

# Corex: Pushing the Boundaries of Complex Reasoning through Multi-Model Collaboration

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) are evolving at an unprecedented pace and have exhibited considerable capability in the realm of natural language processing (NLP) with world knowledge. Benefiting from ultra-large-scale training corpora, a single LLM can manage typical NLP tasks competently. However, its performance in executing complex tasks is still confined by the limitations of its internal representation. To push this boundary further, we introduce *Corex*, a suite of novel general-purpose strategies that transform LLMs into autonomous agents, pioneering multi-model collaborations for task-solving. Inspired by human behaviors, *Corex* is constituted by diverse collaboration paradigms including Discuss, Review, and Retrieve modes, which collectively work towards enhancing the reasoning process. These paradigms foster task-agnostic approaches that enable LLMs to “think outside the box,” thereby overcoming common errors and providing better solutions. Through extensive experiments across four different types of reasoning tasks, we demonstrate that orchestrating multiple LLMs to work in concert yields better results compared to existing strong methods. Further analysis reveals the cost-effectiveness of our method, while also exploring synergies between models of various scales and promoting annotation efficiency.<sup>1</sup>

## 1 Introduction

Large Language Models (LLMs) have succeeded in advancing the state-of-the-arts for a series of Natural Language Processing (NLP) tasks (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023; Touvron et al., 2023; Roziere et al., 2023; Zhao et al., 2023a, *inter alia*). Recent research (Wei et al., 2022a) indicates that scaling up models (Kaplan et al., 2020) can yield improvements in both performance and sample efficiency across a broad spectrum of downstream tasks. Notwithstanding their

<sup>1</sup>Our code and data are available at <https://anonymous.4open.science/r/Corex>.

remarkable proficiency in language understanding and instruction following (Ouyang et al., 2022), the reasoning abilities of LLMs, often seen as a hallmark for assessing their potential, still present challenges (Huang and Chang, 2023). Concurrently, there is a prevailing view that merely increasing model size might not adequately address their inherent limitations in reasoning (Rae et al., 2022).

In response, Wei et al. (2022b) put forth chain-of-thought (CoT) prompting that an LLM generates intermediate steps toward a final answer, contrasting the use of “answer-only” prompts. Subsequently, various approaches have been put forward, such as self-consistency decoding (Wang et al., 2023d) which utilizes a majority voting mechanism to determine the final answer, and program-aided language models (PAL; Gao et al., 2022; Chen et al., 2023b) that leverage code generation to facilitate computations. Nevertheless, these approaches are confined within a static black box (Yao et al., 2023b), wherein the LLM relies exclusively on its internal representation for generating responses and is prone to yielding unreliable answers (Ji et al., 2023; Yin et al., 2023b). These shortcomings underscore that relying solely on crafting decoding strategies and prompts may not serve as a silver bullet for addressing complex reasoning tasks (Qiao et al., 2023). Moreover, recent studies (Huang et al., 2023; Valmeekam et al., 2023; Stechly et al., 2023) reveal that LLMs struggle to reach better responses through self-correction. Alternatively, enabling models to expand their mindset emerges as a promising yet underexplored pathway.

Within the realm of well-established sociological concepts, multiple cognitive processes interact and cooperate will produce a combined effect that is greater than the sum of their individual contributions (Luppi et al., 2022). This principle is echoed within artificial intelligence (Li et al., 2023a). Although the study of intelligent agents has been explored for decades (Minsky, 1988, 2007), the

041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081



Figure 1: An intuitive illustration of *Corex*, employs LLMs as agents to collaboratively solve a problem. The strategies encompass the Discuss, Review, and Retrieve modes, leveraging both the reasoning process and code synthesis. This framework facilitates interactions between models that foster a collaborative environment for the derivation of a well-reasoned answer.

advent of LLMs has rejuvenated interest and introduced novel challenges in this domain. An emerging perspective is that encouraging collaboration and communication between models could potentially pave the way for a new stage for enhancing complex reasoning capabilities.

In this study, we propose *Corex*, a suite of human-inspired strategies that leveraging multi-model collaboration to elicit reasoning for complex task-solving. To facilitate synergies between models, we first assign distinct personas to different models, followed by the design of various collaborative paradigms. This collective intelligence-based method aims to conquer prevalent obstacles in the current landscape of reasoning. It also endeavors to mitigate issues observed in majority voting-based methods, where accurate responses might be overwhelmed by incorrect ones and exorbitant costs. To be specific, *Corex* configures LLMs as a group of autonomous agents, adopting the paradigms shown in Figure 1 for multi-model collaboration: (1) *Discuss*, utilizing group-based discussions among models to effectively enhance factuality and diversity of rationales, thereby reducing fallacies and hallucinations; (2) *Review*, enabling models to scrutinize reasoning chains or synthesized codes from their counterparts to ensure the correctness of generated contents, coupled with potential refinements; (3) *Retrieve*, aiming to enable the model to identify the most faithful option from a pool of candidate chains by ranking, facilitates a higher degree of alignment with the final response. Together, *Discuss*'s critical analysis, *Review*'s focus on reliability, and *Retrieve*'s alignment with faithful response - create a comprehensive approach to complex reasoning tasks.

We conduct extensive experiments with both commercial and open-source LLMs across four types of tasks: mathematical reasoning, symbolic

reasoning, commonsense reasoning, and semi-structured reasoning. The results illustrate that our method achieves substantial performance gains over previous strong baselines. Moreover, each mode distinctly excels in different categories of tasks, showcasing its specific strengths. Further analysis indicates that, in comparison to prevailing methods, *Corex* (1) substantially reduces the reasoning overhead, (2) ignites collaboration among heterogeneous models, and (3) achieves both cost-effectiveness and annotation efficiency.

## 2 Related works

**Chain-of-Thought Prompts Elicits LLM Reasoning.** Chain-of-Thought (CoT; Wei et al., 2022b) prompting, as one of the celebrated capabilities of recent LLMs, is a pivotal breakthrough for performing complex multi-step reasoning when provided with limited examples. Further variants show that CoT can be improved by adding certain “magic phrases” (Kojima et al., 2022), automated demonstrations construction (Zhang et al., 2023a), reasoning in different modalities (Zhang et al., 2023b; Yang et al., 2023; Yao et al., 2023c), and applying modular approaches (Khot et al., 2023). For robustness, researchers transform problems into interleaved reasoning chains (Zhou et al., 2023; Lyu et al., 2023) or adopt ensembling (Wang et al., 2022). Notably, self-consistency methods (Wang et al., 2023d) select answers from multiple reasoning paths by majority voting, have greatly elevated the performance of LLMs in complex reasoning. This approach has been further optimized by utilizing prompts with higher complexity (Fu et al., 2023c). Lately, Yao et al. (2023a) employ heuristic-guided search on “trees” constructed from thoughts to assist LLMs in navigating the problem space.

**Tool Utilization for LLM Reasoning.** Integrating external tools into language models presents a promising approach to enhance their reasoning capabilities, especially in numerical contexts (Mialon et al., 2023). A key example is the program-aided Language model (PAL; Gao et al., 2022; Chen et al., 2023b), which interprets natural language problems and generates programs for execution by external Python interpreters, effectively transforming reasoning tasks into a natural language to code (NL2Code) challenge. This method shows particular strength in handling complex numerical problems. Additionally, beyond mere program synthesis, approaches like integrating computational physics engines into the modeling process (Liu et al., 2023a), and combining LLMs with diverse tools and resources (Lu et al., 2023a), are expanding the scope of language model reasoning.

**Multi-Model Synergy for Task Solving.** Utilizing multiple LLMs collectively to solve problems is still in its preliminary stages, with a wealth of opportunities awaiting exploration. The cornerstone of collaboration is constructing a human-like reasoning architecture (Zhu et al., 2023) for LLMs under different environments (Liu et al., 2023c). Fu et al. (2023b) investigate whether multiple LLMs can autonomously enhance their performance through mutual interactions. Du et al. (2023) and Liang et al. (2023) explore enhancing the factuality of specific tasks, e.g., translation and arithmetic reasoning, by facilitating “debates” among multiple models. LLMs’ collaboration has also been applied to software development (Qian et al., 2023) and text evaluation (Chan et al., 2023) by assigning identities to models to simulate the development process. Furthermore, from the perspective of social intelligence, inducing cognitive synergy and having them take on different characters (Wang et al., 2023e; Yin et al., 2023a) during task execution has been proven to have significant potential (Sclar et al., 2023). Recently, the nascent exploration into artificial societies (Park et al., 2023) also seeks to harness collective intelligence to emulate the efficiency of human social structures (Li et al., 2023a; Webb et al., 2023).

### 3 Corex

We propose *Corex*, a suite of collaborative reasoning approaches inspired by human social interactions. Three main components: Discuss, Review, and Retrieve modes are introduced in the following

sections. Let us assume a set of LLM-based agents  $\{A_1, A_2, \dots, A_n\}$  participating in multi-model collaboration. Each agent  $A_i$  generates the corresponding reasoning chain  $c_i$  and its prediction  $p_i$  when facing a query  $q$ .

#### 3.1 Discuss

Information exchange between models contributes to enhanced factuality (Wang et al., 2023a). In Discuss mode, our agents are divided randomly into two groups, the Red Team and the Blue Team, with one reserved as a judge denoted as  $A_j$ . The discussion within one team involves several rounds, limited to a maximum of  $T$  rounds of communications. In each round  $t$  ( $t = 1, 2, \dots, T$ ), the agents engage in iterative discussions<sup>2</sup> to refine their reasoning chains and predictions. This dynamic interaction allows for the continual modification of viewpoints for  $q$ , as expressed by  $c_i^t$  and predictions  $p_i^t$ .

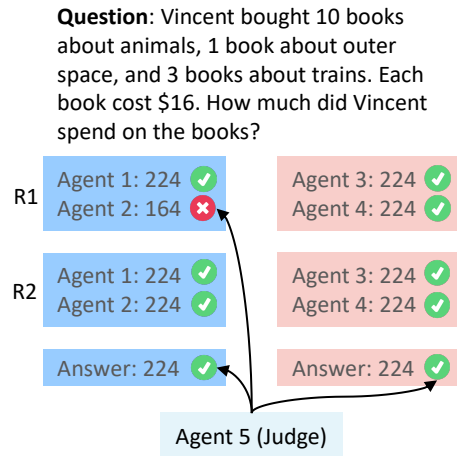


Figure 2: Illustration of 2 rounds of discussion, reasoning chains between agents omitted.

Each team then presents their refined predictions  $p_{\text{red}}^t$  and  $p_{\text{blue}}^t$  at the end of each round. If both teams consistently agree throughout the interactions, i.e.,  $p_{\text{red}}^t = p_{\text{blue}}^t$ , the discussion concludes smoothly. However, in the instance of a discrepancy between the teams’ predictions, every output from each round is presented to  $A_j$ . The judge employs a decision-making process  $h$  to evaluate the quality of the reasoning chains and predictions from each round of the discussion. The conclusion is determined by  $h(c_{\text{red}}^t, p_{\text{red}}^t, c_{\text{blue}}^t, p_{\text{blue}}^t)$  across all rounds, ensuring a comprehensive assessment and a more informed final decision.

<sup>2</sup>Due to the context length limit of GPT-3.5-Turbo, only information from the previous round is stored during the discussion.

Diverging from previous works (Du et al., 2023; Liang et al., 2023; Xiong et al., 2023) employing “debates” among LLMs, our approach adopts the concept of group discussions. We opt not to facilitate models in jointly exchanging their reasoning processes to converge on a single common answer for several reasons: (1) The limited context length inhibits the ability to hold the communication process, (2) Single final answers are not always correct due to erroneous consensus or biases (Wang et al., 2023c), (3) Given the performance gaps among various LLMs, there is a risk of strong models “monopolizing” the collaborations, thereby overshadowing the insights from others. Therefore, we aim to preserve both the factuality and the diversity of thoughts among agents.

### 3.2 Review

Both mainstream CoT and PAL methods are prone to errors in reasoning. Recent studies suggest that LLMs may have limited abilities in self-correcting capabilities (Huang et al., 2023; Valmeekam et al., 2023; Stechly et al., 2023). Specifically, CoT-based methods face two main issues: (1) Cumulative errors, where mistakes tend to amplify and propagate throughout the reasoning chain; and (2) A plateau in text quality that cannot be substantially improved through prompting (Xu et al., 2022; Li et al., 2023b). Alternatively, while PAL ensures computational accuracy (Zhao et al., 2023b), it also faces its challenges: (1) LLMs might misinterpret questions, which inadvertently results in technically correct yet misguided programs; and (2) Generated codes are not always error-free: LLMs may potentially write buggy codes, such as referencing undefined variables or engaging in “Division by Zero” operations. Inspired by recent efforts of LLMs peer-rating (Zheng et al., 2023) and collaborative coding practices prevalent in software engineering, we introduce the Review mode to address the aforementioned issues through collaboration.

To be specific, a single agent  $A_p$  is randomly selected to act as the primary agent. Initially,  $A_p$  takes the responsibility of formulating corresponding reasoning chains for  $q$  along with the prediction, and crafting codes if required. This initial collection of solutions is represented as  $S_p^{(0)} = \{a_p, c_p, m_p\}$ , where  $a_p$ ,  $c_p$ , and  $m_p$  signify the answer, reasoning chain, and codes respectively.  $S_p^{(0)}$  is then subjected to iterative reviews by the other agents that function as reviewers sequentially,

**Question:** While building the house, Charlie noticed that they were running out of nails so he told his father he's going to buy some. If they still have 9 nails left and Charlie bought 2 boxes of nails, the big one containing 55 nails and the small one containing 31, how many nails will they have?

