Aggregation of Reasoning: A Hierarchical Framework for Enhancing Answer Selection in Large Language Models

Zhangyue Yin[◇], **Qiushi Sun**[♡], **Qipeng Guo**[☆], **Zhiyuan Zeng**[◇]
Xiaonan Li[◇], **Tianxiang Sun**[◇], **Cheng Chang**[◇], **Qinyuan Cheng**[◇],
Ding Wang[♣], **Xiaofeng Mou**[♣], **Xipeng Qiu**[◇✉], **Xuanjing Huang**[◇]
[◇]School of Computer Science, Fudan University [♡]National University of Singapore
[☆]Shanghai AI Laboratory [♣]Midea AI Research Center
{yinzy21,cengzy23,changc21,chengqy21}@m.fudan.edu.cn qiushisun@u.nus.edu
guoqipeng@pjlab.org.cn {ding2.wang, mouxf}@midea.com
{lixn20,txsun19,xpqiu,xjhuang}@fudan.edu.cn

Abstract

Recent advancements in Chain-of-Thought prompting have facilitated significant breakthroughs for Large Language Models (LLMs) in complex reasoning tasks. Current research enhances the reasoning performance of LLMs by sampling multiple reasoning chains and ensembling based on the answer frequency. However, this approach fails in scenarios where the correct answers are in the minority. We identify this as a primary factor constraining the reasoning capabilities of LLMs, a limitation that cannot be resolved solely based on the predicted answers. To address this shortcoming, we introduce a hierarchical reasoning aggregation framework *AoR* (Aggregation of Reasoning), which selects answers based on the evaluation of reasoning chains. Additionally, *AoR* incorporates dynamic sampling, adjusting the number of reasoning chains in accordance with the complexity of the task. Experimental results on a series of complex reasoning tasks show that *AoR* outperforms prominent ensemble methods. Further analysis reveals that *AoR* not only adapts various LLMs but also achieves a superior performance ceiling when compared to current methods.

Keywords: Large Language Models, Complex Reasoning, Reasoning Chains Evaluation

1. Introduction

Large Language Models (LLMs) have driven remarkable advancements across various Natural Language Processing (NLP) tasks (OpenAI, 2023; Chowdhery et al., 2022; Touvron et al., 2023a,b; Huang et al., 2022; Zhao et al., 2023). Nonetheless, there remains a discernible gap between the performance of these models and human-level expertise in reasoning tasks (Cobbe et al., 2021; Valmeekam et al., 2022), which cannot be bridged merely by increasing the model's scale (Rae et al., 2022). In this context, the advent of Chain-of-Thought (CoT) prompting (Wei et al., 2022b) technique heralds a stride towards mitigating this disparity. Rather than employing "answer-only" prompts, CoT drives LLMs to generate a series of intermediate steps that lead to the final answer. By decoupling the problem-solving process, CoT not only simplifies the complexity of each step but also offers a novel perspective to addressing complex reasoning tasks.

Beyond the inherent limitations of LLMs (Yin et al., 2023b), Wang et al. (2023d) observe that the CoT exhibits randomness when utilizing a single reasoning chain. As a remedy, they propose modulating the sampling temperature to collect a diverse set of reasoning chains, and then select the most consistent answer as the final prediction.

✉ Corresponding author.

Question: *The difference between simple interest and C.I. at the same rate for Rs.5000 for 2 years in Rs.72. The rate of interest is?*
Answer Choices: (A) 10% (B) 12% (C) 6% (D) 8% (E) 4%

$\boldsymbol{R_1}$: Let the rate of interest be r%. The compound interest would be 5000[(1 + r/100)^2 - 1]... Solving this quadratic equation gives us r = 10%. So the answer is (A).	$\boldsymbol{R_2}$: Let the rate of interest be r. ..|(5000 * r / 100) * 2 - 5000 * ((1 + r/100)^2 - 1)| = 72. Simplifying this equation gives r = 12. So the answer is (B). ✓
$\boldsymbol{R_3}$: Let the rate of interest be r. The difference between simple interest and compound interest at the same rate for 2 years... Since the rate of interest cannot be negative, the answer is 6%. So the answer is (C).	$\boldsymbol{R_4}$: Let the rate of interest be x%. The simple interest for 2 years would be (5000 * 2 * x) / 100 = 100x... So, 5000 * (1 + x/100)^2 - 5000 - 100x = 72. Solving this equation, we get x = 6%. So the answer is (C).
$\boldsymbol{R_5}$: C.I. - Simple Interest = 72... Since interest cannot be negative, we take the solution r = 8%. So the answer is (D).	Correct Answer: (B). Majority Vote: (C). ✗

Figure 1: An illustrative example from AQuA (Ling et al., 2017), with 5 reasoning chains generated through temperature sampling. Although LLM is able to generate the correct answer, majority voting ultimately selects an incorrect answer due to the abundance of incorrect answers.

This ensemble approach based on majority-voting has not only elevated the reasoning capability of LLMs but has also emerged as the predominant paradigm for LLMs in reasoning tasks (Chu et al., 2023; Yu et al., 2023).

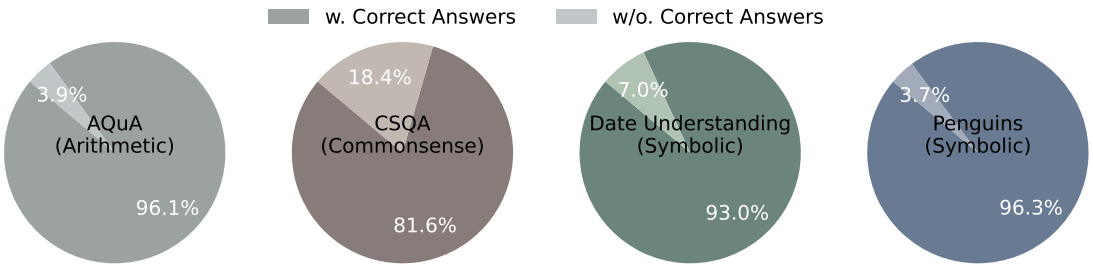


Figure 2: Proportion of samples that correct answers appearing in LLMs' generations among those where majority voting results in an incorrect outcome across various reasoning tasks.

However, when confronted with more complex questions, LLMs often waver among multiple answers. A dilemma arises when the incorrect answers outnumber the correct ones. Even if the LLM is capable of generating the right answer, the majority voting mechanism remains susceptible to skewing the final prediction towards an erroneous one. Figure 1 showcases an illustrative example from the AQuA dataset (Ling et al., 2017). Among the five sampled reasoning chains, four candidate answers: (A), (B), (C), and (D) are generated. While the LLM is capable of generating the correct answer (B) in $\mathcal{R}_2$, the overwhelming presence of erroneous candidates eventually led to the selection of the incorrect answer (C).

To explore this phenomenon, we conduct a pilot analysis on samples spanning various reasoning tasks, where the majority voting results in incorrect predictions. As depicted in Figure 2, over 80% of the samples that LLM has the potential to answer correctly, but majority voting fails. Notably, in AQuA (Ling et al., 2017) and Penguins (Suzgun et al., 2023) datasets, this proportion exceeds 95%. These findings indicate that ensembling reasoning chains, which relies on the frequency of answers, still has significant room for improvement.

Motivated by the observed limitations, we pose the central research question of this work: *"When LLMs are capable of generating the correct answer, how can we mitigate the interference of incorrect answers to accurately select the right one?"* In situations polluted by a myriad of erroneous predictions, relying exclusively on the answers themselves provides limited insight for enhanced accuracy. Consequently, it becomes both essential and promising to focus on the process leading to these answers: the reasoning chains. Thus, we introduce a hierarchical reasoning aggregation framework *AoR* (Aggregation of Reasoning), designed to harness the LLM's ability to evaluate reasoning processes in order to improve the selection of the final answer.

Specifically, given the constraints of LLM's context window (Liu et al., 2023a) that prevents simultaneous evaluation of all reasoning chains, *AoR* initiates by aggregating chains based on their respective answers followed by a two-phase evaluation process. In the first phase: local-scoring, chains yielding identical answers are evaluated. Since the answers are consistent, the evaluation places greater emphasis on the soundness of the reasoning process and the appropriateness of the reasoning steps. For the second phase: global-evaluation, the most logically coherent and methodically valid chains from different answer groups are jointly assessed. The objective is to identify the reasoning chain that best exhibits coherence and consistency between the reasoning process and its corresponding answer, thereby designating this answer as the final output.

