Divide and Conquer for Large Language Models Reasoning

Zijie Meng * *zijie.22@intl.zju.edu.cn*
Zhejiang University-University of Illinois at Urbana Champaign Institute
Zhejiang University

Yan Zhang * *yanzhang.jlu@gmail.com*
Department of Electrical and Computer Engineering
National University of Singapore

Zhaopeng Feng *zhaopeng.23@intl.zju.edu.cn*
Zhejiang University-University of Illinois at Urbana Champaign Institute
Zhejiang University

Yang Feng *fengyang@angelalign.com*
Angelalign Research Institute
Angelalign Technology Inc.

Gaoang Wang *gaoangwang@intl.zju.edu.cn*
Zhejiang University-University of Illinois at Urbana Champaign Institute
Zhejiang University

Joey Tianyi Zhou *zhouty@cfar.a-star.edu.sg*
Institute of High Performance Computing (IHPC), Centre for Frontier AI Research (CFAR)
*Agency for Science, Technology and Research (A*STAR)*

Jian Wu *wujian2000@zju.edu.cn*
Second Affiliated Hospital School of Medicine, School of Public Health
Zhejiang University

Zuozhu Liu † *zuozhuliu@intl.zju.edu.cn*
Zhejiang University-University of Illinois at Urbana Champaign Institute
Zhejiang University

Abstract

Large language models (LLMs) have shown impressive performance in various reasoning benchmarks with the emergence of Chain-of-Thought (CoT) and its derivative methods, particularly in tasks involving multi-choice questions (MCQs). However, current works all process data uniformly without considering the problem-solving difficulty, which means an excessive focus on simple questions while insufficient to intricate ones. To address this challenge, we inspired by humans using heuristic strategies to categorize tasks and handle them individually, propose to apply the Divide and Conquer to LLMs reasoning. First, we divide questions into different subsets based on the statistical confidence score ($\mathcal{CS}$), then fix nearly resolved sets and conquer demanding nuanced process ones with elaborately designed methods, including Prior Knowledge based Reasoning (PKR) and Filter Choices based Reasoning (FCR), as well as their integration variants. Our experiments demonstrate that this proposed strategy significantly boosts the models' reasoning abilities across nine datasets involving arithmetic, commonsense, and logic tasks. For instance, compared to baseline, we make a striking improvement on low confidence subsets of 8.72% for AQuA, 15.07% for ARC Challenge and 7.71% for RiddleSense. In addition, through extensive

*Indicates equal contribution.
†Corresponding author.

analysis on length of rationale and number of options, we verify that longer reasoning paths in PKR could prevent models from referring infer-harmful shortcuts, and also find that removing irrelevant choices in FCR would substantially avoid models' confusion. The code is at `https://github.com/AiMijie/Divide-and-Conquer`.

1 Introduction

Large language models (LLMs) (e.g. GPT3 (Brown et al., 2020), GPT4 (OpenAI, 2023), Palm (Chowdhery et al., 2022), Palm2 (Anil et al., 2023), Lamda (Touvron et al., 2023a), Llama (Touvron et al., 2023a), Llama2 (Touvron et al., 2023b)) have exhibit outstanding performance on tasks involving multi-choice questions (MCQs) by generating step by step rationales before producing answers, which is elicited from Chain-of-Thoughts (CoT) (Wei et al., 2022) and its derivative methods such as Zero-Shot-CoT (Kojima et al., 2022), self-consistency (Wang et al., 2022). However, current works all treat the dataset as a whole regardless of the problem-solving difficulty, which means that simple questions do not necessitate complex, repetitive, or costly procedures, whereas intricate ones cannot be adequately addressed with mere simplistic methods. It is also natural that humans utilize heuristic strategies to categorize tasks into sub-tasks based on the range or type of elements pending solution, and then address each individually, which not only effectively resolves complex issues, but also significantly enhances problem-solving efficiency and method generalization (Heideman et al., 1984; Knuth, 1998). Consequently, we apply this strategy of data partitioning followed by differential process—Divide and Conquer, which is widely deployed across numerous disciplines and scenarios (Bentley & Shamos, 1976; Bentley, 1980; Smith, 1985; Eisenstein, 2006; Mallouk, 2013)—to LLM reasoning aiming to strike a better balance between cost and accuracy. In this context, we need to address two paramount challenges: (1) What criteria should be used to divide the dataset? (2) How should the subsets be processed?

For the first one, we need to explore a method to effectively divide questions based on solving difficulty. In human common perception, answers with high uncertainty prone to be wrong, whereas results recovered from diverse reasoning path are more likely to ensure correctness. So we tentatively probe self-consistency (Wang et al., 2022), which samples different reasoning paths to generate multiple candidates following majority voting to derive final answer, on AQuA (Ling et al., 2017), CMSQA (Talmor et al., 2018), and RiddleSense (Lin et al., 2021). As shown in Fig. 4, it is straightforward to divide the questions into two subsets based on their confidence, where the high confidence one rapidly approaches to the upper bound of accuracy, and the other continuously increases with the algorithm proceeds. This suggests that we can employ self-consistency to compute a statistical confidence score for each problem and divide them based on the respective score ranges.

Move to the second issue, we inspired from the managerial buckets effect (Peter et al., 1969), fix questions with high confidence, which are sufficiently simple for the model eliminating repeated inference, and conquer with more carefully reasoning methods for the low confidence subsets, which demand nuanced process offering greater room for optimization. On one hand, current works always have various limitations, where CoT (Wei et al., 2022) employs meticulously crafted prompts which require substantial manual interventions, Zero-Shot-CoT (Kojima et al., 2022) focus on lower costs and cannot utilize the content generated in divide stage, PHP (Zheng et al., 2023a) merely use the answer as progressive hints without considering rationales. Consequently, we propose Prior Knowledge based Reasoning (PKR), which leverage rationales derived previous results as a reference to guide inference without any human efforts. Meanwhile, we conducted extensive analysis on rationale length and verify that longer ones could prevent models from referring shortcuts generated by the influence of training data, which are harmful for knowledge extraction. On the other hand, Shi et al. (2023) investigated the model's sensitivity to irrelevant information within the questions, but there exists uncertainty regarding distraction by irrelevant answers. To delve into this phenomenon, we conducted preliminary studies on four MCQs datasets: AQuA (Ling et al., 2017), CMSQA (Talmor et al., 2018), OpenBookQA (Mihaylov et al., 2018), and RiddleSense (Lin et al., 2021), discovering a decrease in problem-solving accuracy as the number of choices increased. Following this, we extracted subsets of hardly solved

[1]Inpired from Complex CoT (Fu et al., 2022), we choose the longer rationale, which could bring more helpful knowledge to extract for inference. And we conduct experiments to analyse the effect of sampling strategy as shown in Table 5.