**Agent1:**

```
# Python code by Agent1
def solution():
    current = 9
    big_box = 55
    small_box = 31
    total_nails = current + (2 *
(big_box + small_box))
    return total
```

**Agent2:** `total_nails = current + (2 * (big_box + small_box))` is wrong, which is currently doubling both the big and small boxes of nails. It is incorrect based on the problem statement. Here's the corrected code:

```
...
total_nails = current + big_box +
small_box
...
>>> Python Interpreter >>> 95 ✓
```

Figure 3: Illustration of reviewing erroneous code generated by other agents (first round).

rigorously scrutinizing both the reasoning chain and the code formulated by  $A_p$  or modified by preceding reviewers. It is crucial to highlight that each reviewer receives input from its predecessors, signifying that each subsequent review is grounded on the outcomes and feedback of the preceding ones, fostering a progressively refined solution.

The reviewing process is formalized as  $S_p^{(i+1)} = R_i(S_p^{(i)}, F_i)$ , where  $R_i$  encapsulates the review outcome at the  $i^{th}$  iteration and  $F_i$  represents the feedback received. In essence, the solution set  $S_p^{(i+1)}$  results from an enhancement of its preceding version  $S_p^{(i)}$ , informed by the feedback  $F_i$ . Following the completion of all review iterations, the outcome is determined by the final iteration of the solution set  $S_p^{(n-1)}$ . Specifically, the final prediction  $a_p^{(n-1)}$  is chosen as the answer for  $q$ , and in instances where code is involved, the last revised version  $m_p^{(n-1)}$  is executed by a Python interpreter to produce the outcome.

### 3.3 Retrieve

In the final thread of work, we delve into the Retrieve mode to identify the most faithful answer through collaborations. While previous strategies based on majority voting mechanism (Wang et al., 2023d; Fu et al., 2023c) can mitigate the low-diversity issue of techniques such as beam-search (Li and Jurafsky, 2016), they still present

the following two significant challenges: (1) Correct answers risk being swayed by incorrect ones. (2) Despite facilitating a notable enhancement in performance, it exponentially escalates the computational burden and tends to reach a performance “saturation point” as the sampled chains increase. We attribute these drawbacks to the limited scope of majority voting that prioritizes the prediction while overlooking the value of reasoning chains (Li et al., 2023c). In response, we propose the Retrieve mode, which is specifically engineered to evaluate whether the answer can be expressed by the content (explanation) generated during reasoning (Jacovi and Goldberg, 2020; Lanham et al., 2023).

**Question:** Which describes the composition of carbohydrates?

- (A) lipids bonding to form phospholipids
- (B) monomers bonding to form polymers
- (C) amino acids bonding to form polypeptides
- (D) saccharides bonding to form polysaccharides

**Agent1:** The composition of carbohydrates involves saccharides bonding to form polysaccharides. So the answer is (D). ✓

**Agent2:** The composition of carbohydrates is described by option (C), amino bonding to form polysaccharides. So the answer is (C). ✗

**Agent3:** The correct answer is (D) saccharides bonding to form polysaccharides. Carbohydrates are composed of monosaccharides that can join together through a process called dehydration synthesis to form larger polysaccharides. So the answer is (D). ✓

**Agent4:** Option (C), saccharides bonding to form polypeptides, describes the composition of carbohydrates. So the answer is (C). ✗

**Agent5(Retriever):** The confidence scores are: [0.9, 0.2, 0.9, 0.1]. Agents 2, 4 made errors in their descriptions. Specifically, Agent 2 confused amino acids with saccharides, while Agent 4 incorrectly stated that saccharides bond to form polypeptides. In reality, saccharides bond to form polysaccharides. Thus, they are not faithful in responses. Agents 1 and 3 are correct. ✓

Figure 4: Illustration of retrieving chains with answers.

Concretely, given a query  $q$ , we randomly select an agent  $A_r$  from the pool of  $n$  agents to act as the retriever. The remaining agents  $\{A_1, A_2, \dots, A_{n-1}\}$  independently perform CoT reasoning about  $q$ . Each of these agents derives its own reasoning chains  $c_i$  and corresponding predictions  $p_i$ . Together, they form a candidate pool, denoted by  $\mathcal{P} = \{(c_i, p_i)\}_{i=1}^{n-1}$

The retriever  $A_r$  then scrutinizes the candidates in  $\mathcal{P}$ . For  $(c_i, p_i)$ ,  $A_r$  evaluates the faithfulness between  $c_i$  and  $p_i$ . Based on this assessment, the retriever assigns a confidence score  $s_i$  in the range  $[0, 1]$ , which is denoted as:  $s_i = f_r(c_i, p_i)$  where  $f_r$  indicates the retriever’s evaluation process. Af-

ter that, the most faithful response to the question  $q$  is then determined by the highest confidence:

$$(c^*, p^*) = \operatorname{argmax}_{(c_i, p_i) \in \mathcal{P}} s_i$$

Here,  $(c^*, p^*)$  denotes the chain-prediction pair that the retriever considers most faithful, which will serve as the final answer for the query  $q$ .

Retrieve mode enables the selection of the most aligned combination of reasoning chains and answers from a diversified candidate pool. Distinct from previous text quality assessment methods, which rely on the log probability (Adiwardana et al., 2020) that is computationally inefficient and often unavailable for commercial LLMs, our approach is predicated on model-to-model interactions (Chen et al., 2023c) and is reference-free.

## 4 Experiment

### 4.1 Experimental Setup

**Tasks and Datasets.** We evaluate the effectiveness of *Corex* across four types of reasoning tasks: (1) Arithmetic reasoning over eight mathematical problems, which includes GSM8K (Cobbe et al., 2021), MultiArith (Roy and Roth, 2015), SingleOP/SingleEQ (Koncel-Kedziorski et al., 2016), AddSub (Hosseini et al., 2014), AQuA (Ling et al., 2017), SVAMP (Patel et al., 2021) and GSM-Hard (Gao et al., 2022). (2) Commonsense reasoning covering four datasets, including 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 incorporating four tasks from BigBench (bench authors, 2023; Suzgun et al., 2023), including Date Understanding, Penguins in a Table, Colored Objects, and Repeat Copy. (4) Semi-structured understanding, with a focus on FinQA (Chen et al., 2021b), ConvFinQA (Chen et al., 2022) and TATQA (Zhu et al., 2021). Detailed descriptions and statistics of tasks are listed in Appendix E.

**Baselines.** We compare our method with several widely used strong baselines. (1) Chain-of-Thought prompting (CoT; Wei et al., 2022b). (2) Self-Consistency (CoT-SC; Wang et al., 2023d), which employs a majority voting mechanism to select the most consistent answer from several reasoning chains as the final answer. (3) Complexity-based consistency (ComplexCoT; Fu et al., 2023c) that selects the majority answer from the candidates

	GSM8K	SVAMP	MultiArith	SingleOP	SingleEQ	AddSub	GSM-Hard	FinQA	ConvFinQA
CoT	74.5	78.9	98.5	94.1	93.3	87.8	39.0	46.1	50.4
ComplexCoT	79.7	80.7	97.3	94.3	92.3	86.8	39.7	-	-
CoT-SC(10)	<b>82.8</b>	84.5	<b>99.8</b>	95.4	95.1	89.6	45.2	52.7	57.2
PAL	76.0	83.4	96.7	90.7	95.8	87.6	<u>62.1</u>	54.3	50.8
<i>Corex-Discuss</i>	76.2	82.6	98.7	94.8	93.7	<u>89.7</u>	45.9	50.2	56.7
<i>Corex-Review<sub>NL</sub></i>	80.3	83.2	<u>99.5</u>	95.0	94.3	89.4	50.8	52.5	52.3
<i>Corex-Review<sub>Code</sub></i>	79.2	<b>85.8</b>	98.3	93.6	<b>96.9</b>	89.6	<b>63.6</b>	<b>55.9</b>	54.2
<i>Corex-Retrieve</i>	<u>82.5</u>	<u>85.6</u>	<b>99.8</b>	<b>96.1</b>	<u>96.6</u>	<b>90.9</b>	53.0	<u>55.4</u>	<b>57.7</b>

Table 1: Comparison of accuracy on various mathematical reasoning and semi-structured understanding datasets using *Corex* modes and strong baselines.

with higher reasoning complexity. (4) Program-aided language model (PAL; Gao et al., 2022; Chen et al., 2023b) that uses LLMs to generate programs as intermediate reasoning steps, while offloading the computation to a Python interpreter.

For simplicity and ease of understanding, we denote CoT-SC(x) and ComplexCoT(x) in our experiments and analysis to represent cases utilizing different reasoning paths, where “x” indicates the number of output chains. For all baseline methods, we adhere to the few-shot exemplars to ensure fair comparisons. Details can be found in Appendix A. **Implementation Details.** We access OpenAI and Anthropic models through their respective APIs for main evaluations. Specifically, we employ GPT-3.5-Turbo-0613 for evaluating both *Corex* and baseline methods in the main experiments. Moreover, in further experiments and analysis involving different LLMs for collaboration, we also incorporate the use of GPT-4-0613, Claude-Instant-1.2, and LLaMA-2-Chat (7B/13B). The details of prompts and hyperparameter settings for both baselines and *Corex* are in Appendix G.

## 4.2 Main Results

We report the results of *Corex* over four categories of tasks. For each kind of task, the best results are highlighted in **bold** and the second best results are marked with underline. For Review mode, we use *Corex-Review<sub>NL</sub>* and *Corex-Review<sub>Code</sub>* to describe the scenarios that use CoT or PAL respectively. All modes within *Corex* are configured to operate with 5 LLM-based agents, ensuring favorable cost-effectiveness. For *Corex-Discuss*, the upper bound of discuss rounds is set to 5.

**Mathematical & Semi-structured Reasoning.** Table 1 shows the results across tasks with varying difficulties. Our method achieves notable performance improvements on most benchmarks.

Broadly, we surpass CoT-SC(10) when only 5 agents are involved. Moreover, given the task-agnostic nature of *Corex*, it can tackle highly complex computational challenges like GSM-Hard through code synthesis. For problems of relatively lower complexity, the Retrieve mode can identify answers superior to those from majority voting.

We also demonstrate the results on FinQA and ConvFinQA. It can be observed that for these two challenging tasks which require understanding heterogeneous information and performing calculations simultaneously (Lu et al., 2023b), methods such as CoT-SC offer limited gains. However, through various cooperative paradigms, significant performance improvements can be achieved. Due to the context length restriction of GPT-3.5-Turbo, our experiments on TAT-QA utilized GPT-3.5-Turbo-16k, with the respective results being detailed in Appendix B.1, alongside the evaluations of the other tasks.

**Commonsense Reasoning.** Table 2 showcases the performance of *Corex* in commonsense and factual reasoning tasks<sup>3</sup>. We can observe that various modes contribute to performance enhancements.

	Strategy	CSQA	OpenBook	BoolQ	ARC
CoT	65.3	76.7	82.6	65.1	84.2
ComplexCoT	63.1	77.5	-	-	-
CoT-SC(10)	67.1	<u>78.1</u>	<u>85.2</u>	66.6	85.7
<i>Corex-Discuss</i>	<u>68.4</u>	<b>78.9</b>	83.4	<u>66.9</u>	<b>86.3</b>
<i>Corex-Review<sub>NL</sub></i>	66.9	77.4	84.8	<u>66.9</u>	<u>86.0</u>
<i>Corex-Retrieve</i>	<b>69.3</b>	77.7	<b>87.6</b>	<b>68.0</b>	85.5

Table 2: Comparison of performance on commonsense & factual reasoning between various *Corex* modes and strong baselines.

Notably, our approach surpasses ComplexCoT (over 6% on StrategyQA), achieving a significant improvement without resorting to intricate prompt design and example selection.

<sup>3</sup>Due to the nature of commonsense reasoning tasks, the Review mode only utilizes NL reasoning chains.

**Symbolic Reasoning.** We report the results for symbolic reasoning in Table 3. Empirical evidence substantiates that adopting multi-model collaboration can notably outperform most previous baselines on Big-Bench tasks. It is noteworthy that (1) CoT-SC struggles to ensure consistent outputs on the Repeat Copy. Conversely, through the integration of PAL-based collaboration, we manage to attain a remarkably high level of accuracy. (2) Compared to majority voting, both the Review and Retrieve modes enable more judicious answer selection in counting tasks.

	Date	Penguin	C.Objects	R.Copy
CoT	82.0	81.5	88.0	43.8
CoT-SC(10)	<b>87.9</b>	86.2	<u>94.8</u>	53.1
PAL	81.2	91.3	86.8	<u>93.8</u>
<i>Corex-Discuss</i>	83.2	85.9	91.2	62.5
<i>Corex-Review<sub>NL</sub></i>	84.0	92.0	92.4	59.4
<i>Corex-Review<sub>Code</sub></i>	82.7	<b>93.3</b>	91.6	<b>96.9</b>
<i>Corex-Retrieve</i>	<u>84.6</u>	92.6	<b>95.6</b>	68.8

Table 3: Comparison of accuracy on five symbolic reasoning datasets from Big-Bench (bench authors, 2023; Suzgun et al., 2023) using various Corex modes and other strong baselines.

