Collaborative Reasoning: A Two-Agent Framework for Enhanced Performance on Multiple-Choice Question Answering

Idan Bezalel
idanbezalel@mail.tau.ac.il

Eyal Haluts
eyalhaluts@mail.tau.ac.il

Nadav Marciano
nadavm4@mail.tau.ac.il

Sagi Reiner
sagireiner@mail.tau.ac.il

Abstract

This work explores whether collaborative approaches using multiple Large Language Models (LLMs) can outperform single-model methods on complex reasoning tasks. We introduce a novel two-agent prompting framework specifically designed for multiple-choice question answering (MCQA) tasks, where two independently operating LLMs generate answers with detailed explanations for each choice. We implement two aggregation methods to combine these outputs: *Full Explanation Evaluation*, which compares both models' reasoning across all choices, and *Conflict Resolution*, which focuses only on conflicting choices when models disagree. We evaluate our approach on OpenBookQA and both AI2 ARC datasets (ARC-Easy and ARC-Challenge) against strong baseline prompting methods including Chain-of-Thought (CoT), CoT + Self-Refine, and CoT + Self-Consistency using Meta-Llama-3.2-3B-Instruct and Qwen2.5-3B-Instruct models. Our results demonstrate that collaborative methods can approach or exceed the performance of resource-intensive single-model techniques while requiring fewer inference iterations. The framework reveals interesting patterns in inter-model collaboration, with some models consistently benefiting more from exposure to alternative reasoning paths. This work provides important insights into the potential of collaborative AI systems and establishes a foundation for future research into multi-agent frameworks for complex reasoning tasks. The code for conducting the experiments described in the paper is available in this GitHub repository.

1 Introduction

Collaboration often boosts human problem-solving—can the same hold true for AI? This question has become increasingly relevant as Large Language Models (LLMs) demonstrate impressive capabilities across various tasks but continue to face challenges with complex reasoning problems. While LLMs like GPT, LLaMA, and PaLM have shown remarkable proficiency in many downstream tasks without further fine-tuning, they still struggle with complicated science problems that require deep domain knowledge and multi-step reasoning (1; 2; 3). Traditional approaches to improving LLM performance have focused on prompting techniques for single-model settings, such as Chain-of-Thought (CoT) prompting (1) and various decomposition strategies. However, these methods often fail to fully address the limitations in complex reasoning scenarios, where knowledge gaps, factual errors, and computational mistakes persist. This raises an important research question: **Could collaborative multi-model approaches yield better results than single-model methods for complex reasoning tasks?** In this work, we explore whether two models working together on tasks can outperform a single model using Chain of Thought techniques. We introduce a two-agent prompting framework designed specifically for complex multiple-choice question answering (MCQA) tasks. This framework is motivated by the intuition that different models can provide different insights and knowledge, and combining their outputs may lead to improved answer accuracy compared to relying on a single model. Our framework operates in two stages: First, we prompt two independently operating LLMs with the same MCQA task, where each model generates an answer alongside detailed explanations for why each choice is correct or incorrect (Figure 1A). Second, we employ aggregation methods to combine these outputs and produce a final, more accurate prediction. We implement two primary aggregation approaches: (1) *Full Explanation Evaluation*, which allows a decision-making model to compare both models'

