of 0 to all baselines. For the few-shot prompting baselines, we follow the recommended number of demonstration examples as specified in their original papers.

4.2 Main Results

We compare our method with four zero-shot and three few-shot Baselines for GPT-3.5-turbo, and we only compare with two few-shot methods for GPT-4o due to high cost, which is depicted as follows:

Table 2. Accuracy (%) on Arithmetic Reasoning benchmarks with GPT-4o. The best results are in bold. Underline denotes the second best. SV denotes SVAMP, GS denotes GSM8K, AQ denotes AQuA, alg denotes algebra, cnt denotes counting and probability, geo denotes geometry, ialg denotes intermediate algebra, nt denotes number theory, palg denotes prealgebra, pcalc denotes precalculus, Δ denotes the average performance improvement or decline regarding Manual-CoT.

Method	SV	GS	AQ	MATH							Score	
				alg	cnt	geo	ialg	nt	palg	pcalc	Avg.	Δ
Manual-CoT	94.4	94.7	**85.0**	92.3	79.3	61.7	**61.7**	81.7	87.7	57.0	79.6	0
Auto-CoT	94.4	95.3	82.6	92.7	**80.7**	**64.0**	59.7	**82.0**	87.7	61.3	80.0	+0.4
DUER (ours)	**94.9**	**96.0**	**85.0**	**93.0**	**80.7**	63.0	60.3	80.7	**90.0**	**62.0**	**80.5**	+0.9

Arithmetic Reasoning. Table 1 presents the evaluation results of model GPT-3.5-turbo on six arithmetic benchmarks. In general, the few-shot methods outperform the zero-shot methods due to their ability to leverage a small amount of task-specific examples, which helps the model better understand the nuances and patterns of the task. On average, our proposed DUER exceeds Auto-CoT by 1.5% across the six benchmarks, showcasing the effectiveness of DUER. For the more challenging benchmarks SVAMP and GSM8K that contain richer question information, more detailed reasoning and precise calculations are required. Our method achieves accuracy rates of 87.5% and 85.1% on these two datasets, respectively. Encouragingly, Zero-shot PS+ and Zero-shot DUP achieve comparable or better performance than Manual-CoT on GSM8K, SVAMP, and AQuA benchmarks, highlighting their effective prompt designs for refined reasoning process or semantic understanding.

In the few-shot setting, Auto-CoT performs well due to its construction of more representative samples, which enables more accurate reasoning, and RCoT excels by incorporating automatic error correction during the reasoning process, which further enhances its performance. DUER further outperforms these methods because we thoroughly interpret the problem's meaning and provide the most detailed reasoning examples for LLMs.

As shown in Table 2, we investigate the performance of different methods on challenging reasoning benchmarks, i.e., SVAMP, GSM8K, AQuA, and MATH. Our method outperforms Manual-CoT and Auto-CoT in seven out of ten categories across these datasets. Specifically, DUER demonstrates superior performance on GSM8K, achieving a score of 96.0, and surpasses the other methods in the MATH dataset, particularly in the precalculus and algebra categories. Furthermore, DUER leads to an average improvement of 0.9 points, showing a significant advantage over both Manual-CoT and Auto-CoT in terms of overall accuracy. The results indicate that our method not only shows significant advantages on GPT-3.5-turbo, but also outperforms others on GPT-4o and the MATH dataset. Specifically, DUER demonstrates superior performance across multiple sub-tasks in the MATH dataset, confirming the robustness and applicability of

Table 3. Accuracy (%) on commonsense reasoning benchmarks.

Method	CSQA	StrategyQA	Avg.	Δ
Zero-shot CoT	74.6	73.9	74.3	–
Zero-shot PS+	**79.3**	72.9	**76.1**	+1.8
Zero-shot DUP	76.8	73.9	75.3	+1.0
Few-shot Manual-CoT	78.0	73.3	75.7	+1.4
Few-shot Auto-CoT	77.6	68.4	72.5	−1.8
DUER (Ours)	77.9	**74.3**	**76.1**	+1.8

Table 4. Ablation study of DUER on GSM8K, SVAMP, and AQuA datasets using GPT-3.5-turbo. D-U denotes the deep understanding module and E-R denotes the extended reasoning module.