Following our extensive experiments across 18 tasks, it emerges that the *Discuss mode* is competent for tasks utilizing factual knowledge. For mathematical and counting tasks, the *Review mode* serves to effectively mitigate errors within the reasoning chains and repair flawed code. Across various tasks, the *Retrieve mode* consistently facilitates performance improvements to varying degrees. More experiments for open-source LLMs (Touvron et al., 2023) are presented in Appendix C.

## 5 Analysis

In this section, we first aim to make the collaboration process transparent by delving into models’ internal behaviors. Then, the influence of different backbones is examined to observe how model capability affects performance. Further, we assess the efficiency of Corex.

### 5.1 In-Depth Analysis of Corex Strategies

**Analysis of Interaction Rounds in Discuss Mode.** We study the number of rounds of communication in the Discuss mode of Corex on five tasks, as depicted in Figure 5. Consensus can be reached swiftly for the majority of problems by each team. However, Corex enables LLMs to engage in more exhaustive discussions for problems that are chal-

lenging to reach a consensus on (e.g., over 10% of ConvFinQA problems requiring more than 3 rounds), a small proportion of problems require more interactions. Through observation, we also notice that the Discuss mode exhibits favorable convergence properties, wherein the interactive process serves as a basis for the judge’s decision-making.

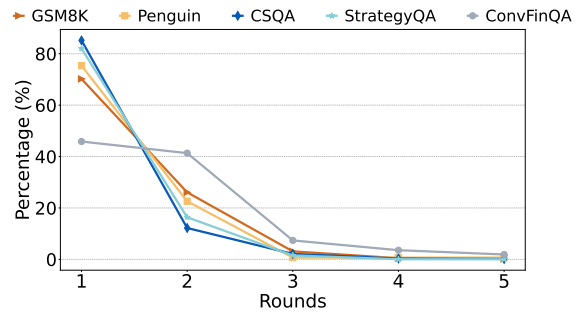


Figure 5: Distribution of the number of discuss rounds required to reach consensus.

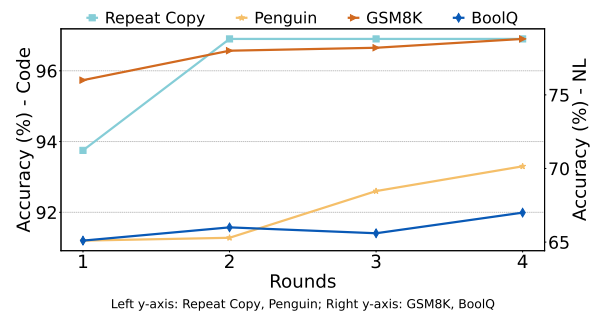


Figure 6: Perf. gains across multiple rounds of review

**Performance Enhancement per Review.** We explore the incremental performance gains achieved in specific tasks with each review cycle in the Review mode. As is demonstrated in Figure 6, we conduct analyses for Repeat Copy and GSM8K with *Review<sub>Code</sub>*, as long as BoolQ and Penguin with *Review<sub>NL</sub>*. The findings indicate that each review contributes to performance enhancement in general, yet occasional deviations leading to performance oscillations are also observed.

Further analysis of error types and gains from collaborations is presented in Appendix D.

### 5.2 Synergies between Different LLMs

**Performance Variability with Diverse LLMs as Judges.** The backbone LLMs of our agents can be diverse. In this part, we discuss the performance when employing different LLMs during the discussion. As shown in Figure 7, we deploy GPT-3.5-Turbo as players and examine the dynamics when different LLMs take the role of judges. The observations indicate that the capa-

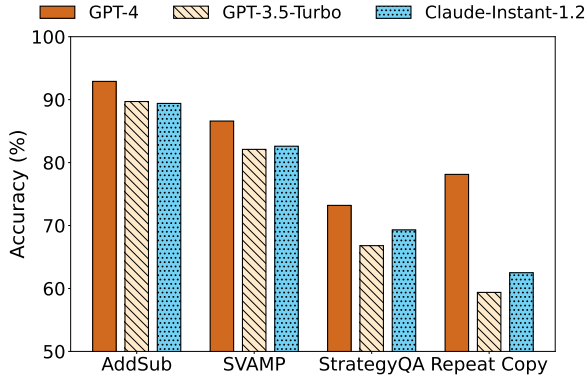


Figure 7: Comparison of using different LLMs as judges in Discuss mode.

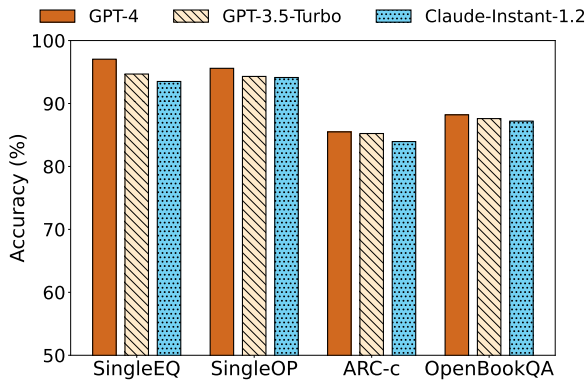


Figure 8: Comparison of using different LLMs as retrievers in Retrieve mode.

bility of the judge positively correlates with task performance, with this relationship being evident as the complexity of tasks escalates. Empirically, This can be attributed to the judge’s role, which requires understanding both the question and the reasoning process of both parties.

**Utilizing Different LLMs as Retrievers.** In Retrieve Mode, the role of the retriever can be played by various LLMs. Based on the candidate answers from GPT-3.5-Turbo agents, we here explore the impact of model selection on the performance, as depicted in Figure 8. Unlike the Discuss mode, our analysis reveals that the model capabilities exert a modest effect on the performance. Given that the performance upper bound is determined by the candidates’ capabilities, the outcomes using different LLMs as retrievers show minimal variance on tasks like ARC-c. Notably, our findings indicate that without the need for especially potent models as retrievers, we can still achieve favorable results.

### 5.3 Cost-Effectiveness Analysis

By encouraging collaboration between LLMs, we manage to reduce the costs associated with reasoning tasks while achieving comparable or even

superior performance. Based on our analysis conducted on AddSub illustrated in Figure 9, it reveals that all three modes of *Corex* consistently match or surpass the prowess of other strong baselines. Significantly, the computational cost of our approach are substantially diminished in comparison to methods using majority voting. In achieving equivalent performance, the resource consumption of *Corex* is confined to a mere 5-10% of that expended by other strategies. To substantiate the generality, we’ve provided additional experiments in Appendix B.2, which further demonstrate a similar trend.

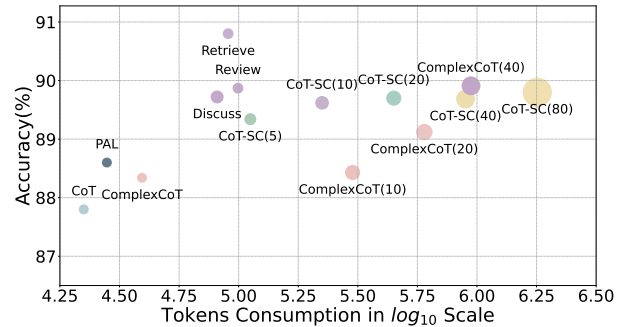


Figure 9: Cost-effectiveness analysis. the x-axis represents the computational costs, calculated in terms of input/output tokens, while the size of each dot is proportional to the avg. number of inferences by each method.

Beyond the efficiency of computational costs, another advantage of *Corex* is its annotation efficiency, which reduces the reliance on curated demonstrations. Further experiments with varying numbers of demonstrations on this aspect can be found in Appendix B.3.

## 6 Conclusion

We introduce *Corex* in this paper, a suite of human-inspired strategies that transform LLMs into autonomous agents, thereby leveraging multi-model collaboration for complex reasoning. This offers a preliminary exploration into the LLM-based multi-model ecosystems. Through unlocking the synergies among LLMs, *Corex* empowers reasoning with external insights through various collaboration paradigms. We conduct extensive evaluations across 18 tasks within 4 categories, and the results demonstrate superior performance compared to previous solutions. Moreover, our methods also exhibit multiple notable advantages including being task-agnostic, cost-effective, and annotation-efficient. We hope that this work may serve as a foundation for further research, offering novel perspectives in complex reasoning, collective intelligence, and autonomous agents.



## 585 Limitations

586 In this section, we discuss the limitations and  
587 broader impacts of our work.

588 **Model Selections.** Considering the limitations in  
589 handling long context, open-source models were  
590 only incorporated in Review mode (in Appendix C)  
591 in our study. However, we hold the view that the  
592 collaboration of several relatively weaker open-  
593 source models (Zhang et al., 2022; Touvron et al.,  
594 2023) might enable them to conquer more scenarios  
595 that only stronger commercial models (Bai et al.,  
596 2022; OpenAI, 2023) could previously accomplish.  
597 Since we have empirically demonstrated that model  
598 collaboration can help reduce redundant computa-  
599 tions and resource usage, it is foreseeable that the  
600 efficiency can be further accentuated when inte-  
601 grating open-source models into the collaboration  
602 framework, making the development of general ar-  
603 tificial intelligence more sustainable (Hessenthaler  
604 et al., 2022). Also, this may enhance the diversity  
605 to address the issues of biases and misconceptions  
606 inherent to a single model (Sun et al., 2022; Shaikh  
607 et al., 2023) can be mitigated, promoting the fair-  
608 ness of future AI systems (Hendrycks et al., 2021).

609 **System Stability.** In the experiments of this  
610 study, multi-model collaboration demonstrated sta-  
611 ble performance across multiple runs. However,  
612 recent work has pointed out issues of instability  
613 with mixed strategies (Yin et al., 2023a; Liu et al.,  
614 2023b). Therefore, when considering scaling up  
615 the number of participants, there may be an ob-  
616 served emergence of instability in integrating the  
617 functions of multiple LLMs. This highlights the  
618 need for more effective strategies to manage inter-  
619 actions among models with distinct attributes and  
620 capabilities (Parsons and McBurney, 2003; Dorri  
621 et al., 2018), which we identified as a pivotal direc-  
622 tion for future research.

## 623 References

624 Daniel Adiwardana, Minh-Thang Luong, David R. So,  
625 Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang,  
626 Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu,  
627 and Quoc V. Le. 2020. *Towards a human-like open-*  
628 *domain chatbot*. *CoRR*, abs/2001.09977.

629 Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda  
630 Askell, Anna Chen, Nova DasSarma, Dawn Drain,  
631 Stanislav Fort, Deep Ganguli, Tom Henighan, et al.  
632 2022. *Training a helpful and harmless assistant with*  
633 *reinforcement learning from human feedback*. *ArXiv*  
634 *preprint*, abs/2204.05862.

BIG bench authors. 2023. *Beyond the imitation game: Quantifying and extrapolating the capabilities of language models*. *Transactions on Machine Learning Research*. 635  
636  
637  
638

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. *Language models are few-shot learners*. In *NeurIPS*, volume 33, pages 1877–1901. 639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650

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*. 651  
652  
653  
654

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. *Accelerating large language model decoding with speculative sampling*. 655  
656  
657  
658

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al. 2021a. *Evaluating large language models trained on code*. *arXiv preprint arXiv:2107.03374*. 659  
660  
661  
662  
663

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023b. *Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks*. *Transactions on Machine Learning Research*. 664  
665  
666  
667  
668

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023c. *Exploring the use of large language models for reference-free text quality evaluation: An empirical study*. 669  
670  
671  
672

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. *FinQA: A dataset of numerical reasoning over financial data*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 673  
674  
675  
676  
677  
678  
679  
680  
681

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. *ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering*. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 682  
683  
684  
685  
686  
687  
688  
689

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul

692	Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways.	
693		
694		
695	Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In <i>Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)</i> , pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.	
696		
697		
698		
699		
700		
701		
702		
703		
704	Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. <i>ArXiv</i> , abs/1803.05457.	
705		
706		
707		
708		
709	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.	
710		
711		
712		
713		
714		
715	Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2018. Multi-agent systems: A survey. <i>IEEE Access</i> , 6:28573–28593.	
716		
717		
718	Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multi-agent debate. <i>arXiv preprint arXiv:2305.14325</i> .	
719		
720		
721		
722	Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023a. Chain-of-thought hub: A continuous effort to measure large language models’ reasoning performance.	
723		
724		
725		
726	Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023b. Improving language model negotiation with self-play and in-context learning from ai feedback.	
727		
728		
729	Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023c. Complexity-based prompting for multi-step reasoning. In <i>The Eleventh International Conference on Learning Representations</i> .	
730		
731		
732		
733	Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. <i>arXiv preprint arXiv:2211.10435</i> .	
734		
735		
736		
737	Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. <i>Transactions of the Association for Computational Linguistics</i> , 9:346–361.	
738		
739		
740		
741		
742		
743	Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning AI with shared human values. In <i>International Conference on Learning Representations</i> .	
744		
745		
746		
	Marius Hesse, Thore Graepel, and Dirk Hovy, and Anne Lauscher. 2022. Bridging fairness and environmental sustainability in natural language processing. In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 7817–7836, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	747 748 749 750 751 752 753
	Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In <i>Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)</i> , pages 523–533. Association for Computational Linguistics.	754 755 756 757 758 759 760
	Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In <i>Findings of the Association for Computational Linguistics: ACL 2023</i> , pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.	761 762 763 764 765
	Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet.	766 767 768 769
	Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In <i>Proc. of ACL</i> .	770 771 772
	Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. <i>ACM Comput. Surv.</i> , 55(12).	773 774 775 776 777
	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.	778 779 780 781
	Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In <i>The Eleventh International Conference on Learning Representations</i> .	782 783 784 785 786 787
	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In <i>Advances in Neural Information Processing Systems</i> .	788 789 790 791
	Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In <i>Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 1152–1157, San Diego, California. Association for Computational Linguistics.	792 793 794 795 796 797 798 799
	Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiuūtė, Karina Nguyen,	800 801 802 803