Furthermore, leveraging the scores derived from the global evaluation phase, *AoR* can estimate the current confidence level of the LLM in its optimal reasoning process and answer. This allows *AoR* to dynamically decide whether it is necessary to sample additional reasoning chains. Experimental results across various reasoning tasks demonstrate *AoR*'s effectiveness in significantly enhancing the reasoning performance of LLMs. Benefited from dynamic sampling, which determines the number of sampling and evaluations by distinguishing between easy and challenging samples, *AoR* also effectively curtails the LLM's reasoning overhead, establishing a balance between performance and computational cost.

The main contributions are listed below:

- We identify that the existing ensemble mechanism, which solely relies on the frequency of answers, is insufficient. This observation underscores the importance of incorporating the reasoning process, leading to the design of our hierarchical reasoning process aggregation framework *AoR*.
- Leveraging the evaluation scores of the optimal reasoning chains, *AoR* integrates the ability to dynamically sample reasoning chains, efficiently minimizing the reasoning overhead.
- Extensive experimental results demonstrate *AoR*'s superior performance and cost efficiency compared to existing reasoning chain ensemble methods.

Feature	*AoR* (our work)	Self-Consistency (Wang et al., 2023d)	ComplexSC (Fu et al., 2023b)	PHP (Zheng et al., 2023)	DIVERSE (Li et al., 2023b)
Task Agnostic?	✓	✓	✓	✗	✓
Training-Free?	✓	✓	✓	✓	✗
Plug-and-Play?	✓	✓	✓	✗	✗
Dynamic Sampling?	✓	✗	✗	✓	✗

Table 1: A comparison of *AoR* to other reasoning chains ensemble methods.

2. Related work

Reasoning with Chain-of-Thought. Chain-of-Thought (CoT; Wei et al., 2022b) prompting has emerged as a pivotal technique for eliciting reasoning capabilities in LLMs (Zhao et al., 2023; Liang et al., 2023). When guided by samples enriched with explicit reasoning steps, LLMs can produce a series of intermediate steps culminating in a multi-step solution (Zhou et al., 2023). Remarkably, CoT can enhance the performance of LLMs in reasoning tasks without necessitating additional training (Huang and Chang, 2022; Min et al., 2022). This characteristic has swiftly garnered widespread attention (Qiao et al., 2023; Chu et al., 2023), with several studies attributing this phenomenon to the emergent capabilities intrinsic to LLMs (Wei et al., 2022a; Kaplan et al., 2020). Subsequent research has concentrated on strengthening the consistency between reasoning paths and answers (Chen et al., 2022; Gao et al., 2022), automating the construction of prompts (Zhang et al., 2023; Li et al., 2023a; Diao et al., 2023), eliciting external knowledge (Wang et al., 2023b; Li and Qiu, 2023) and progressively refining the reasoning processes (Yao et al., 2023; Besta et al., 2023; Sel et al., 2023; Han et al., 2023; Liu et al., 2023b).

Ensemble of Multiple Reasoning Chains. Wang et al. (2023d) identify the randomness in the CoT's single-chain sampling process and subsequently propose the Self-Consistency method. This approach entails sampling multiple reasoning chains and selecting the most frequently occurring answer as the final output, which lays the foundation for a series of reasoning chain ensemble methods. Fu et al. (2023b) observe a positive correlation between the complexity of reasoning chains and the accuracy of generated answers. Based on this insight, they propose filtering reasoning chains based on their complexity before employing a majority voting mechanism for the answers. Furthermore, Li et al. (2023b) train a verifier to score each reasoning chain. The answer corresponding to the highest-scoring reasoning chain is selected as the final output. From a different perspective, Zheng et al. (2023) suggest using previously generated answers as hints to guide LLMs toward producing accurate answers. Furthermore, recent advancements have seen the emergence of strategies encouraging interaction among reasoning chains (Yin et al., 2023a) or transforming LLMs into multiple agents to benefit from diverse cognitive processes (Sun et al., 2023). A comparison of *AoR* with some representative reasoning chain ensemble methods is presented in Table 1. Notably, our method is task-agnostic and does not require additional annotation for training. This plug-and-play characteristic, coupled with dynamic sampling, ensures the functionality and cost-effectiveness of our method.

Evaluation Capability of LLMs. The automated evaluation capability of LLMs has recently become a prominent point of research (Hackl et al., 2023; Hada et al., 2023; Zhu et al., 2023). Liu et al. (2023d) and Wang et al. (2023a) discover that LLMs have the potential to produce evaluation results consistent with human experts. Chiang and yi Lee (2023a) and Shen et al. (2023) further underscores the stability and reliability of assessments generated by LLMs. Kocmi and Federmann (2023) and Liu et al. (2023c) conduct a comparative study between LLM-based evaluation methods and existing automated evaluation metrics. Their results showcase that evaluations derived from LLMs surpassed all current automated benchmarks, indicating the exceptional evaluation capabilities of LLMs. Moreover, the utilization of LLMs for assessment offers several advantages including customizability (Fu et al., 2023a), a diversity of evaluation perspectives (Chen et al., 2023), and training-free (Luo et al., 2023). Given the remarkable evaluation prowess of LLMs (Chan et al., 2023; Chiang and yi Lee, 2023b; Gao et al., 2023), we integrate this capability into the aggregation of reasoning chains, enabling a more accurate assessment and selection of the reasoning processes and answers.

3. Preliminary

In this section, we provide definitions for standard prompting and CoT Prompting. Additionally, we detail the voting procedure of Self-Consistency. These foundational concepts serve as a groundwork for *AoR*. Considering a scenario where there is a question, denoted as $\mathcal{Q}$, along with a prompt, denoted as $\mathcal{T}$, and a LLM, denoted as $P_{\mathcal{M}}$.

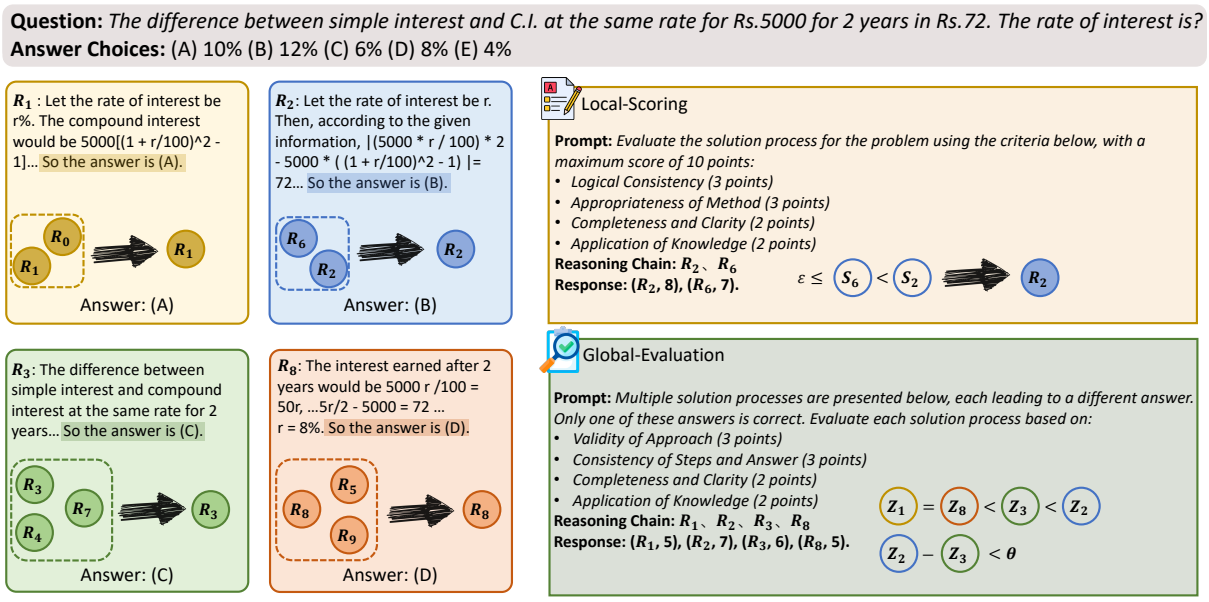


Figure 3: An illustrative example detailing the *AoR* workflow. Initially, 10 reasoning chains are sampled. During the local-scoring phase, reasoning chains with identical answers are compared, filtering out high-quality chains $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_8$ for global evaluation. In the global-evaluation phase, $\mathcal{R}_2$ receives the highest score, but the score margin between $\mathcal{R}_2$ and $\mathcal{R}_3$ fails to surpass the threshold θ.