D-U	E-R	GSM8K	SVAMP	AQuA	Avg.	Δ
✘	✘	82.5	86.8	60.2	76.5	–
✔	✘	83.3 (+0.8)	87.5 (+0.7)	64.6 (+4.4)	78.5	+2.0
✘	✔	84.7 (+2.2)	88.2 (+1.4)	61.8 (+1.6)	78.2	+1.7
✔	✔	85.1 (+2.6)	87.5 (+0.7)	66.5 (+6.3)	79.7	+3.2

our approach in complex reasoning tasks. Moreover, it shows significant potential in solving complex problems that require deeper contextual understanding.

Commonsense Reasoning. Table 3 report the evaluation results on the commonsense reasoning using GPT-3.5-turbo for evaluation. Our approach does not show a clear advantage since the questions in the CSQA and StrategyQA datasets only contain 1-2 sentences and thus do not require complex understanding or reasoning. However, our method is generally on par with Zero-shot PS+ and slightly outperforms the other four baselines. Overall, DUER demonstrates strong performance across a range of benchmarks based on D-U and E-R modules, demonstrating its potential to address both straightforward and complex reasoning challenges.

Compatibility with Self-consistency (SC) [19]. SC samples N reasoning paths and selects the most consistent answer as the final result. We compare the performance of Auto-CoT, DUER, and DUER+ (using GPT-4o-mini as the information extractor) with and without self-consistency (SC) as the decoding strategy. We conduct experiments on the GSM8K and SVAMP benchmarks using GPT-3.5-Turbo, setting the temperature to 0.7 and the number of reasoning paths to 10. As shown in Fig. 3, DUER achieves an average improvement of 3% over Auto-CoT on both GSM8K and SVAMP. More notably, DUER+ based on GPT-4o-mini achieves impressive results of 91.1% on GSM8K and 93.2% on SVAMP, respectively, which are very close to the performance (96.0% and 94.9%) of GPT-4o reported in Table 2. This suggests that our method enhances the robustness, generalization, and stability of LLMs by utilizing the inherent diversity in their outputs during complex reasoning tasks.

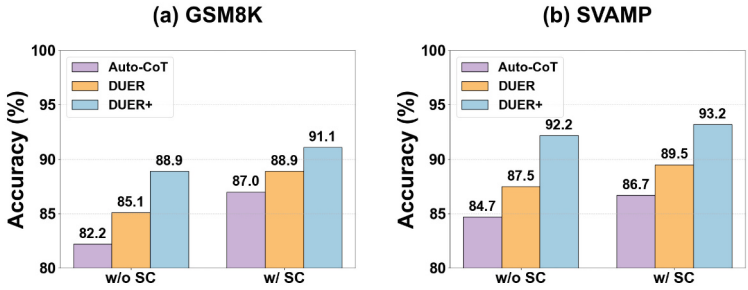

Fig. 3. Comparison results using GPT-3.5-turbo on the GSM8K and SVAMP benchmarks.

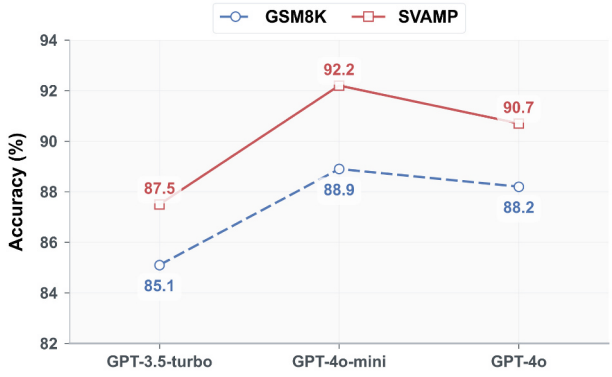

Fig. 4. Analysis of the accuracy of different LLMs as information extractors in D-U. It shows that GPT-4o-mini outperforms GPT-4o, which in turn surpasses GPT-3.5-turbo. The results demonstrate that GPT-4o-mini achieves the highest performance.