804	Newton Cheng, Nicholas Joseph, Nicholas Schiefer,	Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu	862
805	Oliver Rausch, Robin Larson, Sam McCandlish,	Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen	863
806	Sandipan Kundu, Saurav Kadavath, Shannon Yang,	Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Ao-	864
807	Thomas Henighan, Timothy Maxwell, Timothy	han Zeng, Zhengxiao Du, Chenhui Zhang, Sheng	865
808	Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds,	Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie	866
809	Jared Kaplan, Jan Brauner, Samuel R. Bowman, and	Huang, Yuxiao Dong, and Jie Tang. 2023c. Agent-	867
810	Ethan Perez. 2023. <a href="#">Measuring faithfulness in chain-</a>	bench: Evaluating llms as agents. <i>arXiv preprint</i>	868
811	<a href="#">of-thought reasoning</a> .	<i>arXiv: 2308.03688</i> .	869
812	Guohao Li, Hasan Abed Al Kader Hammoud, Hani	Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-	870
813	Itani, Dmitrii Khizbullin, and Bernard Ghanem.	Wei Chang, Ying Nian Wu, Song-Chun Zhu, and	871
814	2023a. Camel: Communicative agents for "mind" ex-	Jianfeng Gao. 2023a. Chameleon: Plug-and-play	872
815	ploration of large language model society. In <i>Thirty-</i>	compositional reasoning with large language models.	873
816	<i>seventh Conference on Neural Information Process-</i>	<i>arXiv preprint arXiv:2304.09842</i> .	874
817	<i>ing Systems</i> .		
818	Jiwei Li and Dan Jurafsky. 2016. Mutual information	Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu,	875
819	and diverse decoding improve neural machine trans-	Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark,	876
820	lation. <i>ArXiv</i> , abs/1601.00372.	and Ashwin Kalyan. 2023b. <a href="#">Dynamic prompt learn-</a>	877
821	Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang,	<a href="#">ing via policy gradient for semi-structured mathe-</a>	878
822	Jason Eisner, Tatsunori Hashimoto, Luke Zettle-	<a href="#">matical reasoning</a> . In <i>The Eleventh International</i>	879
823	moyer, and Mike Lewis. 2023b. <a href="#">Contrastive decod-</a>	<i>Conference on Learning Representations</i> .	880
824	<a href="#">ing: Open-ended text generation as optimization</a> . In	Andrea I Luppi, Pedro AM Mediano, Fernando E	881
825	<i>Proceedings of the 61st Annual Meeting of the As-</i>	Rosas, Negin Holland, Tim D Fryer, John T O'Brien,	882
826	<i>sociation for Computational Linguistics (Volume 1:</i>	James B Rowe, David K Menon, Daniel Bor, and	883
827	<i>Long Papers)</i> , pages 12286–12312, Toronto, Canada.	Emmanuel A Stamatakis. 2022. A synergistic core	884
828	Association for Computational Linguistics.	for human brain evolution and cognition. <i>Nature</i>	885
829	Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen,	<i>Neuroscience</i> , 25(6):771–782.	886
830	Jian-Guang Lou, and Weizhu Chen. 2023c. <a href="#">Making</a>	Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang,	887
831	<a href="#">language models better reasoners with step-aware</a>	Delip Rao, Eric Wong, Marianna Apidianaki, and	888
832	<a href="#">verifier</a> . In <i>Proceedings of the 61st Annual Meet-</i>	Chris Callison-Burch. 2023. Faithful chain-of-	889
833	<i>ing of the Association for Computational Linguistics</i>	thought reasoning. <i>arXiv preprint arXiv:2301.13379</i> .	890
834	<i>(Volume 1: Long Papers)</i> , pages 5315–5333, Toronto,	Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christo-	891
835	Canada. Association for Computational Linguistics.	foros Nalmpantis, Ram Pasunuru, Roberta Raileanu,	892
836	Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang,	Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu,	893
837	Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and	Asli Celikyilmaz, Edouard Grave, Yann LeCun, and	894
838	Shuming Shi. 2023. Encouraging divergent thinking	Thomas Scialom. 2023. <a href="#">Augmented language mod-</a>	895
839	in large language models through multi-agent debate.	<a href="#">els: a survey</a> .	896
840	<i>arXiv preprint arXiv:2305.19118</i> .	Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish	897
841	Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blun-	Sabharwal. 2018. <a href="#">Can a suit of armor conduct elec-</a>	898
842	som. 2017. <a href="#">Program induction by rationale genera-</a>	<a href="#">tricity? a new dataset for open book question</a>	899
843	<a href="#">tion: Learning to solve and explain algebraic word</a>	<a href="#">answering</a> . In <i>Proceedings of the 2018 Conference on</i>	900
844	<a href="#">problems</a> . In <i>Proceedings of the 55th Annual Meet-</i>	<i>Empirical Methods in Natural Language Processing</i> ,	901
845	<i>ing of the Association for Computational Linguistics,</i>	pages 2381–2391, Brussels, Belgium. Association	902
846	<i>ACL 2017, Vancouver, Canada, July 30 - August 4,</i>	for Computational Linguistics.	903
847	<i>Volume 1: Long Papers</i> , pages 158–167. Association	Marvin Minsky. 1988. <i>Society of mind</i> . Simon and	904
848	for Computational Linguistics.	Schuster.	905
849	Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu,	Marvin Minsky. 2007. <i>The emotion machine: Common-</i>	906
850	Soroush Vosoughi, Claire Cui, Denny Zhou, and An-	<i>sense thinking, artificial intelligence, and the future</i>	907
851	drew M. Dai. 2023a. <a href="#">Mind's eye: Grounded language</a>	<i>of the human mind</i> . Simon and Schuster.	908
852	<a href="#">model reasoning through simulation</a> . In <i>The Eleventh</i>	OpenAI. 2023. <a href="#">GPT-4 technical report</a> .	909
853	<i>International Conference on Learning Representa-</i>	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,	910
854	<i>tions</i> .	Carroll Wainwright, Pamela Mishkin, Chong Zhang,	911
855	Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu,	Sandhini Agarwal, Katarina Slama, Alex Gray, John	912
856	Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023b.	Schulman, Jacob Hilton, Fraser Kelton, Luke Miller,	913
857	<a href="#">Plan, verify and switch: Integrated reasoning with</a>	Maddie Simens, Amanda Askell, Peter Welinder,	914
858	<a href="#">diverse X-of-thoughts</a> . In <i>Proceedings of the 2023</i>	Paul Christiano, Jan Leike, and Ryan Lowe. 2022.	915
859	<i>Conference on Empirical Methods in Natural Lan-</i>	<a href="#">Training language models to follow instructions with</a>	916
860	<i>guage Processing</i> , pages 2807–2822, Singapore. As-	<a href="#">human feedback</a> . In <i>Advances in Neural Information</i>	917
861	sociation for Computational Linguistics.	<i>Processing Systems</i> .	918

919	Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simula-	let's not think step by step! bias and toxicity in zero-	976
920	lactra of human behavior. In <i>In the 36th Annual ACM</i>	shot reasoning. In <i>Proceedings of the 61st Annual</i>	977
921	<i>Symposium on User Interface Software and Technol-</i>	<i>Meeting of the Association for Computational Lin-</i>	978
922	<i>ogy (UIST '23)</i> , UIST '23, New York, NY, USA.	<i>guistics (Volume 1: Long Papers)</i> , pages 4454–4470,	979
923	Association for Computing Machinery.	Toronto, Canada. Association for Computational Lin-	980
924		guistics.	981
925			
926	Simon Parsons and Peter McBurney. 2003.	Kaya Stechly, Matthew Marquez, and Subbarao Kamb-	982
927	<a href="#">Argumentation-based communication between</a>	hampati. 2023. <a href="#">Gpt-4 doesn't know it's wrong: An</a>	983
928	<a href="#">agents</a> . <i>Lecture Notes in Artificial Intelligence</i>	<a href="#">analysis of iterative prompting for reasoning prob-</a>	984
929	<i>(Subseries of Lecture Notes in Computer Science)</i> ,	<a href="#">lems</a> .	985
930	2650:164–178.		
931	Arkil Patel, Satwik Bhattamishra, and Navin Goyal.	Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuan-	986
932	2021. <a href="#">Are NLP models really able to solve simple</a>	jing Huang. 2022. <a href="#">BERTScore is unfair: On social</a>	987
933	<a href="#">math word problems?</a> In <i>Proceedings of the 2021</i>	<a href="#">bias in language model-based metrics for text</a>	988
934	<i>Conference of the North American Chapter of the</i>	<a href="#">generation</a> . In <i>Proceedings of the 2022 Conference on</i>	989
935	<i>Association for Computational Linguistics: Human</i>	<i>Empirical Methods in Natural Language Processing</i> ,	990
936	<i>Language Technologies</i> , pages 2080–2094, Online.	pages 3726–3739, Abu Dhabi, United Arab Emirates.	991
937	Association for Computational Linguistics.	Association for Computational Linguistics.	992
938	Chen Qian, Xin Cong, Cheng Yang, Weize Chen,	Mirac Suzgun, Nathan Scales, Nathanael Schärli, Se-	993
939	Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong	bastian Gehrmann, Yi Tay, Hyung Won Chung,	994
940	Sun. 2023. <a href="#">Communicative agents for software de-</a>	Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny	995
941	<a href="#">velopment</a> .	Zhou, and Jason Wei. 2023. <a href="#">Challenging BIG-bench</a>	996
942		<a href="#">tasks and whether chain-of-thought can solve them</a> .	997
943	Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen,	In <i>Findings of the Association for Computational Lin-</i>	998
944	Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang,	<i>guistics: ACL 2023</i> , pages 13003–13051, Toronto,	999
945	and Huajun Chen. 2023. <a href="#">Reasoning with language</a>	Canada. Association for Computational Linguistics.	1000
946	<a href="#">model prompting: A survey</a> . In <i>Proceedings of the</i>		
947	<i>61st Annual Meeting of the Association for Comput-</i>	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and	1001
948	<i>ational Linguistics (Volume 1: Long Papers)</i> , pages	Jonathan Berant. 2019. <a href="#">CommonsenseQA: A ques-</a>	1002
949	5368–5393, Toronto, Canada. Association for Com-	<a href="#">tional answering challenge targeting commonsense</a>	1003
950		<a href="#">knowledge</a> . In <i>Proceedings of the 2019 Conference</i>	1004
951	Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie	<i>of the North American Chapter of the Association for</i>	1005
952	Millican, Jordan Hoffmann, Francis Song, John	<i>Computational Linguistics: Human Language Tech-</i>	1006
953	Aslanides, Sarah Henderson, Roman Ring, Susan-	<i>nologies, Volume 1 (Long and Short Papers)</i> , pages	1007
954	nah Young, et al. 2022. <a href="#">Scaling language models:</a>	4149–4158, Minneapolis, Minnesota. Association for	1008
955	<a href="#">Methods, analysis &amp; insights from training gopher</a> .	Computational Linguistics.	1009
956			
957	Subhro Roy and Dan Roth. 2015. <a href="#">Solving general arith-</a>	Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-	1010
958	<a href="#">metic word problems</a> . In <i>Proceedings of the 2015</i>	bert, Amjad Almahairi, Yasmine Babaei, Nikolay	1011
959	<i>Conference on Empirical Methods in Natural Lan-</i>	Bashlykov, Soumya Batra, Prajjwal Bhargava, et al.	1012
960	<i>guage Processing, EMNLP 2015, Lisbon, Portugal,</i>	2023. <a href="#">Llama 2: Open foundation and fine-tuned chat</a>	1013
961	<i>September 17-21, 2015</i> , pages 1743–1752. The As-	<a href="#">models</a> .	1014
962	sociation for Computational Linguistics.		
963	Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten	Karthik Valmeekam, Matthew Marquez, and Subbarao	1015
964	Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi,	Kambhampati. 2023. <a href="#">Can large language models</a>	1016
965	Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023.	<a href="#">really improve by self-critiquing their own plans?</a>	1017
966	Code llama: Open foundation models for code. <i>arXiv</i>		
967	<i>preprint arXiv:2308.12950</i> .	Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru	1018
968	Melanie Sclar, Sachin Kumar, Peter West, Alane Suhr,	Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao,	1019
969	Yejin Choi, and Yulia Tsvetkov. 2023. <a href="#">Minding lan-</a>	Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang,	1020
970	<a href="#">guage models' (lack of) theory of mind: A plug-and-</a>	Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang,	1021
971	<a href="#">play multi-character belief tracker</a> . In <i>Proceedings</i>	and Yue Zhang. 2023a. <a href="#">Survey on factuality in large</a>	1022
972	<i>of the 61st Annual Meeting of the Association for</i>	<a href="#">language models: Knowledge, retrieval and domain-</a>	1023
973	<i>Computational Linguistics (Volume 1: Long Papers)</i> ,	<a href="#">specificity</a> .	1024
974	pages 13960–13980, Toronto, Canada. Association	Jianing Wang, Nuo Chen, Qiushi Sun, Wenkang Huang,	1025
975	for Computational Linguistics.	Chengyu Wang, and Ming Gao. 2023b. <a href="#">HugNLP:</a>	1026
		<a href="#">A unified and comprehensive library for natural lan-</a>	1027
		<a href="#">guage processing</a> . In <i>Proceedings of the 32nd ACM</i>	1028
		<i>International Conference on Information and Knowl-</i>	1029
		<i>edge Management (CIKM '23)</i> , Demo Papers. Asso-	1030
		ciation for Computing Machinery.	1031