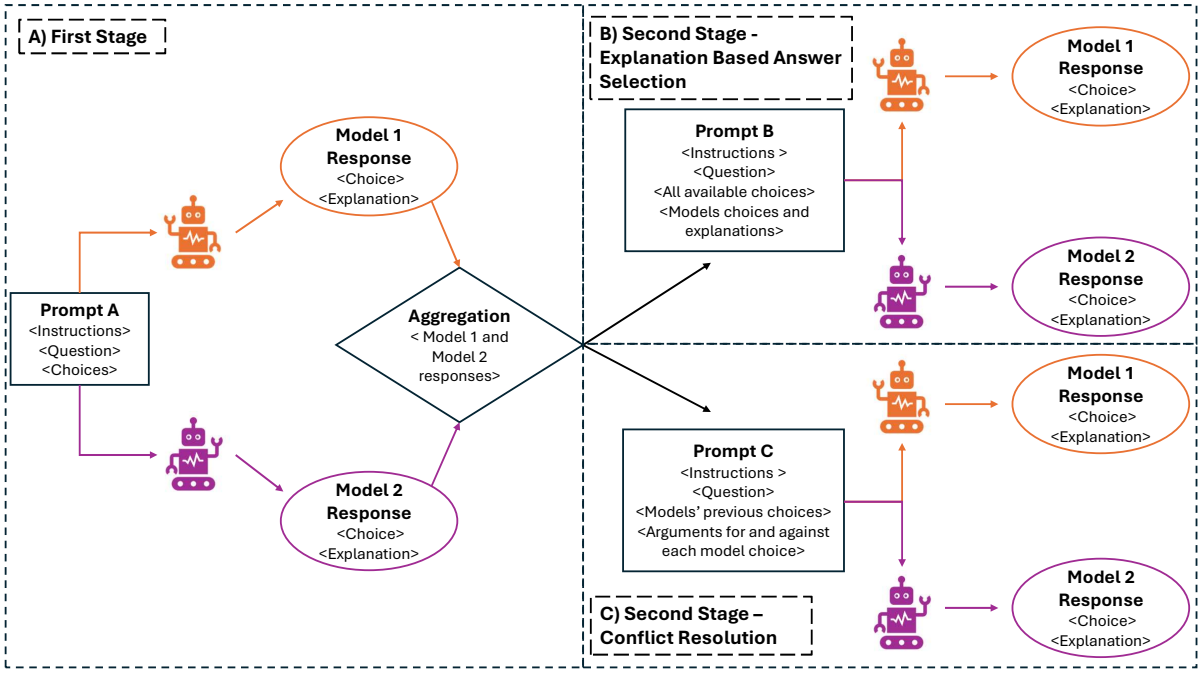

Figure 1: Two agent collaboration framework. Prompts, model responses, and the aggregation step are represented as rectangles, ovals and diamond, respectively. A) First stage. Each model is prompted with a question and possible choices for the correct answer, as well as instructions for how to construct its response. The model's response is constructed from its answer (a label, e.g., 'A') and explanations for why each choice is correct or incorrect (model's reasoning). At the end of this stage the responses from the models are aggregated for use in the second stage. B) Second stage: *Full Explanation Evaluation*. As in the first stage, each model is prompted with a question and all possible choices for the correct answer, but with the addition of each model's previous choice and explanations. The model is instructed to give its response in the same format as in the first stage. C) Second stage: *Conflict Resolution*. Only questions the model's disagreed on (i.e., chose different labels) are given to the models, and the choices are truncated to only the two the models chose in the first stage. The prompt also includes the arguments for and against each choice, taken from the aggregation step in the first round. The model is instructed to give its response in the same format as in the first stage. For answers both model agreed on in the first round, one of the models responses from the first round is given as the response.

reasoning across all choices (Figure 1B), and (2) *Conflict Resolution*, which narrows the decision scope to only conflicting choices when the models disagree (Figure 1C). To evaluate our framework, we conduct experiments on diverse datasets including OpenBookQA (4) and both 'Easy' and 'Challenge' variants of AI2 ARC (5), which represent different levels of complexity and reasoning types. We compare our two-agent framework against strong baseline prompting methods including standard Chain-of-Thought, CoT + Self-Refine, and CoT + Self-Consistency using the models Meta-Llama-3.2-3B-Instruct and Qwen2.5-3B-Instruct (6). Our research makes several key contributions to the field:

1. We quantitatively assess whether collaborative setups yield better results than single-model approaches.

2. We present a novel experimental framework, applied to datasets where iterative inference has proven effective, that evaluates collaborative AI approaches against single-model baselines by testing configurations in which

models assume distinct, complementary roles throughout the problem-solving process.

Our findings provide important insights into the potential of collaborative AI systems.

2 Related Work

Various prompting approaches have been developed to improve LLMs' reasoning capabilities. Some methods prompt LLMs to reason through multiple middle-steps or subproblems (7). Other approaches focus on self-improvement mechanisms, such as self-reflection or self-refinement (8; 9; 10), or planning before problem-solving (11; 12). Despite pushing the boundaries of LLMs' reasoning abilities, these methods still face challenges with complex science problems, often suffering from knowledge gaps, factual errors, and computational mistakes.