Standard Prompting. Under standard prompting, LLM takes the question $\mathcal{Q}$ and the prompt $\mathcal{T}$ as inputs. It then sequentially generates each token of the answer $\mathcal{A}$, aiming to maximize the likelihood at each step.

$$P(\mathcal{A} \mid \mathcal{T}, \mathcal{Q}) = \prod_{i=1}^{|\mathcal{A}|} P_{\mathcal{M}}(a_i \mid \mathcal{T}, \mathcal{Q}, a_{<i}) \quad (1)$$

CoT Prompting. CoT (Wei et al., 2022b) enhances the prompt $\mathcal{T}$ by integrating the problem-solving process and guiding the LLM to generate a rationale $\mathcal{R}$ before generating the answer $\mathcal{A}$. We refer to the pair $(\mathcal{R}, \mathcal{A})$ as a reasoning chain.

$$P(\mathcal{R}, \mathcal{A} \mid \mathcal{T}, \mathcal{Q}) = P(\mathcal{A} \mid \mathcal{T}, \mathcal{Q}, \mathcal{R}) P(\mathcal{R} \mid \mathcal{T}, \mathcal{Q}), \quad (2)$$

where $P(\mathcal{R} \mid \mathcal{T}, \mathcal{Q})$ and $P(\mathcal{A} \mid \mathcal{T}, \mathcal{Q}, \mathcal{R})$ are defined as follows:

$$P(\mathcal{R} \mid \mathcal{T}, \mathcal{Q}) = \prod_{i=1}^{|\mathcal{R}|} P_{\mathcal{M}}(r_i \mid \mathcal{T}, \mathcal{Q}, r_{<i}) \quad (3)$$

$$P(\mathcal{A} | \mathcal{T}, \mathcal{Q}, \mathcal{R}) = \prod_{j=1}^{|\mathcal{A}|} P_{\mathcal{M}}(a_i \mid \mathcal{T}, \mathcal{Q}, \mathcal{R}, a_{<j}) \quad (4)$$

Self-Consistency. Self-Consistency (Wang et al., 2023d) employs CoT to sample n reasoning chains: $\{(\mathcal{R}_1, \mathcal{A}_1), (\mathcal{R}_2, \mathcal{A}_2), \ldots, (\mathcal{R}_n, \mathcal{A}_n)\}$. We define the set of answers as $\{\mathcal{A}\} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$. The final answer $\mathcal{A}^*$ is determined by selecting the answer that appears most frequently within $\{\mathcal{A}\}$.

$$\mathcal{A}^* = \arg\max_a |\{(\mathcal{R}_i, \mathcal{A}_i) \mid \mathcal{A}_i = a\}| \quad (5)$$

4. Methodology

4.1. Overview

The *AoR* approach to aggregating reasoning primarily unfolds in two stages: local-scoring and global-evaluation. Firstly, we utilize CoT to sample n reasoning chains, represented as $\{(\mathcal{R}_1, \mathcal{A}_1), (\mathcal{R}_2, \mathcal{A}_2), \ldots, (\mathcal{R}_n, \mathcal{A}_n)\}$. Supposing there are m unique answers generated, denoted as $\{a_1, a_2, \ldots, a_m\}$, we categorize them into m distinct buckets. The j^{th} bucket is defined as $\{(\mathcal{R}_i, \mathcal{A}_i) \mid \mathcal{A}_i = a_j\}$. In the local-scoring phase, we score the reasoning chains $(\mathcal{R}_i, \mathcal{A}_i)$ within each bucket. The top k chains, based on their scores, are selected as representatives for the bucket. In the global-evaluation phase, a representative is selected from each of the buckets for assessment. After k rounds of evaluations, the bucket with the highest average score determines the final output. Figure 3 provides an illustrative example. Although incorrect answers (C) and (D) are in the majority, the two-phase process of local-scoring and global-evaluation accurately discerns and attributes the highest score to the correct answer (B).

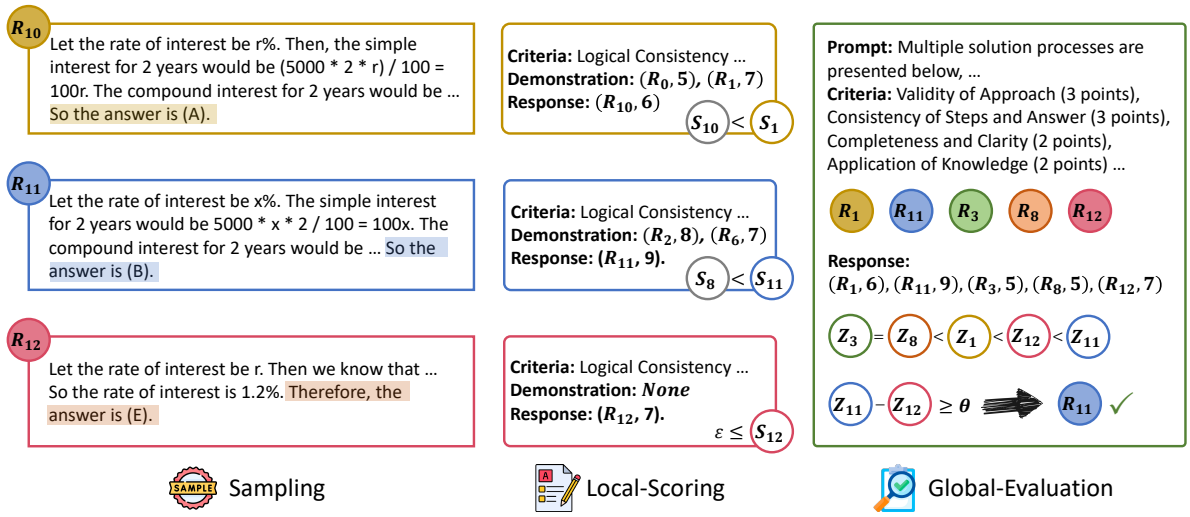


Figure 4: Illustration of the dynamic sampling process, where solid circles represent reasoning chains and hollow circles their respective scores. Due to the minimal score difference between $\mathcal{R}_2$ and $\mathcal{R}_3$, three additional chains $\mathcal{R}_{10}$, $\mathcal{R}_{11}$, and $\mathcal{R}_{12}$ are sampled, yielding answers (A), (B), and (E). $\mathcal{R}_{10}$ and $\mathcal{R}_{11}$ are compared against chains with matching answers. $\mathcal{R}_{10}$ fails to outscore $\mathcal{R}_1$, while $\mathcal{R}_{11}$ surpasses $\mathcal{R}_8$, advancing to global evaluation. $\mathcal{R}_{12}$, introducing a new answer (E), exceeds the threshold ϵ and progresses. In the global evaluation, $\mathcal{R}_{11}$ outperforms others, and with its score difference with $\mathcal{R}_{12}$ exceeding θ, thus answer (B) is selected as the final decision.

Local-Scoring. Local-scoring focuses on selecting high-quality reasoning chains within a group sharing the same answer. While fixing the answer, the evaluation can place a heightened emphasis on the rigor of the rationale logic and the appropriateness of the reasoning steps. Let's assume there are n_j reasoning chains leading to the answer a_j, denoted as $(\mathcal{R}_1^{(j)}, \mathcal{A}_1^{(j)}), \ldots, (\mathcal{R}_{n_j}^{(j)}, \mathcal{A}_{n_j}^{(j)})$, collectively forming bucket j.

When these n_j items are input into the LLM simultaneously, guided by evaluation criteria in the prompt $\mathcal{T}_1$, the LLM assigns a score $\mathcal{S}_i^{(j)}$ to each $\mathcal{R}_i^{(j)}$. Based on a predefined threshold ϵ, high-quality chains are identified as $\{(\mathcal{R}_i^{(j)}, \mathcal{A}_i^{(j)}) \mid \mathcal{S}_i^{(j)} \geq \epsilon\}$. From this refined set, the top k items are selected as representatives of bucket j, denoted as $\mathcal{B}_{topk}^{(j)}$. If no item satisfies $\mathcal{S}_i^{(j)} \geq \epsilon$, then $\mathcal{B}_{topk}^{(j)}$ is an empty set, and items from this bucket will be excluded from the global-evaluation phase.