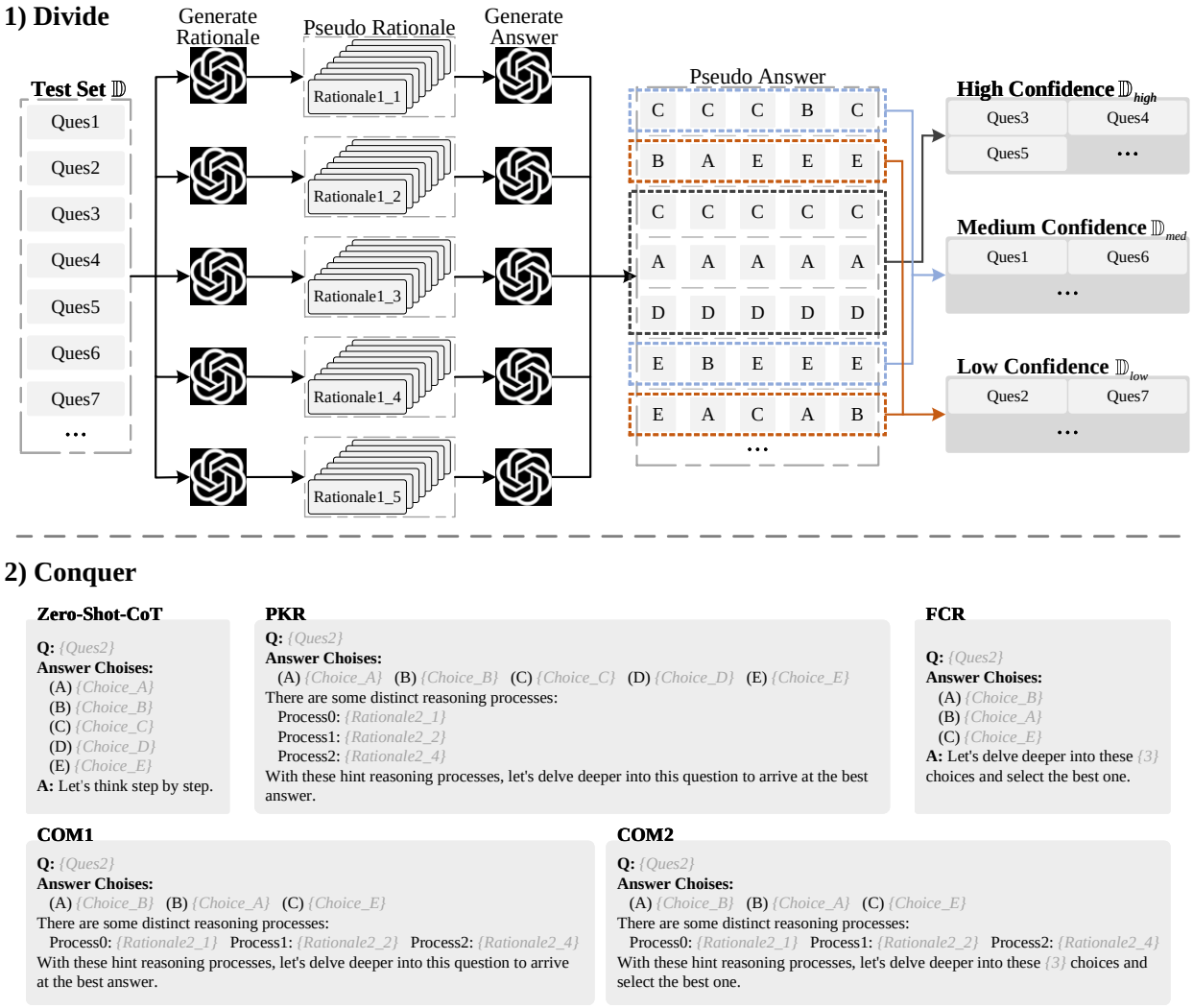


Figure 1: Illustration of Divide and Conquer. (1) Divide. Using Zero-Shot-CoT, we first conduct t (e.g. $t = 5$) times inference. Then, the dataset $\mathbb{D}$ is divided based on the confidence Score $\mathcal{CS}$ of the questions, where subsets with $\mathcal{CS} > \mu$ (e.g. $\mu = 0.8$) are categorized as $\mathbb{D}_{high}$, those with $\mathcal{CS} \leq \nu$ (e.g. $\nu = 0.6$) as $\mathbb{D}_{low}$, and the rest as $\mathbb{D}_{med}$. (2) Conquer. We propose two distinct processing PKR and FCR, furthermore we introduce two combination variants, COM1 and COM2, through different merging strategies. "Ques" in Divide area includes question text and choices list, while denotes only question text in Conquer area. "Choice_x" represents the x-th option in original problem. "Rationale$i$_j" denotes the rationale generated by j-th LLM query for i-th Ques. "Rationale2_4" is the longest rationale [1]in the last three rationales which indicate the same answer.

questions from each test set, removed some irrelevant options, and then re-queried the LLM, resulting in a universal accuracy improvement of over 20%, especially a staggering 75.52% increase on CMSQA, as shown in Table 6. Therefore, we introduce Filter Choices based Reasoning (FCR), which excludes abundant options by using the results from the divide stage. In addition, built on them, we develop two variants, COM1 and COM2, which not only filter irrelevant choices but also introduce prior rationales, while the former inquiry model referring prompt from PKR and the latter from FCR.

Empirically, we evaluate the strategy of Divide and Conquer on nine datasets across arithmetic, commonsense, and logic categories. After divide stage, the high confidence subsets achieved an impressive weighted average accuracy of 92.05%, the medium confidence subsets scored 68.00%, and the low confidence subsets

only reached 43.09%, which indicates the rationality of our divide strategy. At the same time, different methods all achieved improvements in the last subset. Specifically, COM1 obtained a 6.22% improvement in weighted average, and with self-consistency, FCR gained an enhancement of 2.81%. These results demonstrate that this proposed strategy can consistently enhance the reasoning reliability and effectiveness of LLMs. Furthermore, we explore the resource and generalizability of our strategy. Although it is intuitive to divide dataset with the confidence, the method based on self-consistency incurs significant overhead. Therefore, we propose a prompt "Let's substitute the answer back into the question to check it is 'true' or 'false':" enabling the model to directly assess confidence with the answers generated from a single inference. We also extend the Divide and Conquer to the cloze-style dataset GSM8K (Cobbe et al., 2021), using the answers generated in the divide phase as options during the conquer phase, which lead to FCR achieving an 8.31% improvement over Zero-Shot-CoT on the low confidence subset.

In summary, our work has four major contributions as follows: (1) To the best of our knowledge, we pioneeringly employ the Divide and Conquer to LLM reasoning, providing the community with a novel perspective. (2) We first divide dataset based on confidence score, then fix nearly resolved sets and conquer the rest ones with four approaches (PKR, FCR, COM1, COM2), achieving a more favorable trade-off between cost and accuracy. (3) We evaluate this strategy across nine datasets within three distinct tasks, consistently achieving significant improvements. (4) Through extensive analysis, we demonstrate that confidence score is positive correlate with accuracy, longer rationale could offer more helpful knowledge for reference, and irrelevant information in choices list would distract model.

2 Methodology

The overall framework of the Divide and Conquer is illustrated in Fig. 1, which represents our novel perspective for enhancing LLMs' reasoning capabilities in downstream tasks. Given a test set of length n represented as $\mathbb{D} = \{(Q_1, C_1), (Q_2, C_2), ..., (Q_n, C_n)\}$, where Q_i denotes the i-th question context and C_i is the corresponding choices list. In addition, we use R_i and A_i to denote its rationale and answer generated by LLMs, respectively. First, we would introduce two strong baselines utilized in our work: Zero-Shot-CoT (Kojima et al., 2022) and self-consistency (Wang et al., 2022). Next, we will describe the details for our strategy, which divide $\mathbb{D}$ into three confidence subsets: $\mathbb{D}_{high}$, $\mathbb{D}_{med}$ and $\mathbb{D}_{low}$, and then retain $\mathbb{D}_{high}$, while conquer $\mathbb{D}_{med}$ and $\mathbb{D}_{low}$.

2.1 Background

2.1.1 Zero-Shot-CoT

For some complex MCQs which need multi-step reasoning, CoT (Wei et al., 2022) extends the conventional problem-solving representation from the triplet (Q, C, A) to a quadruple (Q, C, R, A), meaning that the model will generate a chain of thoughts before producing the final answer via referring demonstrations. However, CoT requires substantial human labor to annotate task-specific examplars, and as the scale of model increases, zero-shot performance gradually approaches or even surpasses few-shot (Hu et al., 2023; Zhong et al., 2023). See Table 10 in Appendix A for our verification experiment. So we concentrate on Zero-Shot-CoT (Kojima et al., 2022), which is proposed to trigger the generation of intermediate reasoning through prompt "Let's think step by step." without any manual effort.

2.1.2 Self-consistency

Self-consistency (Wang et al., 2022) first samples multiple reasoning paths instead of greedy one to generate various candidate results and then employs a majority voting mechanism to select the most consistent answer. Specifically, if we repeat inference 10 times, there will produce 10 quadruples $\{(Q, C, R_j, A_j)|j \in \{1, ..., 10\}\}$, and the final answer can be determined by $A = \arg\max_{A_j} \sum_{j \in \{1,...,10\}} \mathbf{1}_{A_j = A}$. As a plug-and-play approach, it can significantly enhance performance of CoT prompting and promote fairness by thoroughly exploring various possible reasoning paths.