1032	Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023c. <a href="#">Large language models are not fair evaluators</a> .	1088
1033		1089
1034		1090
1035	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. <a href="#">Rationale-augmented ensembles in language models</a> .	1091
1036		1092
1037		
1038	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. <a href="#">Self-consistency improves chain of thought reasoning in language models</a> . In <i>The Eleventh International Conference on Learning Representations</i> .	1093
1039		1094
1040		1095
1041		
1042		
1043		
1044	Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023e. <a href="#">Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration</a> .	1100
1045		1101
1046		1102
1047		1103
1048	Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. <i>Nature Human Behaviour</i> , pages 1–16.	1104
1049		1105
1050		1106
1051	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. <a href="#">Emergent abilities of large language models</a> . <i>Transactions on Machine Learning Research</i> .	1107
1052		
1053		
1054		
1055		
1056		
1057		
1058	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. <a href="#">Chain of thought prompting elicits reasoning in large language models</a> . In <i>Advances in Neural Information Processing Systems</i> .	1108
1059		1109
1060		1110
1061		1111
1062		1112
1063	Zhenyu Wu, Yaoliang Wang, Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Jingjing Xu, and Yu Qiao. 2023. <a href="#">OpenICL: An open-source framework for in-context learning</a> . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)</i> , pages 489–498, Toronto, Canada. Association for Computational Linguistics.	1113
1064		1114
1065		1115
1066		1116
1067		1117
1068		1118
1069		
1070		
1071	Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. <a href="#">Examining the inter-consistency of large language models: An in-depth analysis via debate</a> .	1119
1072		1120
1073		1121
1074	Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. <a href="#">Learning to break the loop: Analyzing and mitigating repetitions for neural text generation</a> . In <i>Advances in Neural Information Processing Systems</i> .	1122
1075		1123
1076		1124
1077		1125
1078		1126
1079	Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. <a href="#">Mm-react: Prompting chatgpt for multimodal reasoning and action</a> .	1127
1080		1128
1081		1129
1082		1130
1083		1131
1084	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. <a href="#">Tree of Thoughts: Deliberate problem solving with large language models</a> .	1132
1085		1133
1086		1134
1087		1135
	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. <a href="#">React: Synergizing reasoning and acting in language models</a> . In <i>The Eleventh International Conference on Learning Representations</i> .	1136
		1137
	Yao Yao, Zuchao Li, and Hai Zhao. 2023c. <a href="#">Beyond chain-of-thought, effective graph-of-thought reasoning in large language models</a> .	1138
		1139
	Xi Ye and Greg Durrett. 2022. <a href="#">The unreliability of explanations in few-shot prompting for textual reasoning</a> . In <i>Advances in Neural Information Processing Systems</i> .	1140
		1141
	Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023a. <a href="#">Exchange-of-thought: Enhancing large language model capabilities through cross-model communication</a> . In <i>Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing</i> , pages 15135–15153, Singapore. Association for Computational Linguistics.	1142
		1143
	Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023b. <a href="#">Do large language models know what they don't know?</a> In <i>Findings of the Association for Computational Linguistics: ACL 2023</i> , pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.	1144
		1145
	Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. <a href="#">Opt: Open pre-trained transformer language models</a> . <i>ArXiv preprint</i> , abs/2205.01068.	1146
		1147
	Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023a. <a href="#">Automatic chain of thought prompting in large language models</a> . In <i>The Eleventh International Conference on Learning Representations</i> .	1148
		1149
	Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023b. <a href="#">Multi-modal chain-of-thought reasoning in language models</a> . <i>arXiv preprint arXiv:2302.00923</i> .	1150
		1151
	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. 2023a. <a href="#">A survey of large language models</a> .	1152
		1153
	Xu Zhao, Yuxi Xie, Kenji Kawaguchi, Junxian He, and Qizhe Xie. 2023b. <a href="#">Automatic model selection with large language models for reasoning</a> . <i>ArXiv</i> .	1154
		1155
	Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. <a href="#">Judging llm-as-a-judge with mt-bench and chatbot arena</a> .	1156
		1157

1142 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei,  
1143 Nathan Scales, Xuezhi Wang, Dale Schuurmans,  
1144 Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H.  
1145 Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations*.  
1146  
1147  
1148

1149 Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao  
1150 Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-  
1151 Seng Chua. 2021. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.  
1152  
1153  
1154  
1155  
1156  
1157  
1158

1159 Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang,  
1160 Yongfeng Huang, Ruyi Gan, Jiaying Zhang, and Yu-  
1161 jiu Yang. 2023. [Solving math word problems via cooperative reasoning induced language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4471–4485, Toronto, Canada. Association for Computational Linguistics.  
1162  
1163  
1164  
1165  
1166

## A Implementation Details 1167

**LLMs Settings and Prompts.** We access OpenAI and Anthropic models through their respective paid APIs. Specifically, we utilize two versions of the OpenAI<sup>4</sup> models: GPT-3.5-Turbo-0613 and GPT-4-0613, as well as Claude-Instant-1.2 from Anthropic<sup>5</sup>. Following the settings of previous works, the temperature is set as 0 during the generation process. For LLaMA-2-Chat model from Meta, the beam size is set to 2. 1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177

As for the prompts, we maintain the same number of exemplars per task as established by previous studies. The details and examples of prompts are listed in Appendix G. Regarding the complexity-based prompting baseline, we directly utilize the prompts provided by Fu et al. (2023a). 1178  
1179  
1180  
1181  
1182  
1183

For experiments involving APIs, our main experiments (Section 4.2 and Appendix B.1), analysis (Section 5, Appendix B.2 and Appendix B.3) and Case Studies (Appendix F) were completed between August and September 2023. Experiments involving interactions between commercial and open-source models (Appendix C), were conducted between November and December 2023. 1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191

**Evaluation.** We use accuracy to evaluate the tasks of arithmetic reasoning, commonsense reasoning, and Symbolic reasoning. For semi-structured understanding, we employ the official evaluation scripts released along with the literature for FinQA (Chen et al., 2021b) and ConvFinQA (Chen et al., 2022), as well as employing the same test split as in Chen et al. (2023b). For TAT-QA (Zhu et al., 2021), we utilize the arithmetic part to enable the simultaneous use of both CoT-based methods and PAL. Among all the evaluations involve *Corex-Discuss*, the upper bound of rounds is set as 5. 1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204

In the analysis part, due to the rate limits imposed and a restricted budget, we set an upper limit for our sample size. Consequently, each analysis is based on a maximum of 500 samples per run. 1205  
1206  
1207  
1208

We draw upon the design of the self-consistency baseline as delineated by Wu et al. (2023), and partially refer to the benchmarking provided by Wang et al. (2023b) (Apache-2.0 license). 1209  
1210  
1211  
1212

<sup>4</sup><https://openai.com/blog/openai-api>

<sup>5</sup><https://docs.anthropic.com/claude/reference>

**PAL Details.** We use Python 3.9.12 to execute the generated program by PAL in Review mode, following the similar settings from Gao et al. (2022) and Chen et al. (2023b). Unlike Code-davinci-002 (Chen et al., 2021a) employed in these works, GPT-3.5-Turbo and LLaMA-2-Chat are not optimized for generating programs, so we might anticipate a compromise in performance when using them as the backbone model.

## B Extended Results and Analysis

### B.1 Evaluations on Semi-structured Reasoning

As is mentioned in section 4.2, due to the constraint of context length, we conduct experiments on the TAT-QA dataset with GPT-3.5-Turbo-16k along with the other two tasks.

	FinQA	ConvFinQA	TAT-QA	Avg.
CoT	48.0	51.8	69.6	56.5
CoT-SC(10)	51.5	<u>56.5</u>	<b>76.6</b>	<u>61.5</u>
PAL	54.9	51.1	73.4	59.8
<i>Corex-Discuss</i>	50.5	55.8	72.6	59.6
<i>Corex-Review<sub>Code</sub></i>	<b>56.0</b>	52.7	<u>75.2</u>	61.3
<i>Corex-Retrieve</i>	<u>55.1</u>	<b>57.9</b>	73.7	<b>62.2</b>

Table 4: Evaluation of *Corex* on semi-structured understanding tasks with GPT-3.5-Turbo-16k among three tasks.

### B.2 Further Cost-effectiveness Analysis

For all computations related to cost-effectiveness, we define cost as the sum of: tokens of few-shot exemplars, query tokens, and output tokens. Here we conduct additional cost-effectiveness analysis on the ARC-c dataset and Penguins dataset, as shown in Figure 10 and Figure 11. The x-axis represents the computational costs, calculated in terms of input/output tokens, while the size of each dot is proportional to the avg. number of inferences by each method.

### B.3 Annotation Efficiency Analysis

Due to the scarcity and instability of curated prompts (Ye and Durrett, 2022), we aim to mitigate reliance on them through multi-model collaboration. We conduct experiments in scenarios with varying numbers of demonstrations to assess the effectiveness of our approach.

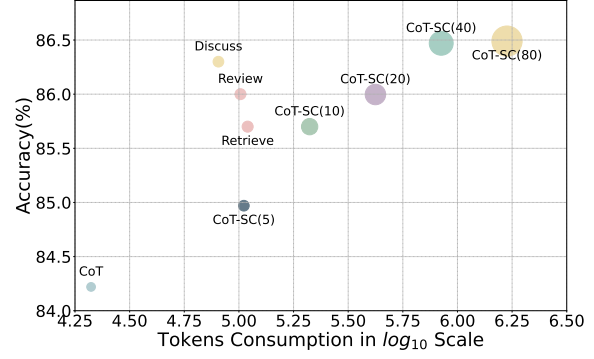


Figure 10: Cost-effectiveness analysis for ARC-c dataset.

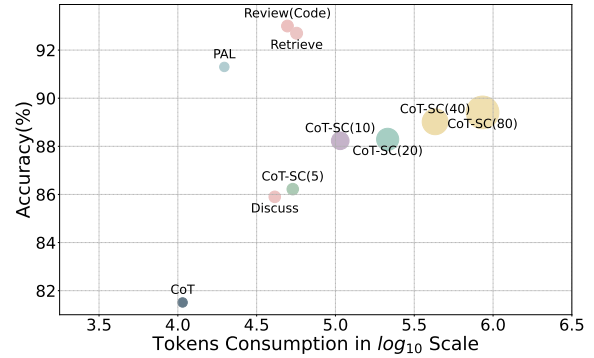


Figure 11: Cost-effectiveness analysis for Penguins in a Table.

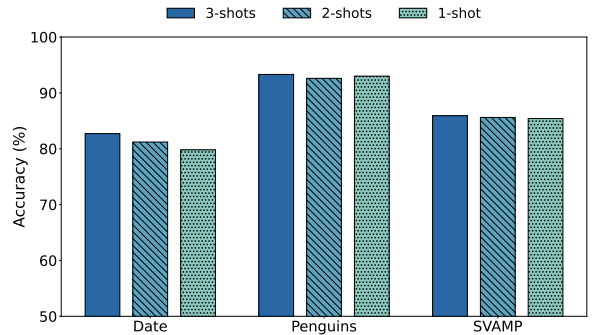


Figure 12: Few-shot performance of Review mode.

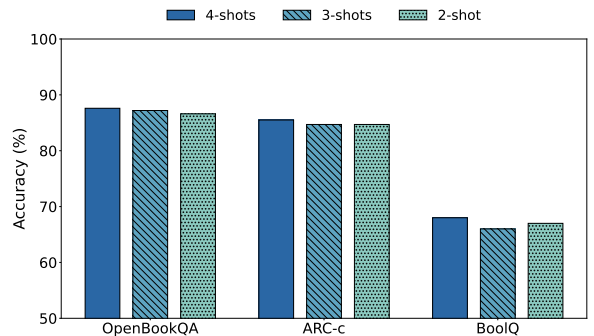


Figure 13: Few-shot performance of Retrieve mode.

As depicted in Figure 12 and Figure 13, we

demonstrate the performance of the Review mode and Retrieve mode under different shots. The results reveal a marginal impact of the number of few-shot examples on performance, underscoring the annotation efficiency of our approach.

### C Collaborations among Open-Source LLMs

In this part, we explore the synergies between commercial and open-source LLMs. We select LLaMA-2-Chat (7B/13B) in conjunction with GPT-3.5-Turbo and Claude-Instant-1.2 for review mode. During the collaboration process, LLaMA models assume the role of the primary agent, while the other models function as reviewers.

The performance using  $Review_{NL}$  is depicted in Figure 14 and Figure 15, while the performance utilizing  $Review_{Code}$  is illustrated in Figure 16 and Figure 17.