Recent work has begun to investigate multi-agent systems built from LLMs. The general idea is to leverage multiple LLMs to tackle complex tasks collaboratively. For example, Guo et al. (13) survey a broad class of LLM-based multi-agent systems, noting that multiple autonomous LLM agents can be specialized with unique skills and engage in joint planning and decision-making. Du et al. (14) explore debate-style interactions between LLM agents. Chen et al. (15) introduce a Collaborative Multi-Agent, Multi-Reasoning-Path framework in which two expert agents with different reasoning paths (e.g., "physicist" vs. "mathematician") jointly solve college-level science problems, followed by a summarizer that reconciles their outputs. Other works simulate human behaviour through multi-agent interactions (16).

Our work builds upon this foundation but specifically investigates whether and how multiple AI agents can collaborate more effectively than single models on complex reasoning tasks. Unlike previous approaches that primarily focus on implementation frameworks, we aim to systematically evaluate the efficacy of collaborative setups compared to single-model baselines and determine the optimal configurations for role-based collaboration in AI systems.

3 Methods

3.1 Two-Agent Prompting Framework

We propose a two-agent prompting framework designed to improve performance on complex multiple-choice question answering (MCQA) tasks. This framework is motivated by the intuition that different models can provide different insights and knowledge, and combining their outputs may lead to improved answer accuracy compared to relying on a single model. In our setup, we prompt two independently operating LLMs, referred to as agents, with the same MCQA prompt. Each model generates an answer alongside a detailed explanation for why each choice is correct or incorrect. These outputs are then used in a second-stage aggregation process that aims to produce a final, more accurate prediction (Figure 1).

3.2 Step 1: Individual Answers with Explanations

To ensure structured and consistent output from both agents, we designed a standardized prompting format that required each model to do the following (Figure 1A):

1. **Select the best answer** from a set of four multiple-choice options.

2. **Explain their decision** by providing a brief explanation for each of the four answer choices, indicating why each one is correct or incorrect.

3. **Format the output** in a strict structure that includes the final selected answer and explanations for each possible choice.

This approach allows us to collect fully reasoned responses from each model in a consistent, readable format. This setup makes it easy to compare the models' reasoning later on, since they both explain why each option is right or wrong-not just the one they picked.

3.3 Method 1: Explanation-Based Answer Selection

The first approach for aggregating the answers is ***Explanation-Based Answer Selection*** (Figure 1B). Here, we provide each model with the following:

- The original question and answer choices.

- The complete outputs of both models, including their final answers and all per-choice explanations.

The model is prompted again with the same instruction as in the initial step, using the same format. This method allows the decision-making model to:

- Compare both models' reasoning per choice.
- Identify strengths and weaknesses in their justifications.
- Make a more informed and possibly more accurate decision.

This method is useful both when the models agree (i.e., select the same answer) and when they disagree, as the decision-making model receives each model's justification for why every possible choice is correct or incorrect, enabling it to evaluate the full reasoning before making a final decision.

3.4 Method 2: Conflict Resolution

The second approach targets situations where the two models disagree on the final answer. Instead of evaluating all options, this method narrows the decision scope to only the two conflicting choices (Figure 1C). The decision-making model is given:

- The original question.
- The two conflicting answers selected by the original models.
- For each of the two answers:
  - **A supporting argument** for this choice, based on the reasoning of the model who picked this answer.
  - **A counter-argument** critiquing the choice, based on the reasoning of the model who picked the conflicting answer.

The model then compares these arguments and selects one of the two choices. For questions where both models agree, one of their initial responses is used as the final answer. This method promotes direct reasoning comparison and reduces distraction from unrelated choices, but it also restricts the model from considering potentially correct answers that neither of the original models selected.

4 Experiments

4.1 Evaluation Datasets

We evaluate our methods using the first **500** questions of three multiple-choice question answering datasets:

- **OpenBookQA**: Questions simulate open-book science exams, requiring a mix of scientific facts and commonsense knowledge.
- **AI2 ARC**:
  - **ARC-Easy**: Simpler science questions requiring basic reasoning.
  - **ARC-Challenge**: Harder questions that stump baseline retrieval and co-occurrence models.