4.3 Ablation Study

We conducted an ablation study using GPT-3.5-turbo to validate the effectiveness of the proposed DUER modules, where the impact of D-U and E-R modules on arithmetic benchmarks, i.e., GSM8K, SVAMP, and AQuA.

The Impact of D-U. As shown in Table 4, the D-U module individually contributed to improvements of 0.8 and 0.7 on GSM8K and SVAMP, respectively. The average performance (79.7%) of DUER containing D-U and E-R modules is better than that (78.2%) of the E-R module alone, highlighting the enhancement of D-U for final performance. Note the multiple-choice AQuA benchmark improvement is as high as 4.4 points. That's because we integrate the question options into the prompt with the information extracted from LLMs, contributing to the deep understanding of the D-U module. The performance increment achieved by the D-U module on dataset AQuA (+4.4%) is higher than that on

Table 5. Effect of the length of the reasoning chain generated by GPT-3.5-turbo on the ablated counterparts of DUER.

D-U	E-R	GSM8K	SVAMP	AQuA	Avg.	Δ
✘	✘	4.9	4.3	3.7	6.5	–
✔	✘	4.8 (−0.1)	4.3 (0)	4.6 (+0.9)	6.9	+0.4
✘	✔	5.5 (+0.6)	5.6 (+1.3)	6.0 (+2.3)	8.6	+2.1
✔	✔	5.5 (+0.6)	5.6 (+1.3)	6.5 (+2.8)	8.8	+2.3

other datasets ($< 1\%$) because GSM8K and SVAMP allow for the generation of longer reasoning chains than AQuA, resulting in a more significant improvement on these two benchmarks.

To analyze the impact of question-related information extraction results on the reasoning performance of LLMs, we used GPT-3.5-turbo, GPT-4o-mini, and GPT-4o as information extractors for comparison on GSM8K and SVAMP, as shown in Fig. 4. GPT-4o-based and GPT-4o-mini-based DUER outperforms GPT-3.5-turbo, demonstrating that better extractors can improve the final accuracy. Interestingly, while GPT-4o achieves 90.7% accuracy on SVAMP, its smaller counterpart GPT-4o-mini attains superior performance (92.2%). This counterintuitive result suggests that divergent training strategies may lead to task-specific specialization. Existing studies have demonstrated that smaller models can outperform larger variants on specific tasks when employing distinct optimization objectives, particularly through customized architecture designs or task-specific training regimens. These findings emphasize that task-aligned optimization often outweighs pure model scaling in information extraction tasks, suggesting a need to re-examine the prevailing paradigm that equates model size with performance improvement.

The Impact of E-R. As shown in Table 5, we analyze the length of the reasoning chain generated by the GPT-3.5-turbo based on different modules of DUER to further explore whether the E-R module actually enhances the reasoning process of LLMs. Enhanced by the E-R module, GPT-3.5-turbo generates reasoning chains that average 2.1 sentences longer per question than those produced by the BASE method. Moreover, DUER containing D-U and E-R modules gains more sentences (+2.3) than the E-R module alone (+2.1), demonstrating the D-U module of DUER also contributes to generating longer reasoning chains toward the answer. Thus, we can confirm that the length of the reasoning chain significantly affects the performance of LLMs. Overall, DUER consistently outperforms its variant counterparts of partial components, demonstrating that all components contribute to the final performance. D-U and E-R complement each other from the semantic understanding and reasoning perspectives.

5 Conclusion

This paper proposes a novel few-shot prompting method, DUER, which enhances the complex reasoning abilities of LLMs from multiple enhancement perspectives, including semantic comprehension and reasoning chain improvement. We introduce a D-U module to enhance semantic understanding and an E-R module to enhance the reasoning chain. By combining these two complementary modules, significant reductions are achieved in semantic misunderstandings, calculation errors, and step-missing errors in reasoning. Experimental results demonstrate that the DUER, DUER+, and DUER+ decoding with SC models, all based on the GPT-3.5-Turbo architecture, achieve accuracy rates of 85.1%, 88.9%, and 91.1% on the GSM8K benchmark, and 87.5%, 92.2%, and 93.2% on the SVAMP benchmark, respectively. Thus, the proposed method establishes new SOTA results on these two popular benchmarks, demonstrating its applicability and effectiveness for complex reasoning tasks.

