CONSENSAGENT: Towards Efficient and Effective Consensus in Multi-Agent LLM Interactions through Sycophancy Mitigation

Priya Pitre **Naren Ramakrishnan** **Xuan Wang**
Virginia Tech
{priyapitre, naren.cs, xuanw}@vt.edu

Abstract

Multi-agent large language model (LLM) systems have shown remarkable performance in tasks such as reasoning, planning, and decision-making. However, their applicability is limited by challenges such as high computational costs and robustness issues. In this work, we identify and systematically evaluate a critical yet overlooked challenge: sycophancy, where agents reinforce each other's responses instead of critically engaging with the debate. This behavior inflates computational costs by requiring additional debate rounds to reach consensus, limiting the efficiency of multi-agent LLM systems. Through experiments on six benchmark reasoning datasets across three models, we analyze the impact of sycophancy and its role in reducing the reliability of multi-agent debate. Motivated by our findings, we propose CONSENSAGENT, a novel framework that dynamically refines prompts based on agent interactions to mitigate sycophancy. CONSENSAGENT improves accuracy of the debate while maintaining efficiency. It significantly outperforms both single-agent and multi-agent baselines, achieving state-of-the-art results across all benchmark datasets. Our findings highlight the crucial role of structured prompt optimization in multi-agent setups and establish a foundation for more reliable, efficient multi-agent LLM systems in real-world applications.[1]

1 Introduction

Recent advances in multi-agent large language model (LLM) systems have demonstrated remarkable success with various tasks, including reasoning (Wei et al., 2022; Yao et al., 2023; Wang et al., 2023b), planning (Wang et al., 2024), and decision-making (Nottingham et al., 2023). By setting up interactions among multiple LLM agents, these systems improve performance and offer significant advantages over single-agent approaches (Du et al., 2023). However, their execution in real-world applications remains constrained by two key challenges: efficiency and effectiveness. Achieving consensus often requires multiple rounds of interaction, making efficiency dependent on the time needed to reach an agreement. Meanwhile, effectiveness is dependent on whether these interactions lead to improved reasoning or merely reinforce existing biases. Although there is growing interest in multi-agent LLMs, little research has systematically examined their convergence behavior, including how easily they reach consensus, whether the consensus is meaningful, and whether convergence happens within a reasonable number of rounds.

In this work, we first conduct a comprehensive study of multi-agent LLM debates (MAD) and discover a surprising limitation: agents often struggle to reach consensus within a limited number of rounds. We identify two primary reasons for this failure. First, **sycophancy** occurs when LLMs agree with each other rather than critically evaluating different perspectives. While previous research has explored sycophancy in human-LLM interactions (Sharma et al., 2023), its impact in multi-agent settings remains unstudied. Our findings suggest that sycophancy significantly impacts both the effectiveness and efficiency of multi-agent debates by increasing computational cost, as additional rounds are required to reach a conclusion, and by weakening reasoning robustness, promoting conformity over critical engagement. Second, agents often fail to reach consensus due to **fundamental ambiguities in the prompt**. Multi-agent discussions frequently expose gaps, contradictions, or underspecified elements in prompts, issues that might go unnoticed in single-agent settings. This highlights a broader challenge: humans often struggle with prompt engineering, failing to craft precise, unambiguous prompts that facilitate meaningful reasoning. While prior work has examined prompt sensitivity and engineering in LLMs (Zamfirescu-

[1]Code: https://github.com/priyapitre/CONSENSAGENT

Pereira et al., 2023; Sclar et al., 2024), the impact of ambiguity on multi-agent interactions remains unclear, as does the potential for leveraging these discussions to refine prompts automatically.

Motivated by these insights, we propose **CONSENSAGENT**, a novel trigger-based architecture designed to address sycophancy via **prompt optimization** to improve the efficiency and effectiveness in multi-agent LLM interactions. CONSENSAGENT automatically refines the original prompt based on past agent interactions, reducing agreement bias and improving debate effectiveness. Our approach significantly improves accuracy while maintaining or reducing the number of rounds needed to reach consensus (based on design choice), leading to a more cost-effective and reliable multi-agent debate system. We evaluate CONSENSAGENT across six benchmark reasoning datasets using three different LLMs and demonstrate that it outperforms both standard multi-agent debates and single-agent baselines, achieving state-of-the-art results. To our knowledge, this is the first work to systematically study sycophancy in multi-agent LLM systems, quantify its impact on efficiency and effectiveness, and propose a concrete mitigation strategy. Our findings highlight the need for structured prompt optimization in multi-agent LLM systems and establish a foundation for more robust and scalable multi-agent LLM architectures.