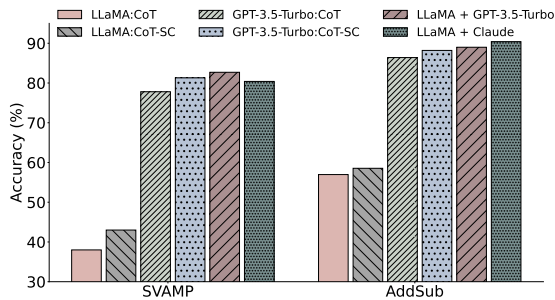


Figure 14: Collaboration with LLaMA-2-Chat(7B) in  $Review_{NL}$ .

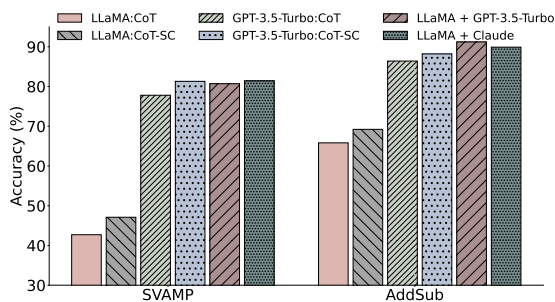


Figure 15: Collaboration with LLaMA-2-Chat(13B) in  $Review_{NL}$ .

In evaluations, it can be observed that although open-source LLMs exhibit weaker performance when operating independently, through collaboration with other models, they can achieve performance comparable to or even surpass that of CoT-SC(10) by commercial models.

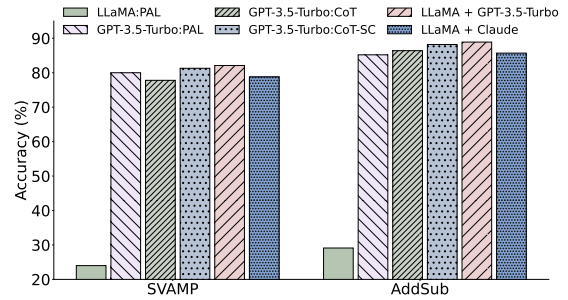


Figure 16: Collaboration with LLaMA-2-Chat(7B) in  $Review_{Code}$ .

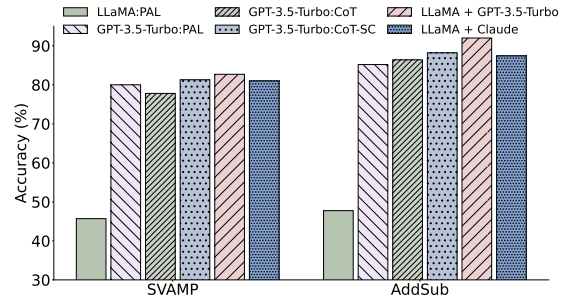


Figure 17: Collaboration with LLaMA-2-Chat(13B) in  $Review_{Code}$ .

In the  $Review_{Code}$ , although LLaMA’s capability in code generation and task delegation is limited on its own, its collaboration with other models yields non-trivial results. Based on this, we have reason to believe that employing smaller models first to generate a “draft” for the desired content (Chen et al., 2023a) is beneficial in achieving better solutions, even though there is a significant gap between the capabilities of LLMs.

### D Analysis of Error with Different Types

In section 5.1, we analyze the performance changes brought about by each review cycle. Here, we further delve into an analysis of the errors made by the models and how collaborations help in mitigating these errors on GSM8K and GSM-Hard datasets.

Our analysis focuses on the  $Review_{Code}$  mode. The types of errors in this mode, as discussed in section 3.2, can be categorized into (1) Bugs, where generated codes are not always error-free, and (2) Problem misinterpretation, where models produce technically correct yet misguided programs.

As depicted in Figure 18, following the review process, there is a noticeable reduction in both types of errors committed by LLMs on GSM8K. Moreover, as illustrated in Figure 19, it can be observed that in more challenging problems from GSM-Hard, collaboration leads to the elimination



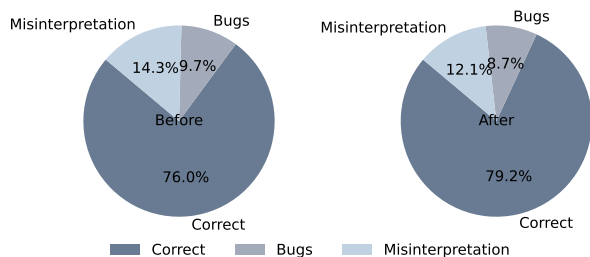


Figure 18: Error Types Analysis on GSM8K: Pre- and Post-collaboration.

of a greater number of errors caused by incorrect problem understanding.

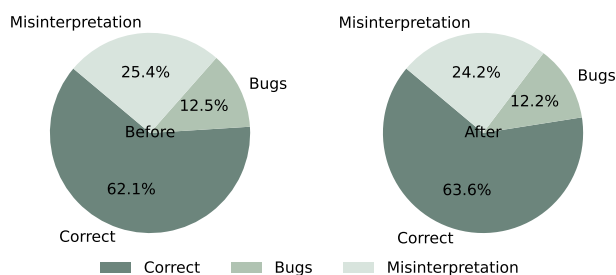


Figure 19: Error Types Analysis on GSM-Hard: Pre- and Post-collaboration.

## E Statistics and Details of Datasets

The detailed information of each dataset is shown in the follow:

### Arithmetic reasoning

- Grade School Math (GSM8K; Cobbe et al., 2021): Linguistically diverse grade school math word problems created by human problem writers. The problems take between 2 and 8 steps to solve and involve elementary calculations using basic arithmetic operations.
- AddSub (Hosseini et al., 2014): A set of simple arithmetic word problems.
- SVAMP (Patel et al., 2021): A challenge set for elementary-level Math Word Problems.
- SingleOP, SingleEQ and MultiArith (Koncel-Kedziorski et al., 2016): Grade-school math dataset that aims at solving multi-sentence algebraic word problems.
- GSM-Hard (Gao et al., 2022): A harder version of the GSM8K dataset, constructed by replacing the numbers in the questions of GSM8K with larger numbers.

### Commonsense & Factual reasoning

- CommonsenseQA (CSQA; Talmor et al., 2019): CSQA is a multiple-choice question answering task. It requires complex semantic reasoning based on prior commonsense knowledge to answer the questions.
- StrategyQA (Geva et al., 2021): It is a commonsense QA task with Yes or No answer format that requires models to perform multi-hop reasoning to answer the questions. We use the open-domain setting (question-only set) from bench authors (2023).
- OpenBookQA (Mihaylov et al., 2018): It is a multi-choice QA task to evaluate commonsense knowledge and promote reasoning over a fixed collection of knowledge.
- ARC-c (Clark et al., 2018): A subset of the AI2 Reasoning Challenge, consisting of challenging science questions that require reasoning and a wide breadth of knowledge to answer the multiple-choice problems correctly. The original dataset is from <https://allenai.org/data/arc>.
- BoolQ (Clark et al., 2019): It is a knowledge-intensive task and the format is “Yes” or “No”. Problems are extracted from real-world internet queries, aiming to foster models capable of contextual understanding to provide binary answers.

**Symbolic Reasoning** We select the following tasks from BIG-Bench (bench authors, 2023) and BIG-Bench Hard (BBH) (Suzgun et al., 2023), with Apache License v.2: <https://github.com/google/BIG-bench/blob/main/LICENSE>.

- Date Understanding: A temporal reasoning task. Given a set of sentences about a particular date, answer the provided question in MM/DD/YYYY format.
- Object Counting: Given a collection of possessions that a person has along with their quantities (e.g., three pianos, two strawberries, one table, and two watermelons), determine the number of a certain object/item class (e.g., fruits).

Dataset	N	Example
GSM8K (Cobbe et al., 2021)	1,319	A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
GSM-Hard (Gao et al., 2022)	1,319	A robe takes 2287720 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
SVAMP (Patel et al., 2021)	1,000	Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?
SINGLEOP (Koncel-Kedziorski et al., 2016)	562	If there are 7 bottle caps in a box and Linda puts 7 more bottle caps inside, how many bottle caps are in the box?
SINGLEEQ (Koncel-Kedziorski et al., 2016)	508	Benny bought a soft drink for 2 dollars and 5 candy bars. He spent a total of 27 dollars. How much did each candy bar cost?
AddSub (Hosseini et al., 2014)	395	There were 6 roses in the vase. Mary cut some roses from her flower garden. There are now 16 roses in the vase. How many roses did she cut?
MultiArith (Roy and Roth, 2015)	600	The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

Table 5: Examples from mathematical reasoning datasets used in this work.

1370	<ul style="list-style-type: none"> <li>• Penguins in a Table: Given a unique table of penguins (and sometimes some new information), answer a question about the attributes of the penguins.</li> </ul>	over more complex and realistic tabular and	1389	
1371		textual data.	1390	
1372				
1373				
1374	<ul style="list-style-type: none"> <li>• Reasoning about Colored Objects: Given a context, answer a simple question about the color of an object on a surface.</li> </ul>	<b>F Case Studies</b>	1391	
1375		We have prepared case studies to provide a more	1392	
1376		intuitive understanding of the three collaboration	1393	
1377	<ul style="list-style-type: none"> <li>• Repeat Copy: Evaluate LLMs’ capability to follow basic natural-language instructions nested within each example’s input.</li> </ul>	paradigms. The cases are presented in Table 9,	1394	
1378		Table 10, and Table 11, each corresponding to a	1395	
1379		specific collaboration paradigm.	1396	
1380	<b>Semi-structured Understanding</b>	<b>G Prompts</b>	1397	
1381	<ul style="list-style-type: none"> <li>• FinQA (Chen et al., 2021b): Question-Answering pairs over financial reports written by experts, which includes financial QA pairs.</li> </ul>	Owing to space constraints and in an effort to main-	1398	
1382		tain a reasonable length for this paper, we present	1399	
1383	<ul style="list-style-type: none"> <li>• ConvFinQA (Chen et al., 2022): A financial-related dataset designed to study the chain of numerical reasoning in conversational QA.</li> </ul>	one exemplar of the prompts for each task in this	1400	
1384		section, as well as the system messages for collabor-	1401	
1385		ation. Our full prompts and formulations can be	1402	
1386		accessed in our attached code repository.	1403	
1387	<ul style="list-style-type: none"> <li>• TAT-QA (Zhu et al., 2021): A QA dataset aiming to stimulate the progress of research</li> </ul>			
1388				

Dataset	N	Example
StrategyQA (Cobbe et al., 2021)	2,290	Hydrogen’s atomic number squared exceeds number of Spice Girls?
CommonsenseQA (CSQA; Talmor et al., 2019)	1,221	A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?
OpenBookQA (Mihaylov et al., 2018)	500	In which location would a groundhog hide from a wolf?
ARC-c (Clark et al., 2018)	1,172	An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase in rotation?
BoolQ (Clark et al., 2019)	3,270	Wherever a gene exists on a DNA molecule, one strand is the coding strand (or sense strand), and the other is the noncoding strand (also called the antisense strand, anticoding strand, template strand or transcribed strand). Question: does it matter which DNA strand is transcribed?

Table 6: Examples from commonsense & factual reasoning datasets used in this work.

Dataset	N	Example
Date Understanding	250	Yesterday was April 30, 2021. What is the date today in MM/DD/YYYY?
Object Counting	250	I have an apple, three bananas, a strawberry, a peach, three oranges, a plum, a raspberry, two grapes, a nectarine, and a blackberry. How many fruits do I have?
Penguins in a Table	146	A Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. How many penguins are more than 5 years old?
Colored Objects	250	On the desk, you see a bunch of items arranged in a row: a gold textbook, a purple puzzle, a teal necklace, and a silver pencil. How many non-gold items do you see to the right of the pencil?
Repeat Copy	32	Repeat the word cat four times. After the second time, also say the word meow.

Table 7: Examples from symbolic reasoning datasets used in this work.


Dataset	N	Example
FinQA (Chen et al., 2021b)	1,147	Question: what percentage of total facilities as measured in square feet are leased? Text: unresolved staff comments not applicable. properties as of december 26 , 2015 , our major facilities consisted of : ( square feet in millions ) united states countries total owned facilities. 30.7 17.2 47.9 leased facilities ... Table: ( square feet in millions )   unitedstates   othercountries   total owned facilities1   30.7   17.2   47.9 leased facilities2   2.1   6.0   8.1 total facilities   32.8   23.2   56.0",
ConvFinQA (Chen et al., 2022)	421	Question: what were the total accumulated other comprehensive losses in 2015? Text: accumulated other comprehensive losses : pmi’s accumulated other comprehensive losses , net of taxes , consisted of the following: reclassifications from other comprehensive earnings the movements in accumulated other comprehensive losses and the related tax impact , for each of the components above ... Table: ( losses ) earnings ( in millions )   ( losses ) earnings 2015   ( losses ) earnings 2014   2013 currency translation adjustments   \$ -6129 ( 6129 )   \$ -3929 ( 3929 )   \$ -2207 ( 2207 ) pension and other benefits   -3332 ( 3332 )   -3020 ( 3020 )   -2046 ( 2046 ) derivatives accounted for as hedges   59   123   63 total accumulated other comprehensive losses   \$ -9402 ( 9402 )   \$ -6826 ( 6826 )   \$ -4190 ( 4190 )
TAT-QA (Zhu et al., 2021)	718	Question: What is the change in Other in 2019 from 2018? Text: Sales by Contract Type: Substantially all of our contracts are fixed-price type contracts. Sales included in Other contract types represent cost plus and time and material type contracts. On a fixed-price type contract, we agree to perform the contractual statement of work for a predetermined sales price ... Table: 2014   2014   Years Ended September 30,   2014 2014   2019   2018   2017 Fixed Price   \$ 1,452.4   \$ 1,146.2   \$ 1,036.9 Other   44.1   56.7   70.8 Total sales   \$1,496.5   \$1,202.9   \$1,107.7


Table 8: Examples from semi-structured reasoning datasets used in this work.