These datasets vary in complexity and reasoning type, allowing us to assess method effectiveness across multiple dimensions.

4.2 Baseline Prompting Methods

For comparison, we implement several established prompting strategies with each model:

- **Chain-of-Thought (CoT)**: Prompts the model to reason step-by-step before answering.
- **CoT + Self-Refine**: The model critiques and revises its CoT response.
- **CoT + Self-Consistency**: Multiple CoT completions are generated (5 or 10); the majority vote determines the final answer.

4.3 Experimental Settings

We use the following language models:

- Meta-Llama-3.2-3B-Instruct
- Qwen2.5-3B-Instruct

Each model is used both as a primary agent and as the decision model in the advanced steps. All models are prompted with identical instructions for consistency and fairness. For evaluation, we focus on accuracy, measured as the percentage of correctly answered questions based on the dataset's gold labels. We do not evaluate explanation quality in this study.

Model	Method	OpenBookQA	ARC-Easy	ARC-Challenge
LLaMA	CoT	76.8	83.2	75.2
	CoT + Self-Refine	69.6	85.6	72.8
	CoT + Self-Consistency	**80.6**	**92.2**	**82.6**
	Individual Answers with Explanations	68.7	85.3	72.2
	Explanation-Based Answer Selection	75.0	86.2	75.2
	Disagreement Resolution	76.0	90.2	77.4
Qwen	CoT	70.0	80.4	68.4
	CoT + Self-Refine	52.8	62.8	54.4
	CoT + Self-Consistency	**79.6**	91.6	**82.4**
	Individual Answers with Explanations	72.3	**93.7**	80.1
	Explanation-Based Answer Selection	75.6	92.2	79.6
	Disagreement Resolution	75.4	91.8	80.4

Table 1: Evaluation of reasoning methods across models and benchmarks. Best results per model are **bolded and underlined**.

5 Results

5.1 Accuracy

The main results are shown in Table 1. For **LLaMA**, we observe a steady improvement from the initial step (*Individual Answers with Explanations*) to the two aggregation methods—*Explanation-Based Answer Selection* and *Conflict Resolution*. These methods achieve accuracy close to that of *CoT + Self-Consistency* with 10 samples, despite requiring fewer iterations. For **Qwen**, we observe a performance increase from the base step to the second-stage methods only on OpenBookQA, while accuracy on ARC-Easy and ARC-Challenge remains similar or slightly lower. On ARC-Easy and ARC-Challenge, where we did not see a performance increase, the results were already close or even higher than those of *Self-Consistency* with 10 samples.

To ensure robustness in the first stage, we ran each model three times on every dataset and observed consistent accuracy across runs. The accuracy reported for *Individual Answers with Explanations* in Table 1 is the average of these three runs. For the subsequent aggregation-based steps, we used one representative run per model. Overall, while our approaches yield slightly lower results compared to *Self-Consistency* with 10 samples, they operate with fewer iterations per question.

5.2 Model Interaction in the Second Stage

We next analyze how often and under what conditions the models changed their answers in the second stage from their original answer in the first stage (Figure 2). To better understand the impact of model interaction, we categorize each question based on first stage answers as follows:

- **Type 1**: Both models answered correctly.
- **Type 2**: Qwen was correct; LLaMA was incorrect.
- **Type 3**: Qwen was incorrect; LLaMA was correct.
- **Type 4**: Both models answered incorrectly.

For each model, we define its 'potential gain' as the percentage of questions it answered incorrectly in the first stage but that the other model answered correctly. This represents the maximum possible improvement in accuracy that could result from fully adopting the other model's correct answers. Formally, the potential gain for Qwen is: $PG(Qwen) = \frac{\#Type3}{500} \times 100$, and the potential gain for LLaMA is: $PG(LLaMA) = \frac{\#Type2}{500} \times 100$.

We also define each model's 'potential loss' as the percentage of questions it answered correctly in the first stage but that the other model answered incorrectly. This represents the maximum possible decrease in accuracy if the model were to be influenced by the other model's incorrect answers. Formally, the potential loss for

Qwen is: $PL(Qwen) = \frac{\#Type2}{500} \times 100 = PG(LLaMA)$ and the potential loss for LLaMA is: $PL(LLaMA) = \frac{\#Type3}{500} \times 100 = PG(Qwen)$.