Global-Evaluation. Global-evaluation is tasked with distinguishing and selecting the reasoning chain among different answers, aiming to pinpoint the one that demonstrates optimal coherence and consistency between the reasoning process and its outcome. Assuming that we have m buckets. A representative is chosen from each bucket, forming a set $\bigcup_{j=1}^{m} \{b^{(j)} | b^{(j)} \in \mathcal{B}_{topk}^{(j)}\}$. When these m representatives are fed into the LLM concurrently, guided by evaluation criteria encapsulated in prompt $\mathcal{T}_2$, the LLM assigns a score $Z^{(j)}$ to each $b^{(j)} = (R^{(j)}, A^{(j)})$.

Representatives from each bucket are sequentially chosen for k rounds of scoring. Ultimately, the bucket j^* with the highest average score determines the final answer. If the number of representatives in a bucket is less than k, previously selected items are resampled to meet the required count.

$$j^* = \arg\max_j \frac{1}{k} \sum_{t=1}^{k} \mathcal{Z}_t^{(j)} \tag{6}$$

It's worth noting that representatives from each bucket are high-quality reasoning chains bearing scores that are identical or closely aligned, each showcasing its unique advantages. Consequently, conducting multiple rounds of scoring not only mitigates the randomness in single-round evaluations but also ensures a comprehensive assessment.

4.2. Dynamic Sampling

Leveraging the scores from the global-evaluation phase, *AoR* dynamically adjusts the sampling of reasoning chains based on the LLM's confidence in the optimal reasoning chain. This process begins by identifying two key answers: $\mathcal{A}^{\alpha}$, which has the highest average score $\bar{\mathcal{Z}}^{\alpha}$, and $\mathcal{A}^{\beta}$, with the second-highest average score $\bar{\mathcal{Z}}^{\beta}$. Drawing inspiration from Roth and Small (2006), we consider the margin $\bar{\mathcal{Z}}^{\alpha} - \bar{\mathcal{Z}}^{\beta}$. If this margin exceeds a predefined threshold θ, it signifies a substantial quality discrepancy between the top two reasoning chains,

leading to the selection of $\mathcal{A}^{\alpha}$ as the final answer and terminating the sampling process.

If $\bar{\mathcal{Z}}^{\alpha} - \bar{\mathcal{Z}}^{\beta} < \theta$, AoR proceeds to sample an additional d reasoning chains. These new chains undergo evaluation against established benchmarks to calculate their scores $\{S_{n+1}, S_{n+2}, \ldots, S_{n+d}\}$. These scores determine their influence on the existing answer hierarchy. If S_{n+1} is either beneath the threshold θ or does not surpass the minimum score within the top k scores of its answer category, the sampled chain $(\mathcal{R}_{n+1}, \mathcal{A}_{n+1})$ does not affect the overall ranking. Conversely, if a sampled chain introduces a new answer $\mathcal{A}_{n+1} = a_{m+1}$ satisfied threshold θ or significantly alters the score ranking within $\mathcal{B}_{topk}$, a re-evaluation during the global-evaluation phase is necessitated to recalibrate scores.

Dynamic sampling ceases once the confidence margin between the two leading answers meets or exceeds θ or when the total number of sampled chains reaches a predefined maximum n_{max}. As illustrated in Figure 4, we present a straightforward instance of dynamic sampling, in which the accuracy of the final decision is enhanced by integrating an additional reasoning chain, confidently pinpointing answer B during the global evaluation phase. This flexible method guarantees a more efficient assessment, reducing unnecessary computational efforts on clear-cut cases and focusing more rigorously on analyzing queries that are complex or have ambiguous interpretations. By adjusting the depth of evaluation according to the estimated complexity of each task, *AoR* efficiently balances precision in its outcomes with optimal use of computational resources.

5. Experiment

5.1. Experimental Setup

Tasks and Datasets. We conduct a comprehensive evaluation of *AoR* across three types of reasoning tasks. (1) **Mathematical reasoning** incorporates six representative datasets, namely GSM8K (Cobbe et al., 2021), MultiArith (Roy and Roth, 2015), SingleEQ (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), AddSub (Hosseini et al., 2014), and AQuA (Ling et al., 2017). (2) **Commonsense reasoning** covers StrategyQA (Geva et al., 2021), CommonsenseQA (CSQA; Talmor et al., 2019), BoolQ (Clark et al., 2019), and AI2 Reasoning Challenge (ARC-C) (Clark et al., 2018). (3) **Symbolic reasoning** comprises four datasets derived from BigBench (bench authors, 2023; Suzgun et al., 2023), including Date Understanding, Penguins in a Table, Colored Objects, and Object Counting. A comprehensive overview and statistical analysis of the dataset is presented in Appendix A.1.

Baselines. We compare *AoR* with several strong baselines detailed in Section 2. These include Chain-of-Thought prompting (CoT; Wei et al., 2022b), Complexity-based prompting (Complex-CoT; Fu et al., 2023b), Self-Consistency (SC; Wang et al., 2023d), Complexity-based Consistency (CC; Fu et al., 2023b), Progressive-Hint Prompting (PHP; Zheng et al., 2023) and DIVERSE (Li et al., 2023b).

In our experiments, we adhere to the settings of Self-Consistency (Wang et al., 2023d) and sampled 40 reasoning chains, denoted as (40). Regarding notations, CoT and ComplexCoT represent prompt exemplars with different reasoning complexities, while SC, CC, PHP, and DIVERSE signify various methods of reasoning chain ensemble. For instance, the notation CoT-SC(40) signifies that 40 reasoning chains are generated using CoT prompts, followed by the application of the Self-Consistency method. For all baselines, we follow their official implementations for fair comparison.

Backbone LLMs. In the main experiments, we employ `GPT-3.5-Turbo-0301`. In the discussion part, we introduce a broader variety of models, including `GPT-4-0314`, `Claude-2`, and the open-source model `LLaMA-2-70B-Chat` and `Mixtral-8x7B`. We access models from OpenAI and Anthropic using their official APIs, while for `LLaMA-2-70B-Chat` and `Mixtral-8x7B`, we utilized model weights and code provided by Touvron et al. (2023b) and Jiang et al. (2024).

When sampling various reasoning chains, we configure the temperature setting differently across models to optimize their performance. Specifically, for `GPT-3.5-Turbo`, `GPT-4`, and `Claude-2`, we maintain a temperature of 1. For LLaMA, we adhere to its official recommendation by setting the temperature at 0.6, and for `Mistral`, we opt for a temperature of 0.7 to achieve optimal performance.

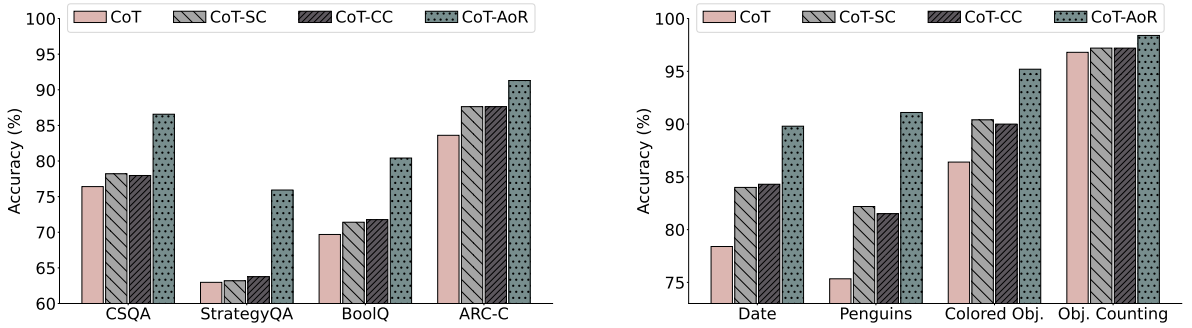
By default, *AoR* initially samples 20 reasoning chains, implementing a dynamic sampling strategy with an upper limit of $n_{max} = 40$ and a batch size $b = 5$, collectively referred to as AoR(20,40). During the local scoring phase, we define a representative count of $k = 3$ and a scoring threshold of $\epsilon = 6$. For dynamic sampling, we establish a termination criterion with a threshold of $\theta = 2$, and with each iteration, we sample an additional 5 reasoning chains. Moreover, we utilize the best and worst reasoning chains within the same answer as evaluation benchmarks to evaluate the newly sampled reasoning chains. Our implementation details and hyperparameters analysis are available in Appendix A.2 and A.3.