---


**Q:** Mark took a test yesterday that consisted of 75 questions. He completed the test at a rate of 5 questions per hour. Today, he took another test of 100 questions at the same rate. If Mark had 8 hours to complete the first test and 6 hours to complete the second one, how many questions did he leave incomplete?

---

 Mark completed the first test at a rate of 5 questions per hour with a total of 8 hours, so he completed  $5 * 8 = 40$  questions on the first test. He completed the second test at the same rate of 5 questions per hour with a total of 6 hours, so he completed  $5 * 6 = 30$  questions on the second test. Therefore, Mark completed a total of  $40 + 30 = 70$  questions from both tests. Since the first test consisted of 75 questions and Mark completed 40 questions, he left  $75 - 40 = 35$  questions incomplete in the first test. Similarly, the second test consisted of 100 questions and Mark completed 30 questions, so he left  $100 - 30 = 70$  questions incomplete in the second test. Therefore, Mark left a total of  $35 + 70 = 105$  questions incomplete.

 Mark completed the first test at a rate of 5 questions per hour, and he had 8 hours to complete it. So he completed  $8 * 5 = 40$  questions. He completed the second test at the same rate, 5 questions per hour, and he had 6 hours to complete it. So he completed  $6 * 5 = 30$  questions. Therefore, Mark completed a total of  $40 + 30 = 70$  questions. Since the first test had 75 questions and Mark completed 70 questions, he left  $75 - 70 = 5$  questions incomplete. Therefore, Mark left 5 questions incomplete.



 Jerry's analysis is mostly correct, but there is a small mistake in the final calculation. Let's break down the solution step by step: From the given information: Mark completed the first test at a rate of 5 questions per hour, and he had 8 hours to complete it. So he completed  $8 * 5 = 40$  questions on the first test. He completed the second test at the same rate, 5 questions per hour, and he had 6 hours to complete it. So he completed  $6 * 5 = 30$  questions on the second test. Therefore, Mark completed a total of  $40 + 30 = 70$  questions. However, the mistake was made in the next step: Since the first test had 75 questions and Mark completed 40 questions, the number of questions he left incomplete on the first test is  $75 - 40 = 35$  questions. Similarly, the second test had 100 questions and Mark completed 30 questions, so he left  $100 - 30 = 70$  questions incomplete on the second test. To find the total number of questions left incomplete, we need to add the number of questions left incomplete on both tests:  $35 + 70 = 105$ . So, the answer is Mark left 105 questions incomplete.

 You are correct, Tom. I apologize for the mistake in my previous calculation. Thank you for pointing it out and providing the correct solution. To recap: Mark completed  $8 * 5 = 40$  questions on the first test. He completed  $6 * 5 = 30$  questions on the second test. Therefore, Mark completed a total of  $40 + 30 = 70$  questions. Since the first test had 75 questions and Mark completed 40 questions, the number of questions he left incomplete on the first test is  $75 - 40 = 35$  questions. Similarly, the second test had 100 questions and Mark completed 30 questions, so he left  $100 - 30 = 70$  questions incomplete on the second test. To find the total number of questions left incomplete, we need to add the number of questions left incomplete on both tests:  $35 + 70 = 105$ . So the answer is Mark left 105 questions incomplete.

---

The discussion ends when both two players reach an agreement: answer=105. Tom and Jerry got the correct answer of 105 instead of Jerry's incorrect answer of 5.

---

Table 9: An illustrative example of the discussion process of *Corex* with a group between Tom  and Jerry  over a mathematical problem from GSM8K.

---

**Q:** Uriah's book bag is getting too heavy for him. He needs to remove 15 pounds from it. His comic books weigh  $1/4$  pound each and his toys weigh  $1/2$  pound each. If he removes 8717992 comic books, how many toys does he need to remove?

---



```
def solution():  
    """Uriah's book bag is getting too heavy for him. He needs to  
    ↳ remove 15 pounds from it. His comic books weigh 1/4 pound  
    ↳ each and his toys weigh 1/2 pound each. If he removes  
    ↳ 8717992 comic books, how many toys does he need to  
    ↳ remove?"""  
    weight_to_remove = 15  
    comic_book_weight = 1/4  
    toy_weight = 1/2  
    comic_books_removed = 8717992  
    total_weight_removed = comic_books_removed * comic_book_weight  
    toys_removed = total_weight_removed / toy_weight  
    result = toys_removed  
    return result
```



The code provided for the question is incorrect. The code erroneously calculates the number of toys Uriah needs to remove by dividing the total weight of the removed comic books by each toy's weight, neglecting the total 15 pounds Uriah aims to remove. It overlooks the weight left to be removed after the comic books are out, leading to an incorrect answer. Here's the corrected code:

```
def solution():  
    """Uriah's book bag is getting too heavy for him. He needs to  
    ↳ remove 15 pounds from it. His comic books weigh 1/4 pound  
    ↳ each and his toys weigh 1/2 pound each. If he removes  
    ↳ 8717992 comic books, how many toys does he need to  
    ↳ remove?"""  
    weight_to_remove = 15  
    comic_book_weight = 1/4  
    toy_weight = 1/2  
    comic_books_removed = 8717992  
    comic_book_weight_removed = comic_book_weight *  
    ↳ comic_books_removed  
    weight_remaining = weight_to_remove - comic_book_weight_removed  
    toys_removed = weight_remaining / toy_weight  
    return toys_removed
```


---


Table 10: An illustrative example of the Review mode (one round) of Corex among two agents over a mathematical problem from GSM-Hard.


---


**Q:** Melanie is selling 4 gumballs for eight cents each. How much money can Melanie get from selling the gumballs?

---


 Melanie is selling 4 gumballs for eight cents each. That means each gumball costs  $8/4 = 2$  cents. So from selling the 4 gumballs, Melanie can get  $4 \times 2 = 8$  cents. The answer is 8 cents.

 Melanie is selling 4 gumballs for 8 cents each. So she can get  $4 \times 8 = 32$  cents from selling the gumballs. The answer is 32 cents.

 Melanie is selling 4 gumballs for eight cents each. So she can get  $4 \times 8 = 32$  cents from selling the gumballs. The answer is 32 cents.

 Melanie is selling 4 gumballs for 8 cents each. So for each gumball, she can get  $8 / 4 = 2$  cents. Therefore, she can get a total of  $2 \times 4 = 8$  cents from selling the gumballs. The answer is 8 cents.

---

 Candidates 2 and 3 are accurate in their computations, emphasizing that each gumball costs 8 cents and that Melanie would earn a total of 32 cents from the sale of all gumballs. In contrast, Candidates 1 and 4 mistakenly deduce the price per gumball, thereby providing an incorrect total earning. Therefore, the most faithful answers are provided by candidate 2 and candidate 3 with a confidence score of 1.

---

Table 11: An illustrative example of the retrieve process of *Corex* among five agents (four candidates and one retriever) over a mathematical problem from SingleEQ.

---

**CSQA**

**Q:** What do people use to absorb extra ink from a fountain pen?

**Options:** (A) shirt pocket (B) calligrapher's hand (C) inkwell (D) desk drawer (E) blotter

**A:** The answer must be an item that can absorb ink.

Of the above choices, only blotters are used to absorb ink.

So the answer is (E).

**CSQA COMPLEXITY-BASED**

**Q:** The building could accommodate many people. The entrance hall alone was impressive, being wide enough to admit a hundred shoulder to shoulder. But the building was owned by a billionaire and used only for his personal entertainment. How would you describe this place?

**Answer Choices:** (A) convention center (B) public building (C) large building (D) school (E) town hall

**A:** The answer should be somewhere that is not for the good and convenience of the people. Of the above choices, large building is the only neutral description of a building owned by a billionaire. So the answer is (C).

---

**STRATEGYQA**

**Q:** Could Brooke Shields succeed at the University of Pennsylvania?

**A:** Brooke Shields went to Princeton University.

Princeton University is about as academically rigorous as the University of Pennsylvania.

Thus, Brooke Shields could also succeed at the University of Pennsylvania.

So the answer is yes.

**STRATEGYQA COMPLEXITY-BASED**

**Q:** Can jackfruit be used as a weapon?

**A:** Jackfruit is the fruit of a species of plant called the Jacktree.

Jackfruit can weigh up to one hundred and twenty pounds.

Jackfruit is covered in little spikes.

Jackfruit can be thrown or flung at an enemy.

A weapon is a thing that is used to cause bodily harm.

Thus, Jackfruit can be used as a weapon

So the answer is yes.

---

Table 12: Examples of few-shot exemplars for CSQA and StrategyQA. 6-shots are adopted for CSQA and 7-shots are adopted for StrategyQA.



---

**OPENBOOKQA**

**Q:** As you look deeper into a Marbel you can see

**Options:** (A) the future (B) minut defects (C) colors (D) the other side

**A:** Marbel is not transparent, so you can not see the other side.

Marbel does not necessarily have multiple colors.

You will see minut defects.

So the answer is (B).

---

**ARC-C**

**Q:** George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?

**Options:** (A) dry palms. (B) wet palms. (C) palms covered with oil. (D) palms covered with lotion.

**A:** Dry surfaces will more likely cause more friction via rubbing than other smoother surfaces,

hence dry palms will produce the most heat.

So the answer is (A).

---

**BOOLQ**

**Q:** is a cello and a bass the same thing?

**A:** The cello is played sitting down with the instrument between the knees, whereas the double bass is played standing or sitting on a stool.

So the answer is no.

---

Table 13: Examples of few-shot exemplars for OpenBookQA, ARC-c, and BoolQ respectively. 4-shots are adopted for all three tasks.

---

### **MATHEMATICAL REASONING: NL PROMPTS**

**Q:** Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left?

**A:** Olivia had 23 dollars. 5 bagels for 3 dollars each will be  $5 \times 3 = 15$  dollars.

So she has  $23 - 15$  dollars left.

$23 - 15$  is 8. So the answer is 8.

---

### **MATHEMATICAL REASONING: PAL PROMPTS**

**Q:** Olivia has \$23. She bought five bagels for \$3 each. How much money does she have left?

```
# Q: Olivia has $23. She bought five bagels for $3 each. How much money  
↪ does she have left?
```

```
# solution in Python:
```

```
def solution():  
    """Olivia has $23. She bought five bagels for $3 each. How much money  
    ↪ does she have left?"""  
    money_initial = 23  
    bagels = 5  
    bagel_cost = 3  
    money_spent = bagels * bagel_cost  
    money_left = money_initial - money_spent  
    result = money_left  
    return result
```

---

### **MATHEMATICAL REASONING: COMPLEXITY-BASED PROMPTS**

**Q:** Angelo and Melanie want to plan how many hours over the next week they should study together for their test next week. They have 2 chapters of their textbook to study and 4 worksheets to memorize. They figure out that they should dedicate 3 hours to each chapter of their textbook and 1.5 hours for each worksheet. If they plan to study no more than 4 hours each day, how many days should they plan to study total over the next week if they take a 10-minute break every hour, include 3 10-minute snack breaks each day, and 30 minutes for lunch each day?

**A:** Let's think step by step.

Angelo and Melanie think they should dedicate 3 hours to each of the 2 chapters,  $3 \text{ hours} \times 2 \text{ chapters} = 6 \text{ hours}$  total.

For the worksheets they plan to dedicate 1.5 hours for each worksheet,  $1.5 \text{ hours} \times 4 \text{ worksheets} = 6 \text{ hours}$  total.

Angelo and Melanie need to start with planning 12 hours to study, at 4 hours a day,  $12 / 4 = 3$  days.

However, they need to include time for breaks and lunch. Every hour they want to include a 10-minute break, so  $12 \text{ total hours} \times 10 \text{ minutes} = 120$  extra minutes for breaks.

They also want to include 3 10-minute snack breaks,  $3 \times 10 \text{ minutes} = 30 \text{ minutes}$ .

And they want to include 30 minutes for lunch each day, so  $120 \text{ minutes for breaks} + 30 \text{ minutes for snack breaks} + 30 \text{ minutes for lunch} = 180 \text{ minutes}$ , or  $180 / 60 \text{ minutes per hour} = 3$  extra hours.

So Angelo and Melanie want to plan 12 hours to study + 3 hours of breaks = 15 hours total.

They want to study no more than 4 hours each day,  $15 \text{ hours} / 4 \text{ hours each day} = 3.75$

They will need to plan to study 4 days to allow for all the time they need.

So the answer is 4.

---

Table 14: Examples of few-shot exemplars for mathematical reasoning tasks. 8-shots are adopted for all the datasets covered in this paper when using CoT, CoT-SC, and ComplexCoT. 3-shots are adopted when using PAL.