In other words, a model's potential loss is equivalent to the other model's potential gain. The maximum improvement in accuracy occurs when a model fully utilizes its potential gain while avoiding potential loss, that is, when it adopts the correct answers of the other model, but is not influenced by incorrect ones. In practice, this means the model should change its answer when it was initially wrong and the other model was correct, and retain its original answer when it was initially correct, even if the other model disagreed.

5.2.1 Explanation Based Answer Selection

We observe that in almost all Type 1 (both correct) and Type 4 (both incorrect) cases, the models rarely changed their answers between stages (Figure 2, top). In contrast, answer changes were significantly more frequent in Type 2 and Type 3 questions. This suggests that the changes in the second stage cannot be explained by prompt variation or randomness alone, but are instead driven by the exposure to conflicting answers and justifications. These results lead us to conclude that accuracy changes from the first stage to the second stage are primarily influenced by Type 2 and Type 3 questions. Consequently, we will focus on these types of questions for the remainder of this section.

LLaMA's accuracy improved in the second round across all three datasets. In contrast, Qwen's accuracy improved only on OpenBookQA; for ARC-Easy and ARC-Challenge, its accuracy either remained the same or declined slightly (Table 1).

This trend can be explained by examining the potential gains and their difference for Qwen and LLaMA for each dataset (Figure 2, top). For OpenBookQA, Qwen's potential gain is 12.8% (64 questions), while LLaMA's is 15.6% (78 questions), yielding a relatively small 2.8% advantage for LLaMA. For ARC-Easy, Qwen's potential gain is only 3% (15 questions), compared to 12% (60 questions) for LLaMA - a 9% gap. Similarly, for ARC-Challenge, Qwen's potential gain is 5.8% (29 questions), while LLaMA's is 15.2% (76 questions), resulting in a 9.4% gap. These differences in potential gain help explain the observed changes in accuracy. For LLaMA, the potential gain exceeded its potential loss across all datasets. Moreover, LLaMA was more likely to revise its answer correctly in Type 2 cases (where it was initially wrong) than to make incorrect changes in Type 3 cases. This selective revision behavior contributed to its consistent accuracy improvement.

In contrast, Qwen's potential gain was smaller than its potential loss on all datasets. Nonetheless, Qwen was more likely to revise Type 3 answers to the correct choice than to incorrectly revise Type 2 answers. On OpenBookQA, where the potential gain and loss were relatively close, this favorable revision pattern led to a slight accuracy improvement. However, on the ARC datasets, the gap was too wide: Qwen would have needed to revise Type 3 questions at a much higher rate, and/or revise Type 2 questions at a lower rate than it did to achieve a net gain in accuracy.

To summarize, our findings suggest that the proposed method of having models reconsider their answers based on combined reasoning can improve accuracy, particularly when the two models have comparable baseline performance on a given dataset (i.e., when the number of Type 2 and Type 3 cases is similar). When this balance is skewed, only the model with initially lower accuracy tends to benefit, and the gain may not be enough to surpass the stronger model's first stage performance.

5.2.2 Conflict Resolution

Based on our earlier observation that models rarely change their answers on Type 1 (both correct) and Type 4 (both incorrect) questions, we hypothesized that re-prompting on these question types offers little opportunity to improve accuracy. To focus the interaction more productively, we conducted an additional experiment in which models were prompted only on questions where they initially disagreed - that is, where Qwen and LLaMA selected different answers in the first stage (Figure 1C). Similarly to the case with the *Explanation Based Answer Selection* method, models are more likely to revise questions they answered incorrectly than revise questions they answered correctly (Figure 2, bottom). Furthermore, we observe that in all cases except one (LLaMA on the OpenBookQA dataset), the models were more likely to revise their answers when using the *Conflict Resolution* method compared to the *Explanation Based Answer Selection* method (Figure 2, top and bottom). This increased both gains and losses for each model. For LLaMA, whose poten-