2.2 Divide

With test item (Q_i, C_i), we initially query the LLM t times to obtain t rationales $R_i = \{r_{i,1}, r_{i,2}, ..., r_{i,t}\}$ and t corresponding answers $A_i = \{a_{i,1}, a_{i,2}, ..., a_{i,t}\}$ based on Zero-Shot-CoT (Kojima et al., 2022). We use δ to denote LLM and define the confidence score $\mathcal{CS}$ of each problem as:

$$\mathcal{CS}_{(Q_i,C_i)} = \max_j p(a_{i,j}|\delta(Q_i, C_i)), i \in \{1, ..., n\}, j \in \{1, ..., t\}, \tag{1}$$

where $p(a_{i,j}|\delta(Q_i, C_i))$ is the frequency of $a_{i,j}$ in all t predicted answers. And we define it as:

$$p(a_{i,j}|\delta(Q_i, C_i)) = \frac{\sum_{k\in\{1,...,t\}} \mathbf{1}_{a_{i,k}=a_{i,j}}}{t}, \tag{2}$$

Then, based on the $\mathcal{CS}$ of each problem, we can divide the dataset with the following rule:

$$(Q_i, C_i) \in \begin{cases} \mathbb{D}_{high}, & \text{if } \mathcal{CS}_{(Q_i,C_i)} > \mu, \\ \mathbb{D}_{med}, & \text{if } \nu < \mathcal{CS}_{(Q_i,C_i)} \leq \mu, \\ \mathbb{D}_{low}, & \text{if } \mathcal{CS}_{(Q_i,C_i)} \leq \nu, \end{cases} \tag{3}$$

where $\mathbb{D}_{high}$, $\mathbb{D}_{med}$ and $\mathbb{D}_{low}$ represent the high, medium and low confidence subsets respectively. μ and ν are thresholds for dividing, which is specified in Sec. 3.2. Next, we would stop the process of the high confidence subset to conserve resources, while delving deeper into the latter two subsets.

2.3 Conquer

In this section, we first introduce two approaches that effectively utilize the outputs from the previous t inferences. Subsequently, we explore different integration strategies and derive two variants.

2.3.1 Prior knowledge based reasoning (PKR)

As illustrated in PKR of Fig. 1, we utilize the rationale R_i generated in divide stage as prior knowledge to stimulate the potential capabilities of the model. Therefore, the input to the model is transformed from (Q_i, C_i) to (Q_i, C_i, R_i). However, considering there may be multiple rationales corresponding to the same answer in previous inferences, we sample the longest one among each related reasoning paths, which inspired from Complex CoT (Fu et al., 2022). Meanwhile, we conduct experiments on various rationale sampling strategies in Table 5, where selecting the longest one could bring the best performance. Consequently, we perform clustering on R_i based on their successive answers, and subsequently select the longest one from each cluster to form a new subset $R'_i \subseteq R_i$. And then we construct the input (Q_i, C_i, R'_i) for the LLM.

2.3.2 Filter choices based reasoning (FCR)

As demonstrated in FCR of Fig. 1, we make use of the results obtained during divide phase as alternative options for the subsequent queries of the LLM. Specifically, we define $C'_i = uniq(A_i)$, where the *uniq* operation signifies the deduplication of $A_i = \{a_{i,1}, ..., a_{i,t}\}$. Then, (Q_i, C'_i) is constructed in the new prompt to the model. It is noteworthy that our method does not merely delete options, rather it involves a synchronous modification of the option symbols (i.e. 'A', 'B', 'C', etc.) based on the number of remaining choices.

2.3.3 Combination (COM1 and COM2)

For the two aforementioned reasoning methods, we propose two integration variants. On one hand, we transforms the choices list from C_i to C'_i, while on the other, we incorporate prior knowledge R'_i into the prompt. Additionally, at the end of the input, one variant (COM1) includes a phrase originating from PKR: "let's delve deeper into this question to arrive at the best answer", while the other (COM2) includes that derived from FCR: "Let's delve deeper into these *{length of options list}* choices and select the best one".

Table 1: The information statistic of datasets. For CMSQA, RiddleSense, LogiDeduction and Reclor, we select their validation sets as there are no publicly available test sets or labels. GSM8K is used to explore the generalization of Divide and Conquer in Sec. 3.4.5, which is a cloze-style dataset without choices list. #Test indicates the number of data items we used in experiments. #ChoicesNum represents the length of choices list for each question in corresponding dataset. Particularly, for LogiDeduction, there are 60 questions with an choices list length of 3, 100 questions with a length of 5, and 140 questions with 7. #DivideBase denotes inference times in divide stage, which generally is equal to the #ChoicesNum, considering the worst-case scenario where all choices could appear in the prior results. For LogiDeduction, given that 20% questions have 3 choices, we make a compromise and choose #DivideBase=4.

Dataset	Task Type	Eval. Split	#Test	#ChoicesNum	#DivideBase
AQuA (AQ.)	Arithmetic	Test	254	5	5
Algebra (Alg.)	Arithmetic	Test	100	4	4
Mathematics (Math.)	Arithmetic	Test	270	4	4
CMSQA (CMS.)	Commonsense	Validation	1221	5	5
OpenBookQA (OB.)	Commonsense	Test	500	4	4
ARC Challenge (ARC.)	Commonsense	Test	1165	4	4
RiddleSense (Rid.)	Logic	Validation	1021	5	5
LogiDeduction (Logi.)	Logic	Validation	300	3, 5 or 7	4
Reclor (Rec.)	Logic	Validation	500	4	4
GSM8K (GSM.)	Arithmetic	Test	1319	-	-
SVAMP (SVA.)	Arithmetic	Test	300	-	-

3 Experiments

3.1 Datasets and evaluation metrics

To evaluate the effectiveness of our approach and to investigate the reasons behind this improvement, we conducted experiments on three categories of tasks:

- Arithmetic. AQuA (Ling et al., 2017) and Abstract Algebra (Algebra), High School Mathematics (Mathematics) from the MMLU dataset (Hendrycks et al., 2020). These three benchmarks all consist of multiple choice math problems that require multi-step reasoning.
- Commonsense. CMSQA (Talmor et al., 2018), OpenBookQA (Mihaylov et al., 2018) and ARC Challenge (Clark et al., 2018). CMSQA incorporate diversity world knowledge into questions. OpenBookQA contains questions require additional common knowledge and rich text comprehension. ARC Challenge includes only grade school level science MCQs that cannot be solved through direct retrieval or word co-occurrence algorithm.
- Logic. RiddleSense (Lin et al., 2021), Logical Deduction (LogiDeduction) from BIG-bench dataset (Srivastava et al., 2022) and Reclor (Yu et al., 2020). RiddleSense is designed to evaluate models' performance on riddle-style questions. LogiDeduction is a benchmark based on deductive reasoning, which includes questions with choices list lengths of 3, 5 or 7. Reclor is extracted from logical reasoning questions of standardized graduate admission examinations.

The statistical details of these nine datasets can be found in Table 1. Additionally, we employed exact match (EM) accuracy to evaluate the performance of various methods, which aligns with previous works (Wei et al., 2022; Kojima et al., 2022).

3.2 Implementation details

We primarily employed currently garnering significant attention and publicly accessible GPT-3.5-Turbo-0613 from OpenAI API [2]. During the dataset divide phase, we set the temperature=0.7, and then conducted

[2] `https://platform.openai.com`

inference according to the #DivideBase in Table 1. Based on these prior results, we divided the dataset into three subsets: $\mathbb{D}_{high}$, $\mathbb{D}_{med}$, and $\mathbb{D}_{low}$ with $\mu = 0.8$ and $\nu = 0.6$, which ensures each subset is meaningful under the #DivideBase setting. Specifically, we set #DivideBase to 4 or 5, equal to the corresponding #ChoicesNum, mainly to ensure that each option could appears in the prior results in the worst-case. And we demonstrated that only minimal inferences are enough for the accuracy of different subsets to enter a stable oscillation zone, as depicted in Fig. 5. The values for μ and ν are chosen from the quintile points with referring #DivideBase. And in over half of datasets, subsets with $\mathcal{CS} \leq 0.4$ are meaningless, as shown in Table 4. The number of data entries in each subset after the division is shown in Table 2. During the conquer phase, we fixed the temperature=0 for reproducibility, while in the case of utilizing self-consistency, we set the temperature=0.7 consistent with previous works (Wang et al., 2022) and inference times is equal to #DivideBase in Table 1. Experiments are conducted on the full dataset by default unless in Fig. 4, Fig. 7, Fig. 8, Table 6 and Table 7, where we randomly sampled 500 items for each dataset except 254 for AQuA. In addition, the final results are all obtained by averaging three random trials.

3.3 Main results

Results are presented in Table 2, and we observed the following findings.

High confidence subsets $\mathbb{D}_{high}$ exhibit nearly resolved accuracy and require no further process. The $\mathcal{CS}$ exceeds 0.8 for each question in $\mathbb{D}_{high}$, indicating that LLM is sufficiently confident for their generated results. As shown in Table 2, the results in the divide phase have already achieved the weighted average accuracy of 92.05%, and more than half of the datasets surpassed 90%. In addition, we conducted an additional test on this subset based on Zero-Shot-CoT, yielding an average result of 89.28%. Consequently, we believe that the majority of the questions in $\mathbb{D}_{high}$ are simple items and do not require any further processing.