5.2. Main Results

Mathematical Reasoning. The results for the mathematical reasoning tasks are presented in Ta-

	GSM8K	MultiArith	SingleEQ	SVAMP	AddSub	AQuA	Avg
CoT	80.0	97.7	91.9	78.1	86.6	54.7	81.50
CoT-PHP	84.6	98.3	93.9	83.9	86.1	65.4	85.37
CoT-SC(40)	88.9	99.3	94.5	85.9	87.6	68.7	87.48
CoT-CC(40)	88.7	99.2	94.3	86.1	87.8	69.3	87.57
CoT-Diverse(40)	89.2	99.3	94.5	86.6	88.7	70.9	88.20
CoT-*AoR* (20, 40)	**91.8**	**99.8**	**95.5**	**89.8**	**90.6**	**75.9**	**90.57**
ComplexCoT	82.8	97.5	92.5	81.0	85.5	57.4	82.78
ComplexCoT-PHP	85.1	98.0	92.9	83.1	85.3	60.6	84.16
ComplexCoT-SC(40)	90.6	98.5	94.9	87.5	87.5	70.5	88.25
ComplexCoT-CC(40)	90.5	98.3	93.3	87.2	87.5	70.0	87.80
ComplexCoT-DiVeRSe(40)	90.8	98.7	94.3	87.8	88.2	72.9	88.78
ComplexCoT-*AoR* (20, 40)	**92.9**	**99.5**	**95.3**	**91.0**	**89.1**	**76.4**	**90.70**

Table 2: Comparison of performance (accuracy %) between *AoR* and several strong baselines across six mathematical reasoning datasets. The highest accuracy scores are underlined. Within the same prompt, standout results are highlighted in **bold**. All methods employ a `GPT-3.5-Turbo-0301` backbone for a fair comparison. Results for ComplexCoT and ComplexCoT-PHP are sourced from Zheng et al. (2023). The average performance across datasets is provided for an overall comparison.



(a) Commonsense Reasoning Tasks.

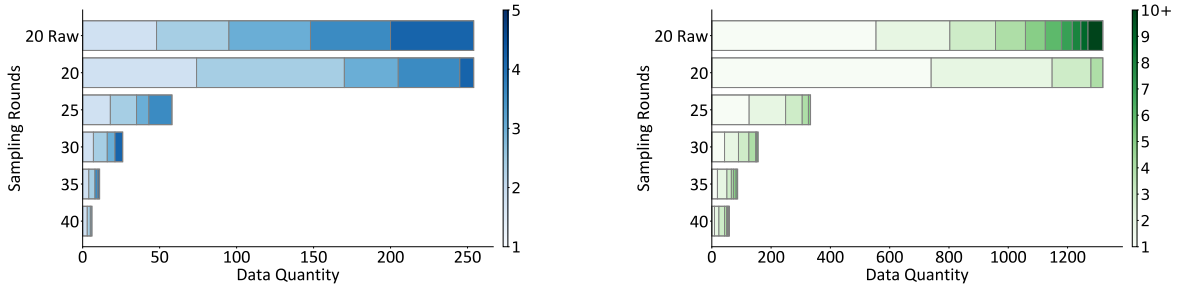
(b) Symbolic Reasoning Tasks.

Figure 5: Performance comparison of AoR and various strong baselines on commonsense reasoning and symbolic reasoning tasks.

ble 2. Across six datasets, *AoR* surpasses all baseline approaches. Under the CoT prompt, when compared to the competitive DiVeRSe method, *AoR* achieves an average performance boost of 2.37% across six datasets. Furthermore, the average performance shows an improvement of 3.09% compared to the SC method, with a significant increase of 7.2% on the AQuA dataset. When employing ComplexCoT prompt, *AoR* maintains its competitive advantage. It shows average performance enhancements of 2.45%, 2.90%, and 1.92% compared to the SC, CC, and DiVeRSe method.

Commonsense and Symbolic Reasoning. Figures 5a and 5b illustrate the performance of *AoR* in commonsense reasoning and symbolic reasoning tasks. For commonsense reasoning tasks, *AoR* demonstrate an average performance improvement of 8.45% and 8.27% compared to SC and CC methods. Notably, on StrategyQA, which emphasizes implicit reasoning strategies, both SC and CC do not significantly outperform the baseline CoT methods. In contrast, *AoR* effectively enhances the LLM's performance on StrategyQA. Moreover, *AoR* consistently achieves significant performance improvements in symbolic reasoning tasks. When compared to the SC method, there are improvements of 5.8% and 8.9% on the Date Understanding and Penguins datasets.

Dynamic Sampling. Figures 6a and 6b illustrate the progression of sample counts during the dynamic sampling process on the AQuA and GSM8K datasets. The color scheme represents the variance in answer counts within the dataset, transitioning from light to dark shades to illustrate the range from singular to multiple answer occurrences. The majority of samples conclude satisfactorily after the first round, with only a select group of more complex samples necessitating further reasoning chains. With each subsequent round of sampling, there's a noticeable decline in the total number of samples, indicating that the newly added reasoning chains contribute to the final answer's determina-



(a) AQuA. (b) GSM8K.

Figure 6: Correlation of sample volume to dynamic sampling iterations in AQuA and GSM8K datasets. The x-axis represents the sample count, while the y-axis indicates the rounds of sampling. Color variations denote the range of answers identified in the global-evaluation phase across different data samples, with "20 Raw" indicating the initial distribution of answer counts.

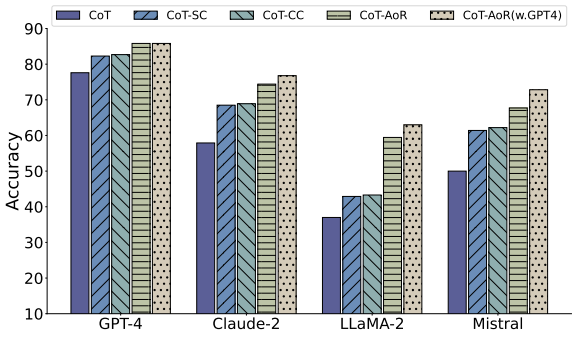


Figure 7: Performance of *AoR* using different LLMs for both backbones and evaluators when solving AQuA problems.

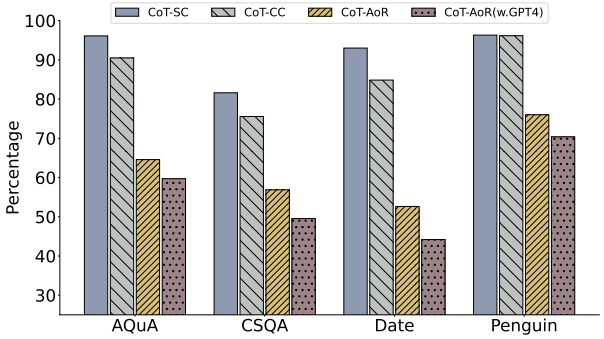


Figure 8: Proportion of samples lead to incorrect final prediction that contain at least one correct candidate answer.

tion. Compared to the AQuA dataset, which uses options as answers, the open-ended nature of the GSM8K dataset results in a broader distribution of initial answers. By observing the distribution of answers in the "20Raw" and "20" phases, it is evident that in the local-scoring phase, after filtering out a substantial number of low-quality reasoning chains, significantly reduces the number of candidate answers, enabling a more accurate final answer selection in the global evaluation.

5.3. Discussion

In this section, we delve into the advantages of the *AoR* method, dissecting the reasons behind its performance improvements from four perspectives.

***AoR* on Various LLMs.** Figure 9 depicts the enhanced performance of *AoR* when applied to four different LLMs. In comparison with SC and CC, *AoR* achieves an average improvement of 8.1% and 7.6%. Our evaluation extends to two prominent open-source models: the dense model `LLaMA-2-70B-Chat` and the Mixture-of-Experts (MoE) model `Mixtral-8x7B`. *AoR* achieves consistent improvements across various LLM architectures. Notably, with the `LLaMA-2` model, the improvement is notably significant, attaining a 16.6% increase compared to SC. Moreover, we conduct an analysis of the evaluation models and obverse that integrating `GPT-4` into the local-scoring and global-evaluation phases results in performance improvements of 2.4%, 3.6% and 5.1% on the `Claude-2`, and `LLaMA-2`, and `Mistral` models. This highlights the potential for a superior evaluation model to enhance the effectiveness of *AoR*.