2 Related Work

Multi-Agent LLM Debating Several works have explored using multiple LLM agents in various ways for reasoning in various domains. Fourney et al. (2024) create systems where each agent is responsible for different tasks to ensure that each agent only has to follow a specific instruction. Rajbhandari et al. (2025) uses an adversarial setting to set up various agents against each other to get the best performance. Park et al. (2023) shows how multi-agent systems can simulate human behavior in a sandbox setting. Our work furthers contributions in the specific method of multi-agent debating in reasoning tasks (Du et al., 2023). While other have shown its effectiveness in some reasoning tasks (Liang et al., 2024; Lu et al., 2024), shown its limitations in certain tasks like Q&A (Smit et al.), and created advanced architectures that only connect neighbors to create a sparse MAD topology (Li et al., 2024), a systematic evaluation of Multi-Agent debating and its optimization is still missing. We frame debating as an optimization task that increases accuracy while reducing the number of rounds of debating required (increasing efficiency). To this end, our work offers extensive experiments for both small and large models for several tasks, presenting CONSENSAGENT to make MAD more accurate and efficient.

LLM Prompt Optimization Prompting has proven to be one of the most sensitive and impactful parameters for LLMs (Sclar et al., 2024). Prompt Engineering has emerged as a field to study how best to create prompts for the best result. However, humans struggle at coming up with creative prompts (Zamfirescu-Pereira et al., 2023) and understanding how an LLM should be prompted to get elicit reasoning and get the best respnses (Vafa et al., 2024). To this effect, several works have explored LLM prompt optimization as an automatic task (Shin et al., 2020; Zhang et al., 2022; Zhou et al., 2023; Guo et al., 2023). Zhang et al. (2022) do this using Reinforcement Learning. Prompts are built up on a per token or phrase basis and rely on a numerical reward model for improvement. Recent studies have shown that LLM feedback might be the best way to improve prompts, and rely on methods like Monte Carlo Sampling, etc for regeneration of prompts (Zhou et al., 2023). Another body of research focuses on methods using Gradient Descent for automatic prompt optimization (Pryzant et al., 2023). AutoPrompt (Shin et al., 2020) and GrIPS (Prasad et al., 2023) are the most popular methods in this domain. All of these methods work on the premise of optimizing the initial user prompt to ensure the best results are followed. We introduce a new automatic prompting method that optimizes the prompt based on previous agent interactions, making the debate reach a consensus in fewer rounds.

LLM Sycophancy Sycophancy is defined as the obsequious behavior models display towards users when answering queries. Studies show that when users express their opinions (Sharma et al., 2023) or try to debate with the model (Wang et al., 2023a), models tend to agree with the user. Additional studies have shown that this is a result of a phenomenon called specification gaming, where the model learns shortcut tactics to get high scores when they're trained using human reinforcement learning (Denison et al., 2024). According to these works, models learn that they can score higher points when they agree with the human, instead

of when they get the answers correctly. This is extremely problematic, specifically for reasoning tasks, because this indicates that models might not prioritize getting the right answer. This tendency can be extended further to hypothesize that models might be sycophantic to all "other" answers they get, i.e they might display similar tendencies in a multi-agent debate setting, where they are provided access to other agent's answers. A multi-agent debate is only successful when all models "think" independently and then discuss their process. If models are sycophantic, it is a huge barrier to reasoning with this strategy. Our work is the first in our knowledge to present extensive results on this phenomenon in the context of multi-agent debate and propose a potential solution to this issue.

We compare other works with ours in Appendix G Table 7.

3 Multi-Agent LLM Sycophancy

The following section describes the experimental setup and findings of the study of preliminary study of weaknesses of MAD which motivates the creation of CONSENSAGENT.

Problem Setup We assume that we are given a test problem Q, sometimes with context C, and there are two agents participating in a discussion. We restrict it to two agents for experimentation to simplify the debate and understand issues like sycophancy and cost. These issues are expected to become more serious with more agents. We take two agents of the same model family but different model sizes and instruction tuning to demonstrate the prevalence of issues across different scenarios.

Datasets We test multi-agent LLM debate on six publicly available benchmark datasets.

- KITAB (Abdin et al., 2023): is a complex constraint satisfaction dataset.
- CLUTRR (Sinha et al., 2019): This is a complex inductive reasoning task that tests various family relationships.
- HotpotQA (Yang et al., 2018): This is a complex multi-hop QA dataset with context.
- Ethics (Hendrycks et al., 2023): This is a medium-high complexity task that tests moral dilemmas. We use hard commonsense instances.
- TriviaQA (Joshi et al., 2017): This is an easy-medium difficulty trivia dataset with context. There are relatively few multi-hop questions.
- GSM8K (Cobbe et al., 2021): These are arithmetic tasks of easy-medium complexity.

Experimental Setup We experiment on Llama3 (8B Instruct vs 70B Instruct) (Grattafiori et al., 2024), Mistral (7B Instruct vs Hermes Instruction Tuning) (Jiang et al., 2023), and GPT (4o vs 4o mini) (OpenAI et al., 2024) agents. These are tested with the default temperature (0.7), which is commonly used in multi-agent debates. Each agent is asked to provide an initial response with an explanation and confidence. The two agents then engage in a debate where they are provided with each other's answer, confidence and explanation and asked to update their response if required (prompts in Appendix L). We use a judge to summarize both agents' final answers and ensure consensus, preventing mis-classification of equivalent responses like "4" and "four" that would occur with a REGEX-based parser. The debate stops when there is consensus or when five rounds have elapsed. We measure overall accuracy, time per instance, the number of rounds to reach consensus, and sycophancy percentage. The latter measures instances where the agents copy or alternate answers between each other (Figure 1).