Dataset	Question Type	Count (% of question)	Qwen		LLaMA	
			Correct (% of type)	Incorrect (% of type)	Correct (% of type)	Incorrect (% of type)
OpenBookQA	Type 1	284 (56.8%)	284 (100%)	0 (0%)	284 (100%)	0 (0%)
	Type 2	78 (15.6%) **	71 (91%)	7 (9%)	57 (73%)	21 (27%)
	Type 3	64 (12.8%) *	20 (31%)	44 (69%)	32 (50%)	32 (50%)
	Type 4	74 (14.8%)	3 (4%)	71 (96%)	2 (<1%)	72 (>99%)
ARC-Easy	Type 1	409 (81.8%)	408 (>99%)	1 (<1%)	409 (100%)	0 (0%)
	Type 2	60 (12%) **	45 (75%)	15 (25%)	9 (15%)	51(85%)
	Type 3	15 (3%) *	7 (46%)	8 (54%)	15 (100%)	0 (0%)
	Type 4	16 (3.2%)	1 (6%)	15 (94%)	0 (0%)	16 (100%)
ARC-Challenge	Type 1	326 (65.2%)	326 (100%)	0 (0%)	324 (>99%)	2 (<1%)
	Type 2	76 (15.2%) **	57 (75%)	19 (25%)	22 (29%)	54 (71%)
	Type 3	29 (5.8%) *	13 (45%)	16 (55%)	28 (97%)	1 (3%)
	Type 4	69 (13.8%)	2 (3%)	67 (97%)	2 (3%)	67 (97%)

Dataset	Question Type	Count (% of question)	Qwen		LLaMA	
			Correct (% of type)	Incorrect (% of type)	Correct (% of type)	Incorrect (% of type)
OpenBookQA	Type 2	78 (15.6%) **	64 (92%)	14 (28%)	42 (54%)	36 (46%)
	Type 3	64 (12.8%) *	29 (45%)	35 (55%)	54 (84%)	10 (16%)
ARC-Easy	Type 2	60 (12%) **	40 (67%)	20 (33%)	30 (50%)	30(50%)
	Type 3	15 (3%) *	10 (67%)	5 (33%)	14 (93%)	1 (7%)
ARC-Challenge	Type 2	76 (15.2%) **	56 (74%)	20 (26%)	42 (55%)	34 (45%)
	Type 3	29 (5.8%) *	20 (69%)	9 (31%)	19 (66%)	10 (34%)

Figure 2: Analysis of model interaction in the second stage for the *Explanation Based Answer Selection* method (top) and the *Conflict Resolution* method (bottom). The number of questions from every question type is shown in the count column, and the percentage out of all 500 questions is shown in parentheses (Type 1: both models are correct, Type 2: Qwen correct; LLaMA incorrect, Type 3: Qwen Incorrect; Llama correct, Type 4: both models are incorrect). For each model, the number of correct and incorrect questions in the second stage is shown. Percentages in the correct and incorrect columns are calculated from the total number of questions in each question type. Potential gains and losses for the models are underlined: * - Qwen's potential gain, ** - LLaMA's potential gain. Actual gains and losses are colored in green and red, respectively. Type 1 and Type 4 questions are not shown for the Conflict Resolution methods, because the models were not prompted with Type 1 questions in this method, and for Type 4 questions the models disagreed on, the correct answer was not given as one of the choices. Therefore, all Type 1 questions remained correct, and all Type 4 questions remained incorrect.

tial gain already exceeded its potential loss across all datasets, this led to a further improvement in accuracy - surpassing its performance in both the first stage and *Explanation Based Answer Selection*. For Qwen, however, the higher switch rate had mixed effects: Because Type 2 questions are more frequent than Type 3 questions, the accuracy gain from corrections was offset by new errors, resulting in no net improvement or a slight decline.

In summary, this focused prompting strategy usually led both models to revise their answers more frequently. The model with greater potential gain (LLaMA) benefited from this increased switch rate, while the stronger model (Qwen) saw diminished returns due to a higher rate of unnecessary answer changes.

6 Discussion

Our purposed framework for collaboration between two models to solve complex multiple-

choice question answering (MCQA) tasks focuses on two relatively small models and three datasets, but can be adaptable to different models of varying sizes and applicable to any type of MCQA dataset. The recent rise in the use of large language models (LLMs), particularly for solving complex tasks, highlights the need for efficient approaches in terms of both time and memory. This underscores the importance of small models, which are compact in memory usage and can be deployed on personal devices, at the cost of reduced performance compared to larger models (17). Model collaboration may be the key to achieving performance comparable to that of larger models while being more efficient in both memory and time.