Medium confidence subsets $\mathbb{D}_{med}$ demonstrate moderate accuracy and obtain limited benefits from additional intervention. In $\mathbb{D}_{med}$, the confidence score for each question is $\mathcal{SC} \in (0.6, 0.8]$, which indicates that although the model can generate diverse answers, it still concentrate on a particular option. This introduces a significant challenge to enhance LLM's reasoning performance by correcting its previously generated mistakes. According to Table 2, while FCR achieved an average improvement of 6.99% without self-consistency, its performance was only 0.59% higher than Zero-Shot-CoT using self-consistency. Meanwhile, PKR's performance on this subset significantly lagged behind FCR, illustrating that the model has an inherent tendency when referring the prior knowledge and it can not effectively correct errors in previously generated rationale. Therefore, merely introducing prior knowledge is insufficient for $\mathbb{D}_{med}$, and we need more refined methods or more powerful models to address this challenge.

Low confidence subsets $\mathbb{D}_{low}$ display poor accuracy and warrant granular attention. Based on our divide criteria, problems in $\mathbb{D}_{low}$ typically have more than two answers in the previous t inference rounds, indicating significant inconsistency in the model's solutions for these questions. In fact, the prior average accuracy for this subset is only 43.09%, substantially lower than the other two confidence subsets. In light of this, we tested all five methods—Zero-Shot-CoT, PKR, FCR, COM1, and COM2—on $\mathbb{D}_{low}$, and introduced the self-consistency to further evaluate our approach, with detailed results presented in Table 2. Without self-consistency, our methods achieved significant improvements across nine datasets spanning three categories. For instance, on the AQuA of arithmetic, FCR outperformed Zero-Shot-CoT by 8.72%; on the ARC Challenge of commonsense, PKR improved by 15.07%; and on the RiddleSense of logic, COM2 enhanced by 7.71%. Additionally, COM2 increased the weighted average accuracy across all datasets by 6.22%. And FCR remained 2.81% higher than Zero-Shot-CoT even with self-consistency.

3.4 Analysis

3.4.1 Comparison agains different LLMs, reasoning methods and prompts

Different LLMs. In order to evaluate the generalizability of our method, we conducted a comparative analysis between Zero-Shot-CoT (Kojima et al., 2022) and relatively superior FCR using the highly popular LLMs, Palm2 (Anil et al., 2023) and Vicuna v1.5 (Zheng et al., 2023d). Specifically, we employed "text-

Table 2: Comparison of problem-solving accuracy (%) among different methods on the nine datasets and their respective subsets. #NUM indicates the number of data items in different subsets after division. Prior denotes the accuracy of results from divide stage. ZTCoT signifies Zero-Shot-CoT (Kojima et al., 2022). From left to right, the nine datasets are AQuA, Algebra, Mathematics, CMSQA, OpenBookQA, ARC Challenge, RiddleSense, LogiDeduction and Reclor. "Weighted Average" represents the weighted average based on the #NUM of different dataset subsets. Results in **bold** are the best performance in each subset.

Subset	Setting	Arithmetic			Commonsense			Logic			Weighted Average
		AQ.	Alg.	Math.	CMS.	OB.	ARC.	Rid.	Logi.	Rec.	
$\mathbb{D}_{high}$	#NUM	81	35	73	574	324	866	393	24	223	288.11
	Prior	96.30	57.14	90.41	93.21	96.30	96.30	89.82	79.17	76.23	92.05
	ZTCoT	92.59	50.48	87.67	89.84	92.39	94.38	86.77	72.22	75.19	89.28
$\mathbb{D}_{med}$	#NUM	43	32	87	274	102	195	222	87	155	133.00
	Prior	67.44	43.75	58.62	74.82	67.65	76.92	72.97	57.47	54.19	68.00
	ZTCoT	55.81	34.38	50.57	63.75	64.05	64.10	56.31	44.06	46.88	56.81
	PKR	50.39	**38.54**	53.26	59.85	63.73	67.69	55.86	52.87	48.17	57.31
	FCR	58.14	**38.54**	**57.47**	**71.65**	**70.92**	**72.82**	**64.86**	49.81	**50.54**	**63.80**
	COM1	58.14	34.38	42.53	64.11	65.36	69.57	59.46	**55.94**	48.82	59.09
	COM2	**59.69**	37.50	50.19	66.30	65.36	68.55	57.66	53.26	48.39	59.54
	with self-consistency										
	ZTCoT	59.69	**39.58**	60.15	69.34	68.95	**73.68**	66.67	**58.62**	51.61	64.63
	PKR	59.69	38.54	51.72	63.75	66.99	70.77	59.31	54.02	50.97	60.29
	FCR	**62.79**	36.46	**62.45**	**70.80**	**71.24**	73.50	**66.97**	54.79	**52.47**	**65.22**
$\mathbb{D}_{low}$	#NUM	130	33	110	373	74	104	406	189	122	171.22
	Prior	47.69	30.30	40.91	47.45	44.59	48.08	43.84	33.86	36.89	43.09
	ZTCoT	44.36	28.28	40.61	44.59	51.35	39.42	37.77	29.10	41.80	40.00
	PKR	47.18	39.39	**44.24**	44.50	51.35	**54.49**	44.58	36.68	40.44	44.34
	FCR	**53.08**	37.37	40.91	**52.55**	**56.76**	50.96	43.02	36.68	**46.17**	**46.57**
	COM1	48.97	35.35	38.48	46.83	53.15	52.56	44.01	**42.68**	44.54	45.43
	COM2	45.13	**40.40**	43.94	50.22	52.70	52.56	**45.48**	39.86	41.80	46.22
	with self-consistency										
	ZTCoT	48.72	38.38	40.91	50.22	51.80	51.60	45.32	34.57	40.16	45.34
	PKR	51.28	**41.41**	40.00	46.56	48.20	**54.81**	44.17	40.92	43.72	45.47
	FCR	48.46	39.39	**50.00**	**54.51**	**57.66**	54.49	44.50	38.62	44.81	**48.15**
	COM1	**55.13**	40.40	43.94	49.06	57.21	52.56	**46.31**	**41.80**	**47.27**	47.89
	COM2	**55.13**	40.40	43.33	52.19	53.15	52.56	45.40	40.92	41.53	47.61

Table 3: Problem-solving accuracy (%) for $\mathbb{D}_{low}$ of AQuA and CMSQA based on different methods.

Method	ZTCoT	ManualCoT	Active-Prompt	PHP	Role-Play	PKR	FCR
AQuA	44.36	37.95	42.82	46.41	50.00	47.18	**53.08**
CMSQA	44.59	53.80	**56.75**	-	47.99	44.50	52.55
Average	44.48	45.88	49.79	46.41	49.00	45.84	**52.82**

bison-001" available on the official platform [3], and "vicuna-13b-v1.5" finetuning on Llama2 (Touvron et al., 2023b), accessible for free download on the Hugging Face [4]. Additionally, our experiments were excuted on four low confidence subsets with spanning three tasks, and asked models by "Let's think step by step." because of their relatively poor reasoning capacity, as shown in Fig. 2. Even though Bison is not the most

[3] https://developers.generativeai.google

[4] https://huggingface.co/lmsys/vicuna-13b-v1.5

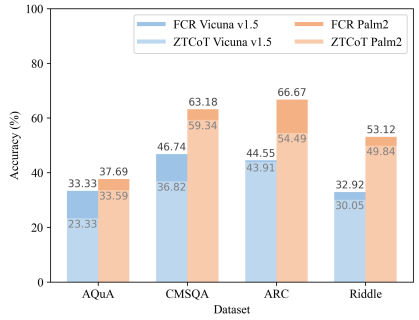


Figure 2: Performance on low confidence subsets with different LLMs.

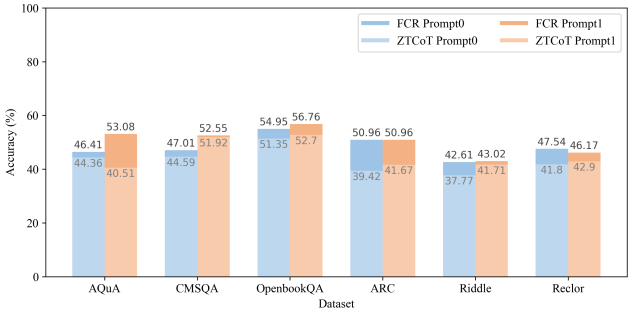


Figure 3: Perfomance on low confidence subsets with different prompts.