Analysis of Incorrect Samples. In Section 1, we analyze the erroneous samples from the SC method, revealing that a majority of these samples did not arise from LLM's inability to produce the correct answer. Instead, the majority voting mechanism failed to identify the right answer. Adopting a similar analytical approach for *AoR*'s incorrect samples, as depicted in Figure 8, we discover a significant reduction in the proportion of samples where *AoR* failed to select the correct answer. This underscores *AoR*'s efficiency in leveraging the information from reasoning chains to boost the likelihood of selecting the correct answer. Moreover, this proportion can be further reduced by employing a more discerning evaluator.

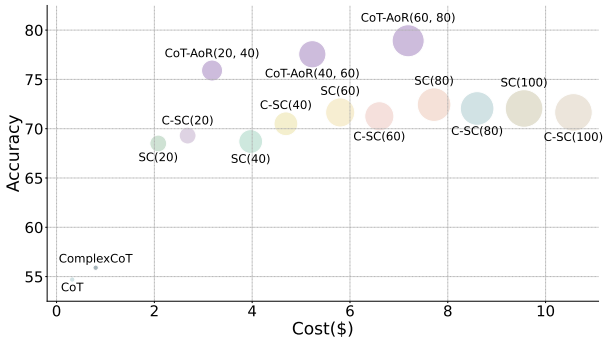


Figure 9: Analysis of cost and performance. The x-axis represents the cost, the y-axis indicates accuracy, and the size of each point corresponds to the number of reasoning chains. For brevity, we use "SC" to represent "CoT-SC", and "C-SC" to denote "Complexity-SC."

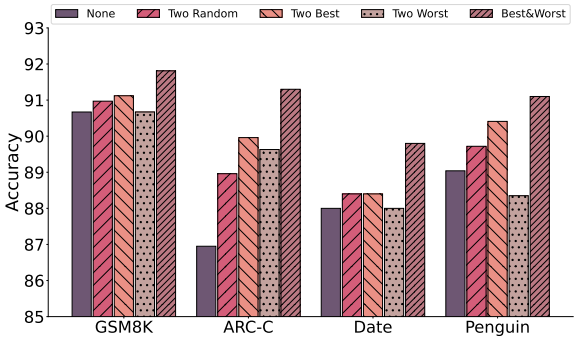


Figure 10: Evaluation of demonstration selections during the local-scoring phase of dynamic sampling. "Best" and "Worst" denote the reasoning chains with the highest and lowest scores, respectively, among those yielding identical answers.

Cost and Performance Analysis. A potential concern revolves around the additional overhead introduced by *AoR*'s evaluation and whether dynamic sampling can effectively reduce reasoning costs. In Figure 9, we analyze the cost and performance of *AoR* and SC on the AQuA dataset using `GPT-3.5`. Notably, CoT-*AoR* (20,40) not only surpasses CoT-SC(40) with a 7.2% boost in performance but also achieves a significant 20% reduction in overhead. Furthermore, CoT-*AoR* (20, 40) outperforms even CoT-SC(100), indicating that compared to majority voting, *AoR*'s evaluation of reasoning chains is a more efficient method for answer selection. It's noteworthy that the SC method exhibits saturation: there is no significant performance improvement when the number of reasoning chains exceeds 60. In contrast, *AoR* continues to show noticeable performance enhancements at sampling chains of 40 and 60. This suggests that AoR possesses a superior performance ceiling in comparison to SC approaches, underlining its cost-effectiveness and higher potential for accuracy improvement.

Analysis of Evaluation Benchmarks. In the local-scoring phase of dynamic sampling, evaluated reasoning chains are leveraged to score newly added chains. Figure 10 assesses the impact of using no demonstrations versus various demonstration strategies on the final answer across different datasets. Strategies include selecting no reasoning chains, two random chains, the two highest, the two lowest, and a combination of the highest and lowest scoring chains for demonstration. While this primarily affects dynamically sampled cases, demonstrations consistently enhance model performance across datasets. This improvement is likely because demonstrations provide the model with insight into the current score distribution, enabling more informed scoring. Notably, employing the highest and lowest scoring chains as demonstrations achieves the best performance, likely because they offer a comprehensive view of the score range, aiding the model in more accurately scoring new chains. However, using the two lowest scoring chains as examples tends to bias the model towards lower scores, often preventing these new chains from advancing to the global evaluation phase and thus impairing performance. Consequently, we utilize both the best and worst reasoning chains within the same answer as evaluation benchmarks mentioned in Section 4.2.

6. Conclusion

In this study, we introduce *AoR* (Aggregation of Reasoning), a pioneering framework that enhances the ensemble methods for reasoning chains by meticulously evaluating and aggregating the reasoning processes. *AoR* employs a two-phase evaluation approach, assessing reasoning chains from multiple perspectives, ensuring the evaluation's validity and comprehensiveness. Notably, *AoR* allows for the dynamic adjustment of the number of reasoning chains according to the complexity of the task, substantially minimizing unnecessary computational overhead. Experimental results illustrate that *AoR* significantly improves the reasoning abilities of LLMs, outperforming several established baselines. Furthermore, our in-depth analysis indicates that *AoR*'s adaptability extends across various LLM architectures, with potential for further enhancements through integrating a more robust evaluator and an increased volume of reasoning chains. Compared to the existing ensemble methods, *AoR* not only presents benefits in terms of performance and efficiency but also effectively mitigates the risk of accurate answers being overshadowed by more frequent but incorrect predictions.

Ethical Statement

In developing the *AoR* framework, our team has prioritized ethical considerations to ensure our work respects privacy and promotes fairness. Specifically, the *AoR* methodology does not involve the collection or utilization of any personally identifiable information. The design of our experimental prompts has been meticulously crafted to prevent any form of discrimination against individuals or groups, thereby safeguarding against privacy breaches and potential socio-ethical implications. Furthermore, we have conducted an in-depth review of the licenses for all datasets employed in our research, as outlined in Appendix A.1.

Limitations

Manual Demonstration Construction for Local-Scoring and Global-Evaluation. Our approach relies on manually crafted demonstrations to guide the model in generating outputs in the desired format for extracting scores. This method's efficacy is contingent on the model's ability to accurately interpret these demonstrations and produce outputs as anticipated. In instances where the model fails to comprehend the demonstrations adequately or deviates from the expected output format, the performance of *AoR* becomes unstable, potentially hindering the completion of its process. Nonetheless, we are optimistic that the evolution of LLMs will bolster their comprehension (Cheng et al., 2024; Naveed et al., 2024) and output formatting capabilities (Liang et al., 2024; Dekoninck et al., 2024), thereby mitigating this issue over time.

Model Context Window Size Limitations. The limitations imposed by the model's context window size restrict the number of examples that can be processed simultaneously. At present, models face challenges in handling an extensive array of reasoning chains, necessitating a balance between performance assessment and computational expenditure. While smaller parameter models can navigate through the AoR process, their ability is often limited to evaluating single reasoning chains, thereby escalating the computational demands of AoR. However, we believe this to be a temporary constraint. Recent models like `Mistral` (Jiang et al., 2023) and `InternLM` (Team, 2023) have demonstrated evaluation capacities comparable to those of `GPT` with appropriate prompting. Moreover, we are encouraged by recent advancements that have significantly expanded the models' context windows (Xiao et al., 2023; Liu et al., 2024). As long-context models continue to evolve (Ratner et al., 2023; Wang et al., 2023c), we anticipate that AoR will be able to conduct evaluations on larger batches of reasoning chains, substantially reducing computational costs and enhancing efficiency.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62236004). We are grateful to the reviewers for their insightful comments and suggestions, which have significantly improved the quality of this manuscript.

Bibliographical References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of thoughts: Solving elaborate problems with large language models.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate.

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

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: An empirical study.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension.

Cheng-Han Chiang and Hung yi Lee. 2023a. Can large language models be an alternative to human evaluations?

Cheng-Han Chiang and Hung yi Lee. 2023b. A closer look into automatic evaluation using large language models.

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

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. 2024. Controlled text generation via language model arithmetic.

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

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023a. Gptscore: Evaluate as you desire.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023b. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Andrew Gao. 2023. Prompt engineering for large language models. *Available at SSRN 4504303*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt.

Veronika Hackl, Alexandra Elena Müller, Michael Granitzer, and Maximilian Sailer. 2023. Is gpt-4 a reliable rater? evaluating consistency in gpt-4 text ratings.

Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2023. Are large language model-based evaluators the solution to scaling up multilingual evaluation?