---

**DATE UNDERSTANDING: NL PROMPTS**

**Q:** Today is Christmas Eve of 1937. What is the date 10 days ago in MM/DD/YYYY?

**Options:**

- (A) 12/14/2026
- (B) 12/14/1950
- (C) 12/14/2007
- (D) 12/14/1937
- (E) 07/14/1938
- (F) 12/14/1988

**A:** Let's think step by step.

If today is Christmas Eve of 1937, then today's date is December 24, 1937.

10 days before today is December 14, 1937, that is 12/14/1937. So the answer is (D).

---

**DATE UNDERSTANDING: PAL PROMPTS**

**Q:** 2015 is coming in 36 hours. What is the date one week from today in MM/DD/YYYY?

```
# solution using Python:
# Q: 2015 is coming in 36 hours. What is the date one week from today in
  ↳ MM/DD/YYYY?
# If 2015 is coming in 36 hours, then today is 36 hours before.
today = datetime(2015, 1, 1) - relativedelta(hours=36)
# One week from today,
one_week_from_today = today + relativedelta(weeks=1)
# The answer formatted with %m/%d/%Y is
one_week_from_today.strftime('%m/%d/%Y')
```

---

**COLORED OBJECTS: NL PROMPTS**

**Q:** On the nightstand, there is a red pencil, a purple mug, a burgundy keychain, a fuchsia teddy bear, a black plate, and a blue stress ball. What color is the stress ball?

**Options:** (A) red (B) orange (C) yellow (D) green (E) blue (F) brown (G) magenta (H) fuchsia (I) mauve (J) teal (K) turquoise (L) burgundy (M) silver (N) gold (O) black (P) grey (Q) purple (R) pink

**A:** Let's think step by step.

According to this question, the color of the stress ball is blue. So the answer is (E).

---

**COLORED OBJECTS: PAL PROMPTS**

**Q:** On the nightstand, there is a red pencil, a purple mug, a burgundy keychain, a fuchsia teddy bear, a black plate, and a blue stress ball. What color is the stress ball?

```
# Generate Python3 Code to solve problems
# Q: On the nightstand, there is a red pencil, a purple mug, a burgundy
  ↳ keychain, a fuchsia teddy bear, a black plate, and a blue stress ball.
  ↳ What color is the stress ball?
# Put objects into a dictionary for quick look up
objects = dict()
objects['pencil'] = 'red'
objects['mug'] = 'purple'
objects['keychain'] = 'burgundy'
objects['teddy bear'] = 'fuchsia'
objects['plate'] = 'black'
objects['stress ball'] = 'blue'
# Look up the color of stress ball
stress_ball_color = objects['stress ball']
answer = stress_ball_color
```

---

Table 15: Examples of few-shot exemplars for Date Understanding and Colored Objects. For CoT, CoT-SC and PAL, 3-shots are adopted for both tasks.

---

**OBJECT COUNTING: NL PROMPTS**

**Q:** I have a blackberry, a clarinet, a nectarine, a plum, a strawberry, a banana, a flute, an orange, and a violin. How many fruits do I have?

**A:** Let's think step by step.

We first identify the fruits on the list and include their quantity in parentheses:

- blackberry (1)
- nectarine (1)
- plum (1)
- strawberry (1)
- banana (1)
- orange (1)

Now, let's add the numbers in parentheses:  $1 + 1 + 1 + 1 + 1 + 1 = 6$ . So the answer is 6.

---

**OBJECT COUNTING: PAL PROMPTS**

**Q:** I have a peach, an onion, two garlicks, a nectarine, a yam, a carrot, a potato, a head of broccoli, a cabbage, a banana, and a stalk of celery. How many vegetables do I have?

```
# solution using Python:
def solution():
    """Q: I have a raspberry, a cat, a rabbit, a mouse, a pig, two snails,
    ↪ a fish, two cows, a snake, a goat, and a duck. How many animals do
    ↪ I have?
    """
    # note: I'm not counting the raspberry as an animal
    animals_to_count = {
        'cat': 1,
        'rabbit': 1,
        'mouse': 1,
        'pig': 1,
        'snail': 2,
        'fish': 1,
        'cow': 2,
        'snake': 1,
        'goat': 1,
        'duck': 1
    }
    num_animals = sum(animals_to_count.values())
    return num_animals
```

---

**REPEAT COPY: NL PROMPTS**

**Q:** Q: say java twice and data once, and then repeat all of this three times.

**A:** Let's think step by step.

java java data java java data java java data

---

**REPEAT COPY: PAL PROMPTS**

**Q:** say java twice and data once, and then repeat all of this three times.

```
# solution using Python:
def solution():
    """Q: say java twice and data once, and then repeat all of this three
    ↪ times."""
    result = []
    tmp = ["java", "java", "data"]
    for i in range(3):
        result.extend(tmp)
    return " ".join(result)
```

---

Table 16: Examples of few-shot exemplars for Object Counting and Repeat Copy. For CoT, CoT-SC and PAL, 3-shots are adopted for Object Counting and 4-shots are adopted for Repeat Copy.

---

### PENGUINS: NL PROMPTS

**Q:** Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. We now add a penguin to the table: James, 12, 90, 12

How many penguins are less than 8 years old?

**Options:** (A) 1 (B) 2 (C) 3 (D) 4 (E) 5

**A:** Let's think step by step.

This question focuses on age. We know the following: Louis is 7 years old, Bernard is 5 years old, Vincent is 9 years old, and Gwen is 8 years old. Now, we add James to this table: James is 12 years old.

The penguins that are less than 8 years old are Louis and Bernard.

There are 2 penguins less than 8 years old. So the answer is (B).

---

### PENGUINS: PAL PROMPTS

**Q:** Here is a table where the first line is a header and each subsequent line is a penguin: name, age, height (cm), weight (kg) Louis, 7, 50, 11 Bernard, 5, 80, 13 Vincent, 9, 60, 11 Gwen, 8, 70, 15 For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height of Bernard is 80 cm. We now add a penguin to the table: James, 12, 90, 12

How many penguins are less than 8 years old?

**Options:** (A) 1 (B) 2 (C) 3 (D) 4 (E) 5

```
"""
Q: Here is a table where the first line is a header and each subsequent
  ↳ line is a penguin:
name, age, height (cm), weight (kg)
Louis, 7, 50, 11
Bernard, 5, 80, 13
Vincent, 9, 60, 11
Gwen, 8, 70, 15
For example: the age of Louis is 7, the weight of Gwen is 15 kg, the height
  ↳ of Bernard is 80 cm.
We now add a penguin to the table:
James, 12, 90, 12
How many penguins are less than 8 years old?
"""
```

```
# Put the penguins into a list.
penguins = []
penguins.append(('Louis', 7, 50, 11))
penguins.append(('Bernard', 5, 80, 13))
penguins.append(('Vincent', 9, 60, 11))
penguins.append(('Gwen', 8, 70, 15))

# Add penguin James.
penguins.append(('James', 12, 90, 12))

# Find penguins under 8 years old.
penguins_under_8_years_old = [penguin for penguin in penguins if penguin[1]
  ↳ < 8]

# Count number of penguins under 8.
num_penguin_under_8 = len(penguins_under_8_years_old)
answer = num_penguin_under_8
```

---

Table 17: Examples of few-shot exemplars for Penguins. 3-shots are adopted for CoT, CoT-SC, and PAL.

---

**FINQA: NL PROMPTS**

Read the following text and table, and then answer a question:

\$ in millions | year ended Dec 2014 | year ended Dec 2013 | year ended December 2012  
fixed income currency and commodities client execution | \$ 8461 | \$ 8651 | \$ 9914  
equities client execution | 2079 | 2594 | 3171 ...  
pre-tax earnings | \$ 4317 | \$ 3929 | \$ 5634

**Q:** what was the percentage change in pre-tax earnings for the institutional client services segment between 2012 and 2013?

**A:** The pre-tax earnings for the institutional client services segment in 2012 was \$ 5634 million, and in 2013 was \$ 3929 million.

The net change in pre-tax earnings was \$ 1705 million, and the percentage change was 30.3%. So the answer is 30.3%.

---

**FINQA: PAL PROMPTS**

**Q:** what percentage of total cash and investments as of dec . 29 2012 was comprised of available-for-sale investments?

Read the following text and table, and then write code to answer a question:

( in millions ) | dec 282013 | dec 292012  
available-for-sale investments | 18086 | \$ 14001 ...  
trading assets | 8441 | 5685  
total cash and investments | \$ 31561 | \$ 26302

```
#Python
available_for_sale_investments_dec_29_2012 = 14001
total_cash_and_investments_dec_29_2012 = 26302
percent_available_for_sale_investments_dec_29_2012 = available_for_sale_investments_dec_29_2012 /
↳ total_cash_and_investments_dec_29_2012
ans = percent_available_for_sale_investments_dec_29_2012
```

---

**TAT-QA: NL PROMPTS**

Read the following text and table, and then answer a question:

The average monthly number of employees (including Executive Directors but excluding third-party contractors) employed by the Group was as follows: — | 2019 | 2018

— | Number | Number  
Customer operations | 370 | 380  
... Corporate | 115 | 130  
Total | 802 | 822

**Q:** What are the categories of employees listed in the table?

**A:** The answer can be found directly in the table above.

So the answer is ['Customer operations', 'Product and technology', 'Corporate'].

---

**TAT-QA: PAL PROMPTS**

**Q:** What are the categories of employees listed in the table?

Read the following text and table, and then write code to answer a question:

The average monthly number of employees (including Executive Directors but excluding third-party contractors) employed by the Group was as follows: — | 2019 | 2018

— | Number | Number  
Customer operations | 370 | 380  
... Corporate | 115 | 130  
Total | 802 | 822

```
#Python
ans = ['Customer operations', 'Product and technology', 'Corporate']
```

---

Table 18: Examples of few-shot exemplars for FinQA and TAT-QA. For CoT, CoT-SC and PAL, 4-shots are adopted for FinQA and 8-shots are adopted for TAT-QA.

---

**CONVFINQA: NL PROMPTS**

Read the following text and table, and then answer the last question in a series of questions:

- | shares available for awards | shares subject to outstanding awards

2009 global incentive plan | 2322450 | 2530454

2004 stock incentive plan | - | 5923147

**Q:** how many shares are subject to outstanding awards is under the 2009 global incentive plan? what about under the 2004 stock incentive plan? how many total shares are subject to outstanding awards? what about under the 2004 stock incentive plan? Question: what proportion does this represent?

**A:** The share subject to outstanding awards under the 2009 global incentive plan is 2530454, and the share subject to outstanding awards under the 2004 stock incentive plan is 5923147. The total share subject to outstanding awards is 8453601.

The proportion is 70.1%. So the answer is 70.1%.

---

**CONVFINQA: PAL PROMPTS**

Read the following text and table, and then answer the last question in a series of questions:

- | shares available for awards | shares subject to outstanding awards

2009 global incentive plan | 2322450 | 2530454

2004 stock incentive plan | - | 5923147

**Q:** how many shares are subject to outstanding awards is under the 2009 global incentive plan? what about under the 2004 stock incentive plan? how many total shares are subject to outstanding awards? what about under the 2004 stock incentive plan? what proportion does this represent?

```
#Python
shares_subject_to_outstanding_awards_2009_global_incentive_plan = 2530454
shares_subject_to_outstanding_awards_2004_stock_incentive_plan = 5923147
total_shares_subject_to_outstanding_awards =
    ↪ shares_subject_to_outstanding_awards_2009_global_incentive_plan +
    ↪ shares_subject_to_outstanding_awards_2004_stock_incentive_plan
proportion =
    ↪ shares_subject_to_outstanding_awards_2009_global_incentive_plan /
    ↪ total_shares_subject_to_outstanding_awards
ans = proportion
```

---

Table 19: Examples of few-shot exemplars for ConvFinQA. 4-shots are adopted for both CoT, CoT-SC and PAL.

---

**DISCUSS PROMPTS**

"Tom" : "You are Tom, a middle school student renowned for exceptional math skills. Your friends often approach you for help with their math homework due to your ability to explain complex concepts in an easy-to-understand manner. Your task here is to meticulously consider the example provided, and draw upon your deep understanding of mathematical concepts to answer the subsequent question. Please explain your solution step by step, demonstrating your thought process clearly.",

"Jerry" : "You are Jerry, a middle school student renowned for exceptional math skills. Your teachers are always impressed with your innovative methods for solving math problems. Your task here is to meticulously consider the example provided, and draw upon your deep understanding of mathematical concepts to answer the subsequent question. Please explain your solution step by step, demonstrating your thought process clearly."

---

**REVIEWER PROMPTS**

You are a dedicated code reviewer and need to review the provided Python code snippet to see whether it correctly reflects the problem it aims to solve.

(1) If you believe the problem solved by the code is not equivalent to the actual problem, please modify the code so it can solve the problem correctly.

(2) If you believe the code is correct for solving the problem, you should return the original code with annotations for readability. You should not only provide NL explanations but also make sure to return the code.

(3) When the code for the given problem is missing, please write the code to solve it.

Here are some examples ...

---

**RETRIEVER PROMPTS**

You are an excellent mathematician. Given the following candidates: ...

Assign a confidence score between 0 and 1 for each candidate on how faithful their answer is and identify the number(s) of the most faithful answer(s).

Please respond in the format: The confidence scores are: [score1, score2, ...]. The most faithful answer(s) is/are provided by candidate number(s) X (and Y, Z,... if applicable). Here are some examples ...

---

Table 20: Examples of prompts and system messages for collaboration.