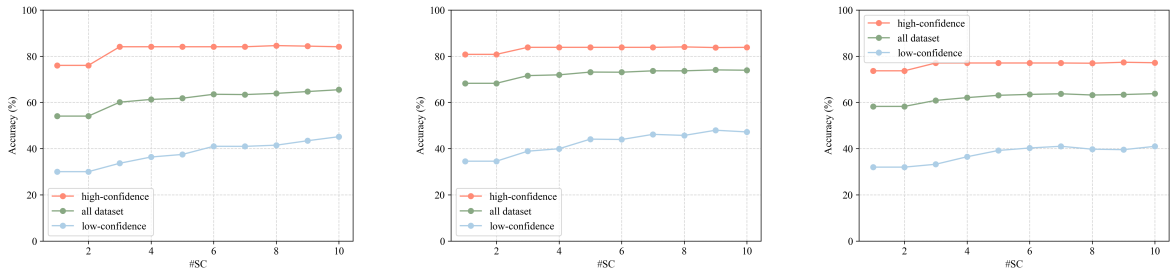
powerful model in Palm2 series, it still broadly surpasses Vicuna v1.5. Simultaneously, FCR introduced in the Conquer phase is uniformly superior to ZTCoT, which illustrate our methods can be applied in various LLMs.

Different reasoning methods. In Sec. 4.2, we enumerate various methods in LLM reasoning and analyse their interconnections. Therefore, we selected some representative few-shot works: ManualCoT (i.e. CoT) (Wei et al., 2022), Active-Prompt (Diao et al., 2023), and PHP (Zheng et al., 2023a), along with some zero-shot methods: Zero-Shot-CoT (Kojima et al., 2022) and Role-Play Prompting (Kong et al., 2023), for comparison with our proposed zero-shot based PKR and FCR, as shown in Table 3. Considering the variations of datasets tested in different methods, we chose $\mathbb{D}_{low}$ of two widely used AQuA and CMSQA for our experiments. Notable, we solely compared the performance of PHP on AQuA since it only reported results on some math tasks. Although PKR's performance significantly trails behind FCR, it still surpasses Zero-Shot-CoT and is comparable to the few-shot based ManualCoT, which suggests that we can achieve commendable grade without any manual intervention. Moreover, the accuracy of PKR on AQuA is 1.03% higher than PHP, indicating that using rationale as prior knowledge is more effective than merely taking the answer as hint. Furthermore, FCR achieves the best average accuracy surpassing the sub-optimal Role-Play Prompting by 3.3%, which highlights the strong efficacy of our method.

Different prompts. Unlike PKR, where we set prompts for refering introduced prior knowledge, the most distinctive feature of FCR is more concise choices list compared to original problems. Therefore, we compared the performance between FCR and Zero-Shot-CoT (Kojima et al., 2022) by applying different prompts, as displayed in Fig. 3. Specifically, the prompt0 denotes "Let's think step by step" from Zero-Shot-CoT and the prompt1 represents "Let's delve deeper into these *{length of options list}* choices and select the best one" from native FCR. We utilized these two prompts separately to Zero-Shot-CoT and FCR, and tested on six datasets covering all three tasks. From horizontal perspective, different prompts did not significantly effect accuracy. However, vertically, FCR consistently led to an improvement, which indicates FCR's insensitiveness to the prompt, further demonstrating that the performance enhancement stems primarily from our core idea: reducing irrelevant choices, rather than prompt engineering.

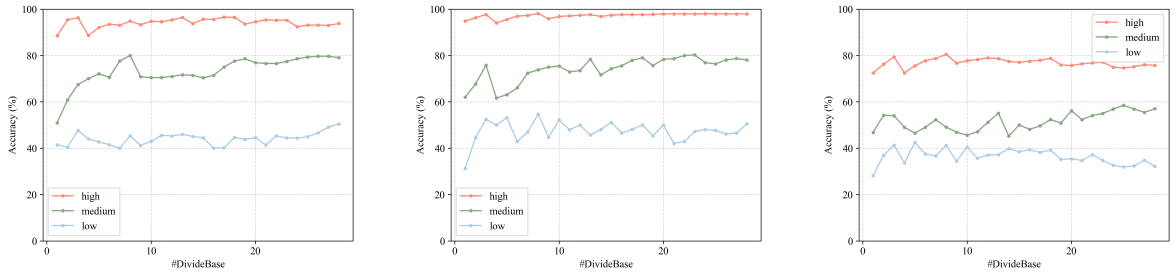
3.4.2 Deep study for self-consistency, #DivideBase and divide results

Self-consistency can effectively divide dataset. With the enhancement of LLM's capability, many problems do not require sampling extensive repetitive reasoning paths in self-consistency, as illustrated in Fig. 4. Based on the results of the first five inference (#SC=5), data with $\mathcal{CS} \geq 0.8$ was classified into the high confidence subset, while the rest was assigned to the low confidence subset. Across various datasets, it was consistently observed that the accuracy of the high confidence subsets tends to stabilize after #SC=3, whereas the low confidence subset continues to rise as the number of #SC increases. Furthermore, the model performs substantially better on the high confidence subset compared to the low confidence one. This highlights enhancing the accuracy on the low confidence subset has a significant impact for the overall performance, while only a few sampling iterations are needed for the high confidence subset. Therefore,



(a) AQuA (b) CMSQA (c) RiddleSense

Figure 4: Problem-solving accuracy of high and low confidence subsets under different #SC. We take the results from $\#SC = 5$ as the basis for confidence division. The answer, whose frequency is less than 0.8, is considered low confidence otherwise is of high confidence. "#SC" denotes the number of inference times.



(a) AQuA (b) OpenBookQA (c) Reclor

Figure 5: Prior accuracy of different confidence subsets after divide stage based on different #DivideBase.

we can make use of self-consistency to divide questions avoiding redundancy of high confidence subset and providing more fine-grained process for low confidence one.

Effect of different #DivideBase on Prior accuracy. As Sec. 3.3, the improvements realized in the Conquer phase primarily depend on the results of the Divide phase. A key measure reflecting the effectiveness of different confidence subsets division is the Prior accuracy, where higher confidence should correlate with higher Prior accuracy. Consequently, we selected three datasets: AQuA, OpenBookQA, and Reclor, to observe the impact of varying #DivideBase from 3 to 30 on Prior accuracy, as illustrated in Fig. 5. There is a clear distinction in accuracy across different confidence subsets regardless of the #DivideBase, and only minimal inferences are needed to enter the oscillatory zone. This suggests that as the model's capabilities enhance, setting #DivideBase based on the initial length of the choices list to ensure coverage of all options in the worst scenario is reasonable.

Further divide for low confidence subsets $\mathbb{D}_{low}$. In Table 2, we divide all questions with $\mathcal{CS} \leq 0.6$ into $\mathbb{D}_{low}$, which deserve a deeper analysis. Consequently, we further divided them of nine datasets into two subcategories: Low-Top and Low-Bottom, where Low-Top encompasses questions with $0.4 < \mathcal{CS} \leq 0.6$, while Low-Bottom covers the remaining items, as shown in Table 4. Based on the division setting, Low-Top has a relatively higher confidence level, hence its Prior weighted average accuracy surpasses Low-Bottom by 14.63%. Similarly, the improvement achieved in the Conquer phase is only 5.91%, which is 3.45% lower than the 9.36% of Low-Bottom. This reaffirms the effectiveness of our strategy: using $\mathcal{CS}$ to divide the dataset and focusing on in-depth process of the low confidence subset. Additionally, the performance of PKR on the Low-Bottom subset is 8.29% lower than FCR, primarily due to the lower confidence of this subset leading to an excessive number of rationales being included in the prompt. This results in overly lengthy prompts, which in turn hinders the model to extract useful information from the prior knowledge.

Table 4: Accuracy (%) of deeper division for $\mathbb{D}_{low}$. "Top" denotes questions with $0.4 < \mathcal{CS} <= 0.6$, while "Bottom" refers to questions with $\mathcal{CS} <= 0.4$. For some bottom subsets, "-" indicates insufficient data which results in a lack of reliability in answers.