Chengcheng Han, Xiaowei Du, Che Zhang, Yixin Lian, Xiang Li, Ming Gao, and Baoyuan Wang. 2023. DialCoT meets PPO: Decomposing and exploring reasoning paths in smaller language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8055–8068, Singapore. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023a. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. MoT: Memory-of-thought enables ChatGPT to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374, Singapore. Association for Computational Linguistics.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga,

Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. 2024. Controlled text generation for large language model with dynamic attribute graphs.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts.

Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023b. Plan, verify and switch: Integrated reasoning with diverse X-of-thoughts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2822, Singapore. Association for Computational Linguistics.

Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. 2024. Scaling laws of rope-based extrapolation.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023c. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023d. Calibrating llm-based evaluator.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

OpenAI. 2023. GPT-4 technical report.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2022. Scaling language models: Methods, analysis & insights from training gopher.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Dan Roth and Kevin Small. 2006. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 413–424. Springer.

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

Chenhui Shen, Liying Cheng, Yang You, and Lidong Bing. 2023. Are large language models good evaluators for abstractive summarization?

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM.

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

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

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is chatgpt a good nlg evaluator? a preliminary study.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2023b. Boosting language models reasoning with chain-of-knowledge prompting.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023c. Augmenting language models with long-term memory.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pages 41618–41650. Curran Associates, Inc.

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

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023a. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153, Singapore. Association for Computational Linguistics.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023b. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. 2023. Towards better chain-of-thought prompting strategies: A survey.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

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

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can chatgpt reproduce human-generated labels? a study of social computing tasks.

Language Resource References

BIG-bench authors. 2023. *Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models*.

Clark, Christopher and Lee, Kenton and Chang, Ming-Wei and Kwiatkowski, Tom and Collins, Michael and Toutanova, Kristina. 2019. *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions*. Association for Computational Linguistics.

Peter Clark and Isaac Cowhey and Oren Etzioni and Tushar Khot and Ashish Sabharwal and Carissa Schoenick and Oyvind Tafjord. 2018. *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge*.

Karl Cobbe and Vineet Kosaraju and Mohammad Bavarian and Mark Chen and Heewoo Jun and Lukasz Kaiser and Matthias Plappert and Jerry Tworek and Jacob Hilton and Reiichiro Nakano and Christopher Hesse and John Schulman. 2021. *Training Verifiers to Solve Math Word Problems*.

Geva, Mor and Khashabi, Daniel and Segal, Elad and Khot, Tushar and Roth, Dan and Berant, Jonathan. 2021. *Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies*. MIT Press.

Hosseini, Mohammad Javad and Hajishirzi, Hannaneh and Etzioni, Oren and Kushman, Nate. 2014. *Learning to Solve Arithmetic Word Problems with Verb Categorization*. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Wang Ling and Dani Yogatama and Chris Dyer and Phil Blunsom. 2017. *Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems*. Association for Computational Linguistics.

Patel, Arkil and Bhattamishra, Satwik and Goyal, Navin. 2021. *Are NLP Models really able to Solve Simple Math Word Problems?* Association for Computational Linguistics.

Subhro Roy and Dan Roth. 2015. *Solving General Arithmetic Word Problems*. The Association for Computational Linguistics.

Suzgun, Mirac and Scales, Nathan and Schärli, Nathanael and Gehrmann, Sebastian and Tay, Yi and Chung, Hyung Won and Chowdhery, Aakanksha and Le, Quoc and Chi, Ed and Zhou, Denny and Wei, Jason. 2023. *Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them*. Association for Computational Linguistics.

Talmor, Alon and Herzig, Jonathan and Lourie, Nicholas and Berant, Jonathan. 2019. *CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge*. Association for Computational Linguistics.

A. Appendices

A.1. Dataset Statistics

In our experiment, we meticulously select 14 datasets encompassing mathematical reasoning, commonsense reasoning, and symbolic reasoning domains. The specifics and statistical details of each dataset, including the data source, task type, answer type, number of prompt samples, total test samples, and dataset licenses, are comprehensively outlined in Table 3.

A.2. Implementation Details

Prompting Exemplars. AoR utilizes the Wei et al. (2022b) and Fu et al. (2023b) provided prompt exemplars to sample reasoning chains, with the number of prompt exemplars for each dataset detailed in Table 3. In the local scoring phase, given that the answers are identical, our evaluation focuses more on the soundness of the reasoning process and the correctness of the reasoning method. Specifically, we require the LLM to evaluate reasoning chains that share the same answer from four perspectives as follows:

- **Logical Consistency (3 points):** The coherence and soundness of the reasoning are evaluated to ensure logical progression.
- **Appropriateness of Method (3 points):** The suitability of the used method is verified, emphasizing that the approach is not unnecessarily complex.
- **Completeness and Clarity (2 points):** All necessary steps must be clearly shown without omission, ensuring easy follow-through.
- **Application of Knowledge (2 points):** The correct and relevant application of formulas, theorems, or facts is assessed.

The global-evaluation phase prioritizes the correctness of the method and the consistency between reasoning steps and the answer, enabling the model to filter out the correct reasoning chain from those with differing answers. Specifically, we require the LLM to evaluate reasoning chains with different answers from the following four perspectives:

- **Validity of Approach (3 points):** The employed method effectively addresses the problem, confirming the appropriateness of the approach.
- **Consistency of Steps and Answer (3 points):** It is ensured that all steps are not only correct but also consistent with the final answer.
- **Completeness and Clarity (2 points):** Essential steps are delineated and presented unambiguously, maintaining clarity throughout.
- **Application of Knowledge (2 points):** The precision and appropriateness in the use of formulas, theorems, or facts are verified.

In line with the findings of Gao (2023), providing as much detailed information as possible in the input facilitates the generation of the desired outcome. Thus, additional statistical information, such as the number of reasoning chains within a bucket and the number of candidate answers, is incorporated into the prompt. For the complete prompt, please refer to our Github repository.

Evaluation. We employ accuracy as the metric to assess performance across tasks involving mathematical reasoning, commonsense reasoning, and symbolic reasoning. For datasets where the answer is numerical, such as GSM8K, we utilize regular expressions to extract the answer following the phrase "the answer is" and conduct a numerical comparison with the provided answer. For datasets where the answers are choices, such as AQuA, we compare the extracted choice with the correct option to verify consistency. In cases where the dataset answers are binary (yes/no), such as StrategyQA, we evaluate whether the extracted result aligns with the provided label. If a reasoning chain fails to correctly extract an answer, it is excluded from further consideration. Similar to the approach by Xie et al. (2023), we fine-tune task-specific verifiers to assign weights to the sampled reasoning chains to implement the DIVERSE (Li et al., 2023b).

Computation Cost. Computational costs are quantified based on OpenAI's official pricing for the `GPT-3.5-Turbo-0301` API, calculated as follows: Input Tokens $\times\ 0.0015/1000$ + Output Tokens $\times\ 0.002/1000$.

Our primary experiments, as outlined in Section 5.2 were conducted from July to September 2023. Discussion in Section 5.3 and the Ablation Study in Appendix A.3 for both commercial and open-source models were completed between October and December 2023.

Due to rate limits and budget constraints, we set an upper limit on our sample size for each analysis. Consequently, our analysis is based on a maximum of 500 samples per run.

A.3. Ablation Study

To facilitate the intricate reasoning chain aggregation process in AoR, we establish essential hyperparameters during the local scoring and global evaluation phases, such as the representative count

Dataset	Reasoning Task	Answer Type	# Prompts	# Test	License
GSM8K (Cobbe et al., 2021)	Arithmetic	Number	8	1,319	MIT License
MultiArith (Roy and Roth, 2015)	Arithmetic	Number	8	600	Unspecified
SingleEQ (Koncel-Kedziorski et al., 2016)	Arithmetic	Number	8	508	Unspecified
AddSub (Hosseini et al., 2014)	Arithmetic	Number	8	395	Unspecified
SVAMP (Patel et al., 2021)	Arithmetic	Number	8	1,000	MIT License
AQUA (Ling et al., 2017)	Arithmetic	Multi-choice	4	254	Apache-2.0
StrategyQA (Geva et al., 2021)	Commonsense	T/F	6	2,290	MIT license
CommonsenseQA Talmor et al., 2019	Commonsense	Multi-choice	7	1,221	Unspecified
BoolQ (Clark et al., 2019)	Commonsense	T/F	4	3,270	CC BY-SA 3.0
ARC-C (Clark et al., 2018)	Commonsense	Multi-choice	4	299	CC BY-SA 4.0
Date Understanding (Suzgun et al., 2023)	Symbolic	Multi-choice	3	250	MIT license
Penguins in a Table (Suzgun et al., 2023)	Symbolic	Multi-choice	3	146	MIT license
Colored Objects (Suzgun et al., 2023)	Symbolic	Multi-choice	3	250	MIT license
Object Counting (Suzgun et al., 2023)	Symbolic	Multi-choice	3	250	MIT license

Table 3: Overview of datasets utilized in our experiments. # Prompts indicates the number of Chain-of-Thought (CoT) (Wei et al., 2022b) prompting exemplars used for few-shot prompting. # Test denotes the total count of test samples in each dataset.