The main results, summarized in Table 1, show that *CoT + Self-Consistency* generally achieves the highest accuracy. In contrast, our methods operate with far fewer iterations, and the overall performance gap remains small—less than five percentage points across all benchmarks, suggesting that model collaboration can approximate the performance of larger sampling with a lower computational cost.

Our experiments show that **LLaMA** consistently improves across all datasets when moving from individual answering to aggregation methods, suggesting it benefits from exposure to alternative reasoning. In contrast, **Qwen** shows improvement only on OpenBookQA, with little or no change on ARC-Easy and ARC-Challenge. This limited gain, and even decline, was not expected and may result from Qwen's lower potential gain and higher potential loss (Figure 2), which decreased the potential benefit from exposure to the other model's answers and reasoning, while increasing the number of cases where incorrect answers were adopted. Our results suggest that both models can benefit from the proposed framework when they have comparable yet distinct capabilities in a specific task.

Our study also raises the question of how to choose the decision-making model. We used one of the original agents as the final evaluator, but the optimal choice may depend on the specific pairing or dataset. Previous studies used an independent model as a 'judge' to resolve models conflict (18), so an interesting question is whether using a similar method can improve results in the cases we tested.

Another observation is that the number of disagreements between models decreased after the second iteration in both aggregation settings. This could indicate a convergence where repeated exposure to the other model's reasoning gradually leads to agreement. Future work could explore additional interaction rounds to test if this convergence continues and whether it contributes to accuracy improvements.

Limitations

This study has three primary limitations. First, we only evaluated a single additional reasoning step and did not explore multi-round interactions, which could offer further insight into model convergence or divergence over time. Second, we limited our evaluation to just two similarly sized models - Qwen and LLaMA. Expanding to larger or more diverse models may result in performance differences and potentially stronger gains from multi-agent collaboration. Third, we tested our framework on only three MCQA datasets. Results may not apply for different datasets or tasks.

7 Conclusion

We present a two-agent prompting framework for MCQA that leverages complementary reasoning from multiple LLMs. Our aggregation methods—*Full Explanation Evaluation* and *Conflict Resolution* exceed or approach single-model baselines using fewer inference iterations. Future work includes dynamic agent selection, richer aggregation strategies, and extension to other models and reasoning tasks.

References

[1] J. Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903, 2023.

[2] X. Ma et al. Query Rewriting for Retrieval-Augmented LLMs. arXiv:2305.14283, 2023.

[3] C. Xu et al. WizardLM: Empowering LLMs to Follow Complex Instructions. arXiv:2304.12244, 2023.

[4] T. Mihaylov et al. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. EMNLP, 2018.

[5] P. Clark et al. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457v1, 2018.

[6] Qwen Team. Qwen2.5: A Party of Foundation Models. Qwen Blog, September 2024.

[7] Y. Yao, Z. Li, and H. Zhao. Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning. arXiv:2305.16582, 2024.

[8] S. Hao et al. Reasoning with Language Model is Planning with World Model. arXiv:2305.14992, 2023.

[9] N. Shinn et al. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366, 2023.

[10] A. Madaan et al. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651, 2023.

[11] D. Gao et al. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. arXiv:2308.15363, 2023.

[12] Z. Sun et al. Recitation-Augmented Language Models. arXiv:2210.01296, 2023.

[13] T. Guo et al. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. arXiv:2402.01680, 2024.

[14] Y. Du et al. Improving Factuality and Reasoning in Language Models through Multiagent Debate. arXiv:2305.14325, 2023.

[15] P. Chen, B. Han, and S. Zhang. CoMM: Collaborative Multi-Agent, Multi-Reasoning-Path Prompting for Complex Problem Solving. arXiv:2404.17729, 2024.

[16] J. S. Park et al. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442, 2023.

[17] F. Wang et al. A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness. arXiv:2411.03350, 2024.

[18] K. Xiong et al. Examining Inter-Consistency of Large Language Models Collaboration: An In-depth Analysis via Debate. Findings of the Association for Computational Linguistics: EMNLP 2023, 2023.