Subset	Setting	Arithmetic			Commonsense			Logic			Weighted Average
		AQ.	Alg.	Math.	CMS.	OB.	ARC.	Rid.	Logi.	Rec.	
Top	#NUM	73	33	107	271	71	101	246	145	120	-
	Prior	56.16	30.30	41.12	52.77	46.48	49.50	50.81	36.55	35.83	46.44
	ZTCoT	49.32	28.28	39.25	46.13	51.64	38.94	40.51	28.51	41.11	41.02
	PKR	56.16	**39.39**	**43.93**	46.86	50.70	**54.46**	**48.24**	**39.08**	41.11	46.59
	FCR	**56.62**	37.37	40.81	**52.28**	**56.81**	51.16	44.31	35.63	**46.67**	**46.93**
Bottom	#NUM	57	0	3	102	3	3	160	44	2	-
	Prior	36.84	-	-	33.33	-	-	33.12	25.00	-	31.81
	ZTCoT	38.01	-	-	40.52	-	-	33.54	31.06	-	34.85
	PKR	35.67	-	-	38.24	-	-	38.96	28.79	-	35.92
	FCR	**48.54**	-	-	**53.27**	-	-	**41.04**	**40.15**	-	**44.21**

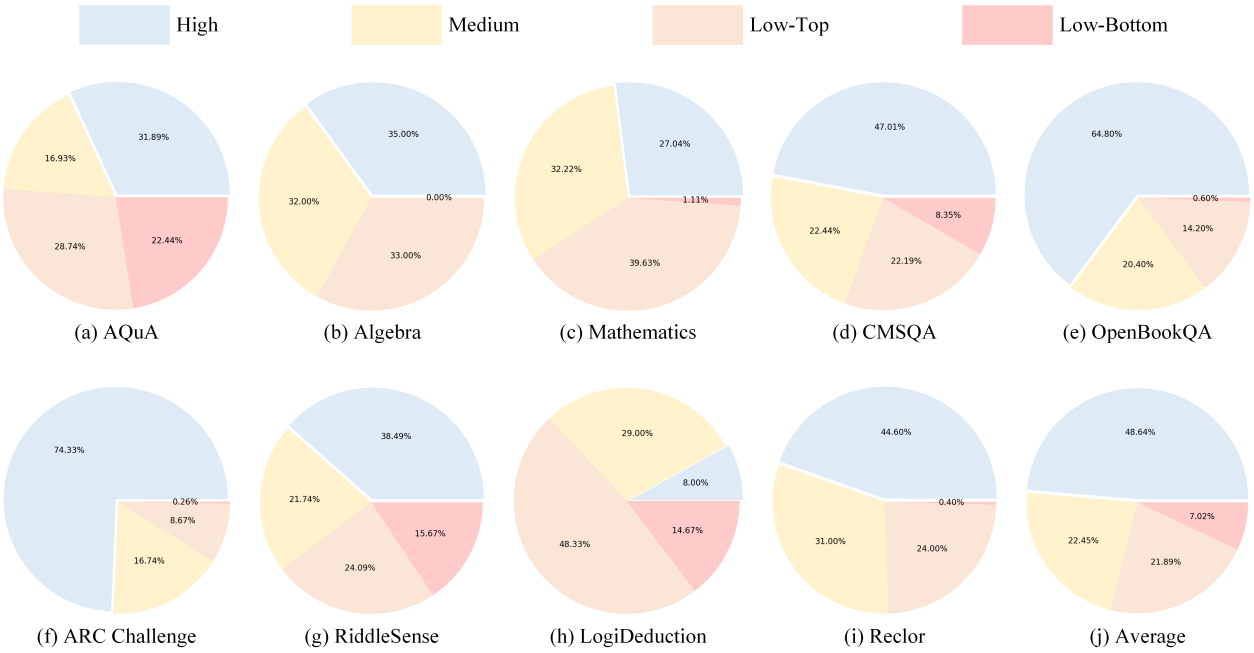


Figure 6: The proportion of different confidence subsets in various datasets. "High", "Medium", "Low-Top" and "Low-Bottom" denote four confidence subsets based on $\mathcal{CS}$.

The proportion of different confidence subsets. Incorporating the results of dataset division from Table 2 and Table 4, we conducted a visual statistical analysis for proportion of different confidence subsets, as illustrated in Fig. 6. The high confidence subsets, represented by the light blue area, account for more than a quarter in different datasets except LogiDeduction, and have an average proportion nearing 50% across all datasets. Therefore, by utilizing the Divide and Conquer strategy, we can effectively avoid redundant process on high confidence subsets, instead concentrating more resources on the low confidence subsets (Low-Top & Low-Bottom) that constitute less than 30% on average, which significantly reduces overall expenditure.

Table 5: Accuracy (%) of various rationale sampling strategies in PKR among different $\mathbb{D}_{low}$. Results in **bold** and underline indicate the best and sub-optimal performance respectively.

Strategy	Arithmetic			Commonsense			Logic			Average
	AQ.	Alg.	Math.	CMS.	OB.	ARC.	Rid.	Logi.	Rec.	
Longest	**47.18**	**39.39**	**44.24**	44.50	51.35	**54.49**	**44.58**	**36.68**	40.44	**44.76**
Random	45.90	30.30	37.88	43.79	50.90	39.74	38.51	32.10	39.89	39.89
Shortest	43.08	28.28	37.88	**44.95**	**51.80**	39.10	38.92	31.39	**41.26**	39.63

Table 6: Problem-solving accuracy (%) for unsolved subsets with different construction methods of choices list.

Dataset	FullChoices	Random Sample		w Prior	w/o Prior	w/o Prior & 2 Choices
		3 Choices	2 Choices			
AQuA	2.16	21.65	29.00	18.18	26.41	**29.39**
CMSQA	0.97	47.58	62.10	29.46	65.01	**75.52**
OpenBookQA	0.57	31.03	45.98	32.76	**58.05**	48.85
RiddleSense	1.97	28.11	**49.25**	24.04	44.02	45.70
Average	1.42	32.09	46.58	26.11	48.37	**49.87**

3.4.3 Deep study for rationale and prompts length in PKR

Longer rationale yields superior results. For various rationales obtained from the previous t times inference, we sample only one item for the same answer to serve as prior knowledge. Therefore, we conducted an analysis for different selection strategies, with results presented in Table 5. Compared to random sampling and choosing the shortest rationale, selecting the longest one yielded significant improvements across two-thirds of the datasets, boasting an average accuracy of 44.76% that is 4.87% higher than second-best strategy, random selection. Derived from Complex CoT Fu et al. (2022), the model is influenced by its training data and inevitably takes shortcuts during inference. So using rationales that contain these shortcuts as references will hamper the model's ability to extract effective knowledge, leading to a decline in performance.

Shorter prompts provide optimal efficiency. The PKR attempts to incorporate prior knowledge into the prompts, aiming to extract more information through previously generated rationales, and guiding the model to select the accurate answer through analysis and comparison. However, compared to FCR, PKR generally underperforms across various scenarios, and COM1 with prompt originating from PKR is generally inferior than COM2 from FCR. Furthermore, FCR has superior performance over both COM1 and COM2 in most cases, as displayed in Table 2. These primarily stems from current LLMs prefer to handle concise queries, while extended inputs often result in challenges like forgetting of conditions and oversight of details. Recently, both SelfCheck (Miao et al., 2023) and Natural Program (Ling et al., 2023) have indicated that it is highly inefficient for models to directly verify the complete reasoning chains they generate at once. Empirical study from Natural Program have shown that GPT-3.5-turbo is only able to identify 7% of the errors in faulty chains. Hence, these works are dedicated to breaking down complex reasoning processes into smaller steps for the model to examine one by one. In summary, when utilizing LLMs for reasoning, we have to take a trade-off between cost and accuracy, while striving to interact with the model through concise prompts and solve the problem with minimal content generation.

3.4.4 Deep study for irrelevant choices in FCR

Irrelevant information may distract LLM. Shi et al. (2023) investigated the sensitivity of LLM to irrelevant information within questions and validated various methods suffering a decline due to such distraction. They also proposed an instruction sentence "feel free to ignore irrelevant information in the problem description" and added exemplars containing irrelevant information can effectively reduce LLM's distractibility. In fact, this irrelevant information is not solely limited to the questions' context, but also contained in options

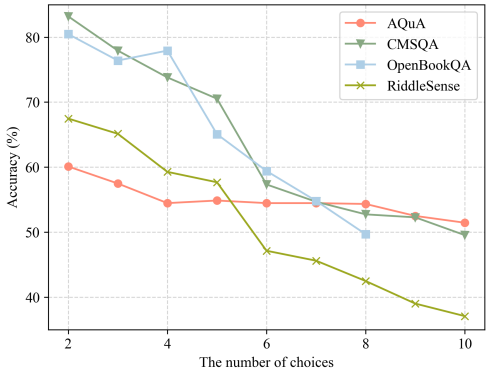


Figure 7: Problem-solving accuracy for different number of choices. We extend choices list by randomly combining incorrect choices.