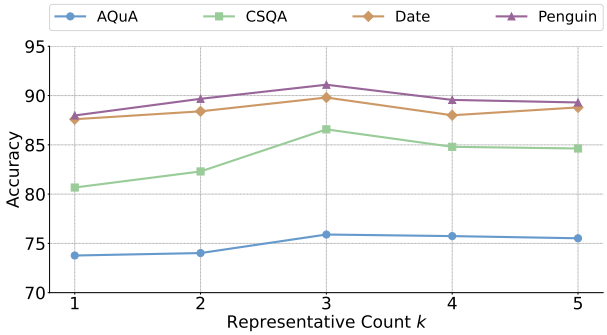


Figure 11: Ablation on representative count k on various reasoning datasets.

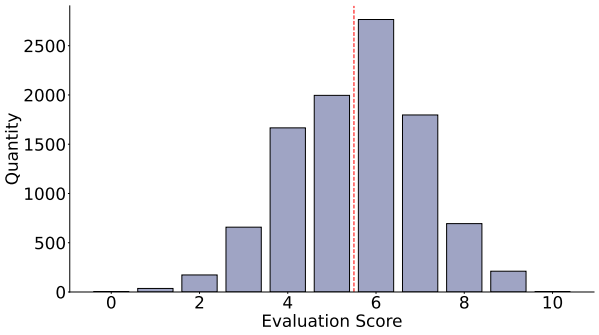


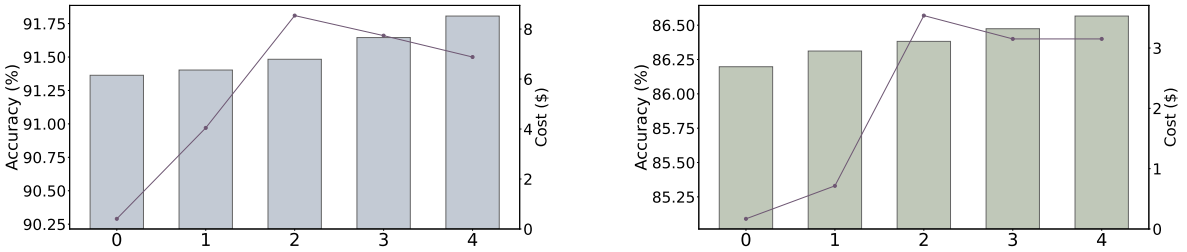
Figure 12: Distribution of evaluation scores in the local-scoring phase on the GSM8K dataset.

k, score threshold ϵ, termination threshold θ, and batch size b. Below, we conduct ablation experiments using the `GPT-3.5` model to examine the impact of each hyperparameter on the overall performance.

Analysis of Representative Count k. We analyze four datasets to investigate how the representative count k impacts accuracy, as shown in Figure 11. When $k = 1$, only the highest-scoring reasoning chain from each bucket is evaluated, which puts rigorous demands on the scoring model and can result in fluctuating outcomes. Selecting more representatives from each bucket enhances the comprehensiveness and stability of the evaluation, harmonizing the quality and diversity of reasoning chains. However, our findings suggest that increasing the number of representatives when $k > 3$ does not lead to significant performance gains but does incur additional computational overhead. As a result, we chose $k = 3$ as it strikes an optimal balance between performance and computational cost.

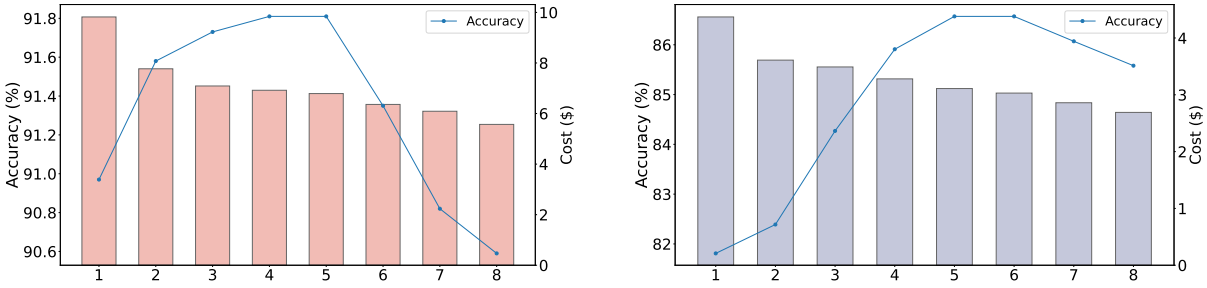
Analysis of Score Threshold ϵ. Figure 12 illustrates the score distribution during the local-scoring phase on the GSM8K dataset, where we observe a normal distribution of scores. The model seldom assigns very low scores (0-2 points). A lower score threshold ϵ leads to an excessive number of reasoning chains proceeding to global evaluation; for instance, setting ϵ to 3 results in over 95% of reasoning chains moving to global evaluation. Conversely, a higher ϵ enforces stricter filtering; setting ϵ to 8 results in fewer than 10% of reasoning chains moving forward to global evaluation, leading to many samples having only one reasoning chain in the global-evaluation phase. Some samples might even finish dynamic sampling without any reasoning chains proceeding to global evaluation. Therefore, we determine the score threshold ϵ to be 6, which ensures a balance by maintaining high-quality reasoning chains and allowing a sufficient number to undergo global evaluation.

Analysis of Termination Threshold θ. Figure 13 demonstrates the impact of various termination thresholds (θ) on accuracy and computational cost. A threshold of $\theta = 0$ implies that we select the answer associated with the highest-scoring reasoning chain as the final answer without sampling



(a) GSM8K. (b) CSQA.

Figure 13: The effect of varying termination thresholds θ on accuracy and computational cost for the GSM8K and CSQA datasets. Line graphs illustrate accuracy, while bar graphs depict computational costs.



(a) GSM8K. (b) CSQA.

Figure 14: The impact of batch size b variations on accuracy and computational cost on the GSM8K and CSQA datasets. Line graphs represent accuracy, while bar charts indicate computational costs.

additional reasoning chains. While this approach incurs lower costs, it results in the poorest performance on both the GSM8K and AQuA datasets. This suggests that relying solely on the model's confidence in the highest-scoring reasoning chain does not guarantee its correctness. As the threshold increases, we observe a gradual improvement in accuracy. This indicates that imposing additional constraints and introducing new reasoning chains when necessary can aid the model in selecting the correct reasoning process. However, performance tends to saturate beyond a threshold of 2. We note that at a threshold of 4, more than 15% of samples in the GSM8K dataset fail to produce a final answer even upon reaching the maximum number of sampled reasoning chains. Furthermore, excessively high thresholds also lead to significant increases in computational costs. Therefore, we establish the termination threshold θ at 2, achieving an optimal balance between the accuracy of the outputs and the sampling costs of reasoning chains.

Analysis of Batch Size b. Figure 14 illustrates the impact of varying batch sizes (b) on accuracy and computational costs. During our analysis, samples exceeding the context window are excluded. We observe consistent performance improvements on both the GSM8K and AQuA datasets when evaluating multiple samples simultaneously, as opposed to assessing each sample individually. One possible explanation is that evaluating samples together allows the LLM to compare differences across reasoning chains, thereby providing more reliable scores. As the batch size increases, accuracy improves gradually until it reaches a batch size of 6, beyond which accuracy begins to fluctuate and even decline. At this point, the model's output becomes unstable, with some samples exceeding the model's context window, resulting in failed evaluations. Concurrently, we noted a gradual decrease in computational costs with increasing batch size, attributed to the reduced overhead of repetitive prompts. However, this trend starts to slow down when $b > 2$. Therefore, we selected a batch size of $b = 5$, which not only achieves optimal accuracy and lower computational costs but also avoids evaluation failures due to samples exceeding the model's context window.