Acknowledgments. We truly thank the reviewers and editors for their great efforts. This work is partially supported by the research project funded by the Fundamental Research Funds for the Central Universities (21624325, 21624338), the Key Laboratory of Smart Education of Guangdong Higher Education Institute, Jinan University (2022LSYS003), and the Guangdong Basic and Applied Basic Research Foundation (2024A1515140144).

References

1. Achiam, J., et al.: GPT-4 technical report. ArXiv preprint (2023)
2. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
3. Cobbe, K., et al.: Training verifiers to solve math word problems. ArXiv preprint (2021)
4. Fu, Y., Peng, H., Sabharwal, A., Clark, P., Khot, T.: Complexity-based prompting for multi-step reasoning. In: The Eleventh International Conference on Learning Representations (2023)
5. Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., Berant, J.: Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. Trans. Assoc. Comput. Linguist. **9**, 346–361 (2021). https://doi.org/10.1162/tacl_a_00370
6. Hendrycks, D., et al.: Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874 (2021)
7. Hosseini, M.J., Hajishirzi, H., Etzioni, O., Kushman, N.: Learning to solve arithmetic word problems with verb categorization. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 523–533. Association for Computational Linguistics, Doha, Qatar (2014). https://doi.org/10.3115/v1/D14-1058
8. Hurst, A., et al.: GPT-4o system card. arXiv preprint arXiv:2410.21276 (2024)

9. Jin, M., et al.: The impact of reasoning step length on large language models. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics, pp. 1830–1842. Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.findings-acl.108
10. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In: Advances in Neural Information Processing Systems, vol. 35, pp. 22199–22213 (2022)
11. Ling, W., Yogatama, D., Dyer, C., Blunsom, P.: Program induction by rationale generation: learning to solve and explain algebraic word problems. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 158–167. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1015
12. Luo, H., et al.: Wizardmath: empowering mathematical reasoning for large language models via reinforced evol-instruct. arXiv preprint arXiv:2308.09583 (2023)
13. Miao, S.y., Liang, C.C., Su, K.Y.: A diverse corpus for evaluating and developing English math word problem solvers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 975–984 (2020)
14. Patel, A., Bhattamishra, S., Goyal, N.: Are NLP models really able to solve simple math word problems? In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2080–2094. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-main.168
15. Roy, S., Roth, D.: Solving general arithmetic word problems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processin, pp. 1743–1752. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/D15-1202
16. Talmor, A., Herzig, J., Lourie, N., Berant, J.: CommonsenseQA: a question answering challenge targeting commonsense knowledge. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4149–4158. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1421
17. VanLehn, K.: Cognitive skill acquisition. Ann. Rev. Psychol. **47**, 513–39 (1996). https://api.semanticscholar.org/CorpusID:9247247
18. Wang, L., et al.: Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2609–2634. Association for Computational Linguistics, Toronto, Canada (2023). https://doi.org/10.18653/v1/2023.acl-long.147
19. Wang, X., et al.: Self-consistency improves chain of thought reasoning in language models. In: The Eleventh International Conference on Learning Representations (2023)
20. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837 (2022)
21. Xu, X., et al.: Re-reading improves reasoning in large language models. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 15549–15575 (2024)
22. Xue, T., Wang, Z., Wang, Z., Han, C., Yu, P., Ji, H.: RCOT: detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. arXiv preprint arXiv:2305.11499 (2023)

23. Zhang, Z., Zhang, A., Li, M., Smola, A.: Automatic chain of thought prompting in large language models. In: The Eleventh International Conference on Learning Representations (2023)
24. Zhong, Q., Wang, K., Xu, Z., Liu, J., Ding, L., Du, B., Tao, D.: Achieving >97% on GSM8K: deeply understanding the problems makes LLMs perfect reasoners. arXiv preprint arXiv:2404.14963 (2024)
25. Zhou, D., et al.: Least-to-most prompting enables complex reasoning in large language models. In: The Eleventh International Conference on Learning Representations (2023)
26. Zhou, Y., et al.: Large language models are human-level prompt engineers. In: The Eleventh International Conference on Learning Representations (2022)