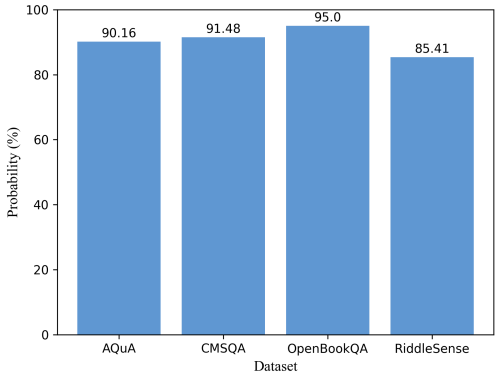


Figure 8: The probability of containing the correct answer when retaining choices from prior inference based results.

list. Therefore, we conducted an analysis on three datasets consisting of MCQs spanning both arithmetic, commonsense and logical categories, and we take a statistic under different choices in Fig. 7. As we can see, the model's performance exhibits a noticeable decline as the number of incorrect options increases. Furthermore, in order to investigate the impact of choices list's length on the problem-solving capabilities of LLM deeply, we divided subsets from AQuA, CMSQA, OpenBookQA, and RiddleSense that the model was unable to solve with majority voting based on 5 inferences. Then we conducted the following setting: 1) Provide the full choices list to the model for resolution. 2) Construct the choices list by randomly sampling 1 or 2 incorrect options, while retaining the correct option. 3) Create the new choices list by combining the correct option and deduplicated results from prior five inference. 4) Use the correct option and choices not present in the previous generated results to construct the choices list. 5) Based on 4), retain the correct option and randomly pick one from the remaining choices to create the final list containing only two choices. As shown in Table 6, the accuracy for data with full choices list close to 0, while retaining a small subset of options through random sampling or prior results based filtering can significantly enhance performance. However, this is an ideal scenario where we can obtain the correct answers for the test set. Therefore, in FCR, we utilize results from previous inference to filter the choices and we quantified the probability of the correct answer contained in remaining options, as shown in Fig. 8. There are average 90.51% of the data retaining the correct answers, which indicates that FCR can be considered as an approximation, albeit still containing those answers that once led the model into illusions.

Fewer choices lead to better outcomes. Reviewing the results presented in Table 2, we observed that FCR demonstrates outstanding performance across both $\mathbb{D}_{med}$ and $\mathbb{D}_{low}$, irrespective of applying self-consistency, which highlights that irrelevant information embedded within the choices list can detrimentally impact the model's performance. In fact, employing prior inference results to eliminate incorrect options guides the model in accurately identifying the correct answer, which aligns with human cognitive process of eliminating incorrect choices until viable one remains. Furthermore, Shi et al. (2023) also noted that LLM is more sensitive to the irrelevant information which has higher degree of lexical overlap with the original question. This phenomenon is applicable to the choices list as well, where different options exert varying degrees of interference on the model. It is evident that incorrect options previously chosen have a deeper disruption on the model compared to those not selected. Thus, as shown in Table 6, there is a significant improvement from the "w Prior" to the "w/o Prior", with an average increase of 22.26% across four datasets. Moreover, retaining two choices consistently surpasses three choices in random sample, which can be attributed to a lower probability of strong distractors being present when only two options

Table 7: The probability of the prior results (strong distractors) appearing in the choices list after random sampling.

Setting	AQ.	CMS.	OB.	Rid.
2 Choices	22%	23%	12%	29%
3 Choices	51%	46%	26%	61%

are reserved, as illustrated in Table 7. Therefore, exploring more effective ways to eliminate those strongly distracting options will become a significant research direction for us in the future.

3.4.5 Exploration of lower resource and better generalizability

Divide based on less inferences. Utilizing confidence score $\mathcal{CS}$ of specific questions to divide dataset is a rational and effective method, yet deriving $\mathcal{CS}$ through self-consistency incurs substantial costs. Consequently, we would explore how to complete the dataset partition with less inferences. Given that limited queries might not produce highly reliable $\mathcal{CS}$, we only divided the high and low confidence subset. Specifically, we proposed a prompt "Let's substitute the answer back into the question to check it is 'true' or 'false':", asking the model to directly judge the confidence of the question based on checking the single answer generated previously. Testing on the AQuA, CMSQA, and RiddleSense, we discovered this prompt is effective, achieving an average accuracy of 73.02% for high subset and 35.07% for low with only two-fifth of the computational overhead (#DivideBase=5 in our original strategy), as illustrated in Table 8. In the future, we will concentrate on how to better categorize datasets based on a single inference and guide fine-grained process with even lower budget.

Table 8: Prior accuracy (%) for different confidence subsets based on less inferences.

Subsets	AQ.	CMS.	Rid.	Avg.
high	77.86	73.23	67.98	73.02
low	35.96	39.42	29.83	35.07

Application beyond MCQs. Although we had conducted experiments on nine datasets across three categories to validate the effectiveness of our method, all these datasets comprise MCQ, meaning that the correct answer is guaranteed within the options list. Consequently, we explored extending the Divide and Conquer to GSM8K (Cobbe et al., 2021), a high quality dataset of cloze-style grade school math questions. Initially, we queried the entire test set of GSM8K 5 times, consistent with AQuA, and utilized the generated answers to construct choices list for each problem. This process resulted in a dataset, GSM8K-MCQ, formally equivalent to MCQ. We could then proceed to divide GSM8K-MCQ, and applied our methods for a deeper conquer on the subset. Based on the prior accuracy from high confidence to low subsets, which are 97.42%, 88.79%, and 46.73% respectively, we validate PKR, FCR, and COM2 only on the low confidence subset without self-consistency. Since the choices list for GSM8K-MCQ is constructed based on previous results, PKR and COM1 are equivalent in this case. From Table 9, FCR achieved significant improvements of 8.31% compared to Zero-Shot-CoT, even superior than results of Prior with self-consistency, which indicates effectively applicable of Divide and Conquer to datasets beyond MCQs.

Table 9: Accuracy (%) on GSM8K with our methods.

Subsets	Prior	ZTCoT	PKR	FCR	COM2
Low	46.73	38.94	34.16	**47.25**	32.92

4 Related Work

4.1 CoT prompting in LLMs reasoning

Recently, CoT prompting methods have significantly enhanced the multi-step reasoning abilities of LLMs by generating intermediate reasoning steps before arriving at the final answer, without any parameters update. As the pioneer of CoT prompting, CoT (Wei et al., 2022) greatly augmented reasoning performance on complex tasks by integrating rationales into few-shot examplars. Following CoT, self-consistency (Wang et al., 2022), LTM (Zhou et al., 2022), ToT (Yao et al., 2023), GoT (Besta et al., 2023), AoT (Sel et al., 2023), Tab-CoT (Jin & Lu, 2023), ResPrompt (Jiang et al., 2023), COP (Yan et al., 2023) and Deb et al. (2023) are all dedicated to optimizing the thinking process. PAL (Gao et al., 2023), PoT (Chen et al., 2022), ChatCoT (Chen et al., 2023), LPML (Yamauchi et al., 2023) and Jie et al. (2023) employ external tools to disentangle computation from LLMs, thereby unlocking greater reasoning capabilities. Auto-CoT (Zhang et al., 2022), Active-Prompt (Diao et al., 2023), Automate-CoT (Shum et al., 2023), Iter-CoT (Sun et al., 2023), Meta-CoT (Zou et al., 2023) are exploring how to guide models in reasoning through the construction of examplars in distinct manners. RCoT (Xue et al., 2023), SelfCheck (Miao et al., 2023), Self-Convince

(Zhang et al., 2023), Natural Program (Ling et al., 2023) and Self-Verification (Weng et al., 2023) build upon the model's self-correction capabilities, introducing the concept of verification into the community. In addition, Shi et al. (2023) delves into the distractibility of LLMs by irrelevant context in questions and suggests that LLMs are able to ignore this redundancy through demonstrations or instructions. HtT (Zhu et al., 2023) proposes a two-stage method that comprises an inductive phase for generating a rule library from training data and a deductive phase for utilizing these rules to problem-solving, aiming to addresses issues of LLMs' implicit knowledge inaccuracies or inconsistencies with the tasks' requirements. GV-consistency (Li et al., 2023b) identifies the prevalent inconsistency between generating and validating an answer in LLMs and proposes using fine-tuning to mitigate the models' hallucinations. However, all the aforementioned methods overlook the issue: not every question in even the most complex datasets requires special handling, and in the vast majority of cases, the simplest methods suffice. Therefore, we propose a novel perspective on LLMs reasoning—Divide and Conquer—wherein the dataset is first partitioned, and then select subsets to deeply process.

4.2 Prior knowledge can improve reasoning

Unlike Zero-Shot-CoT (Kojima et al., 2022) that stimulates the model's reasoning abilities through designed prompts "Let's think step by step.", CoT (Wei et al., 2022) achieves superior performance by using manually constructed examplars, which not only spark the model to generate intermediate rationale but also provide it with some prior knowledge. Works like Active-Prompt (Diao et al., 2023), Automate-CoT (Shum et al., 2023), and Iter-CoT (Sun et al., 2023) are dedicated to finding better examples to guide the model, essentially exploring what kind of prior knowledge is more beneficial for problem-solving. Recent research, such as EchoPrompt (Mekala et al., 2023), Step-Back Prompting (Zheng et al., 2023c), SEC (Li et al., 2023a), Analogical Prompting (Yasunaga et al., 2023), AgentInstruct (Crispino et al., 2023) aims to enable the model to generate prior knowledge by themselves, thereby enhancing the model's zero-shot performance. PHP (Zheng et al., 2023a) proposes a method that utilize previously generated answers as hints to progressively guide the model to the correct answer, while Role-Play Prompting (Kong et al., 2023) defines specific roles for the model based on the particular task. In fact, they design elaborate prompts both in order to introduce prior knowledge to the model. These underscores the pivotal role of prior knowledge for enhancing model performance in downstream tasks. Therefore, after dividing the dataset based on confidence using several inference results, we can fully utilize the previous rationale as prior knowledge to guide the subsequent problem-solving process.

4.3 LLM reasoning for MCQs

As a problem format that lists alternative answers, MCQs are prevalent in the real world and have been extensively applied to the reasoning scenarios of LLMs. And it has given rise to numerous related datasets, such as MMLU (Hendrycks et al., 2020), BIG-bench (Srivastava et al., 2022), AGIEval (Zhong et al., 2023), CEVAL (Huang et al., 2023). Simultaneously, many works have emerged in the MCQs community. Robinson et al. (2022) explores how presenting the model with question and the choices list jointly, and then guiding the model to output the symbol corresponding to the chosen option, can lead to a marked improvement. Pezeshkpour & Hruschka (2023) has discovered that LLMs exhibit a pronounced position bias, meaning that the order of the choices can significantly impact the model's performance. PriDe (Zheng et al., 2023b) has found selection bias, where LLMs display a clear preference for choosing options from specific positions. Our work explores the model's sensitivity to the number of options in MCQs and verifies that filtering incorrect choices can yield benefits.

5 Conclusion

In this paper, we propose a novel perspective on LLMs reasoning: Divide and Conquer, which categorize problems based on confidence score $\mathcal{CS}$ from the results of prior inference and subsequently delving deeper into not fully resolved questions. For this purpose, we introduce two methods: Prior Knowledge based Reasoning (PKR) and Filter Chioces based Reasoning (FCR), and two integration variants, COM1 and COM2, based on different merging strategies. Evaluation results on nine datasets across three tasks prove

that our method not only effectively reduces redundant inference on simple problems but also enhances accuracy on more challenging ones. In addition, through extensive analysis, we verify confidence score's positive relation with accuracy, longer rationale yielding superior results and fewer choices leading to better outcomes. However, utilizing previously generated results to filter choices can not effectively eliminate strong distractors and computing confidence score $\mathcal{CS}$ based on self-consistency incurs substantial costs. Therefore, we will explore more powerful ways to exclude distracting options and lower resource to divide datasets, as well as encourage more researchers to drive progress within the community.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Jon Louis Bentley. Multidimensional divide-and-conquer. *Communications of the ACM*, 23(4):214–229, 1980.

Jon Louis Bentley and Michael Ian Shamos. Divide-and-conquer in multidimensional space. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pp. 220–230, 1976.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Zhipeng Chen, Kun Zhou, Beichen Zhang, Zheng Gong, Wayne Xin Zhao, and Ji-Rong Wen. Chatcot: Tool-augmented chain-of-thought reasoning on chat-based large language models. *arXiv preprint arXiv:2305.14323*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Nicholas Crispino, Kyle Montgomery, Fankun Zeng, Dawn Song, and Chenguang Wang. Agent instructs large language models to be general zero-shot reasoners. *arXiv preprint arXiv:2310.03710*, 2023.

Aniruddha Deb, Neeva Oza, Sarthak Singla, Dinesh Khandelwal, Dinesh Garg, and Parag Singla. Fill in the blank: Exploring and enhancing llm capabilities for backward reasoning in math word problems. *arXiv preprint arXiv:2310.01991*, 2023.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*, 2023.

Michael Eisenstein. Divide and conquer. *Nature*, 441(7097):1179–1179, 2006.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Michael Heideman, Don Johnson, and Charles Burrus. Gauss and the history of the fast fourier transform. *IEEE Assp Magazine*, 1(4):14–21, 1984.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Yi Hu, Haotong Yang, Zhouchen Lin, and Muhan Zhang. Code prompting: a neural symbolic method for complex reasoning in large language models, 2023.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

Song Jiang, Zahra Shakeri, Aaron Chan, Maziar Sanjabi, Hamed Firooz, Yinglong Xia, Bugra Akyildiz, Yizhou Sun, Jinchao Li, Qifan Wang, et al. Resprompt: Residual connection prompting advances multi-step reasoning in large language models. *arXiv preprint arXiv:2310.04743*, 2023.

Zhanming Jie, Trung Quoc Luong, Xinbo Zhang, Xiaoran Jin, and Hang Li. Design of chain-of-thought in math problem solving. *arXiv preprint arXiv:2309.11054*, 2023.

Ziqi Jin and Wei Lu. Tab-cot: Zero-shot tabular chain of thought. *arXiv preprint arXiv:2305.17812*, 2023.

Donald Ervin Knuth. Sorting and searching. *The art of computer programming*, 3, 1998.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*, 2023.

Rui Li, Guoyin Wang, and Jiwei Li. Are human-generated demonstrations necessary for in-context learning? *arXiv preprint arXiv:2309.14681*, 2023a.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. *arXiv preprint arXiv:2310.01846*, 2023b.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376*, 2021.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. *arXiv preprint arXiv:2306.03872*, 2023.

Thomas E Mallouk. Divide and conquer. *Nature chemistry*, 5(5):362–363, 2013.

Rajasekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. Echoprompt: Instructing the model to rephrase queries for improved in-context learning. *arXiv preprint arXiv:2309.10687*, 2023.

Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*, 2023.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

Laurence J Peter, Raymond Hull, et al. *The peter principle*, volume 4. Souvenir Press London, 1969.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*, 2022.

Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Lu Wang, Ruoxi Jia, and Ming Jin. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

KaShun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *arXiv preprint arXiv:2302.12822*, 2023.

Douglas R Smith. The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58, 1985.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. *CoRR, abs/2212.09561*, 2023.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. *arXiv preprint arXiv:2305.11499*, 2023.

Ryutaro Yamauchi, Sho Sonoda, Akiyoshi Sannai, and Wataru Kumagai. Lpml: Llm-prompting markup language for mathematical reasoning. *arXiv preprint arXiv:2309.13078*, 2023.

Shaotian Yan, Chen Shen, Junjie Liu, and Jieping Ye. Concise and organized perception facilitates large language models for deductive reasoning. *arXiv preprint arXiv:2310.03309*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*, 2023.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv:2002.04326*, 2020.

Haodi Zhang, Min Cai, Xinhe Zhang, Chen Jason Zhang, Rui Mao, and Kaishun Wu. Self-convinced prompting: Few-shot question answering with repeated introspection. *arXiv preprint arXiv:2310.05035*, 2023.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*, 2023a.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. On large language models' selection bias in multi-choice questions. *arXiv preprint arXiv:2309.03882*, 2023b.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*, 2023c.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023d.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*, 2023.

Anni Zou, Zhuosheng Zhang, Hai Zhao, and Xiangru Tang. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models. *arXiv preprint arXiv:2310.06692*, 2023.

A Continuously improving zero-shot performance.

With the increasing scale of LLMs, their performance is correspondingly advancing, leading to a notable phenomenon where model's zero-shot capabilities on downstream tasks are gradually nearing or even surpassing their few-shot counterparts. By comparing the performance of LLM across multiple datasets in both Zero-Shot-CoT (Kojima et al., 2022) and Few-Shot-COT (i.e. CoT) (Wei et al., 2022) settings in Table 10, our findings align with the conclusions in recent research on LLM's reasoning (Hu et al., 2023; Zhong et al.,

Table 10: Problem-solving accuracy (%) between Zero-Shot-CoT and Few-Shot-CoT across various datasets.

Method	AQuA	GSM8K	SVAMP	CMSQA	Average
Zero-Shot-CoT	54.86(±0.67)	79.33(±0.34)	78.20(±1.82)	69.67(±0.77)	70.52
Few-Shot-CoT	53.67(±0.67)	79.67(±1.18)	81.60(±1.34)	77.47(±0.96)	73.10

2023). Therefore, our work focuses on the setting that is entirely free from human intervention, void of data labels, and circumvents exemplars construction, while attaining efficiency and performance superior to conventional methods.