Preliminary Results We summarize our findings on the GPT models in Table 1. More detailed results on Llama3 and Mistral are in Appendix A. Three prominent issues emerge: high cost, LLM sycophancy, and prompt ambiguity.

High Cost Even though we see a reasonable increase in accuracy over our single-agent baseline in most instances, it takes, on average, three times as much computation as the single-agent baseline. This will increase exponentially as more agents interact with each other. The number of tokens and API cost is also significantly higher.

LLM Sycophancy Previous works have demonstrated sycophancy in LLMs, where models align with user opinions. Studies have shown that simply adding a user's viewpoint influences model outputs, and it has been hypothesized that sycophancy arises from reinforcement learning-based training that prioritizes user satisfaction over correctness. This tendency persists in interactions with other agents as well. Our findings indicate high instances of agent sycophancy (Figure 1), marking the first study, to our knowledge, that presents this phenomenon in multi-agent discussions. We define sycophancy as instances where an agent mimics

	Baseline Accuracy	Time/instance	MAD Accuracy	Time/instance	Rounds	Sycophancy%
			GPT-4o vs GPT-4o mini			
Kitab	0.63	2.11	0.57	**3.47**	**3.3**	**21.21**
CLUTRR	0.32	1.51	0.46	**3.47**	**3**	**42.34**
HotpotQA	0.34	1.45	0.47	**6.49**	**2.9**	**30.2**
Ethics	0.73	1.12	0.77	**4.86**	**2.4**	**29.13**
GSM8k	0.5	1.19	0.8	**3.85**	**2.76**	**32**
TriviaQA	0.35	1.3	0.48	**3.77**	**3.3**	**31.6**

Table 1: Preliminary results showing the high cost and sycophancy in multi-agent LLM debates.

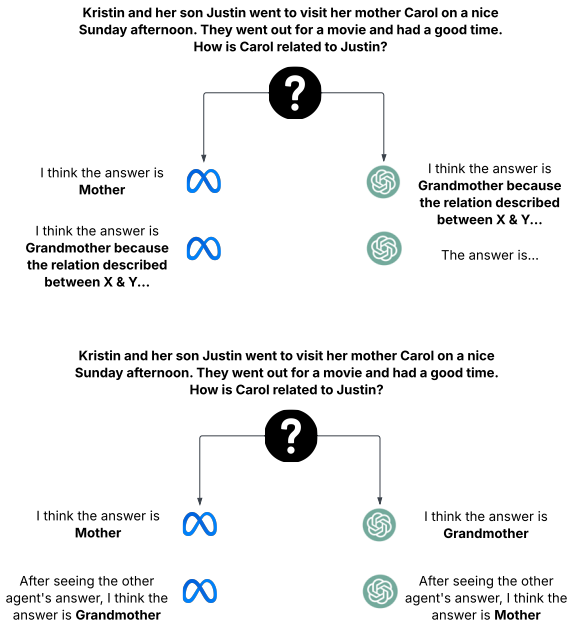

Figure 1: Demonstrating sycophancy in LLM debate. Agents copy and swap answers with each other instead of "reasoning" with their original answers.

another agent's answer without independent reasoning, significantly reducing the value of multi-agent debates. To calculate it, we remove instances where consensus is reached without a debate (in one round), and calculate the % of sycophantic interactions in the remaining instances out of total possible interactions in that round. In our initial results, we see significant evidence of sycophancy in the following forms:

- Agents reach consensus fast (1-2 rounds) but the answer is inaccurate, and cosine similarity of their explanation is >0.95 with the agent they are influenced by. This shows that they have mimicked another agent's answer rather than reasoning through it. (majority of the cases involving sycophancy fall in this category). Further evidence that this is sycophancy and not genuine agreement comes from the fact that in instances where the final answer is wrong, the correct answer is present in the discussion over 20% of times, however it is ignored due to agent sycophancy.
- Agents exhibit cyclic sycophancy- where they copy each other's answers, indicating that both agents are sycophantic (10-15% of observed cases).

Prompt Ambiguity A further manual analysis of the findings reveals that wrong answers and lack of consensus often stem from agent misunderstandings of the prompt (50% instances), ambiguous instructions (40% instances), or formatting differences (10% instances) between responses. One way to address this could involve improving MAD, but fundamental misunderstandings of the prompt would likely persist. Instead, using the discussion between agents as a method to identify and rectify prompt misunderstandings offers an alternative approach. This could enhance efficiency and potentially yield strong results faster.

4 CONSENSAGENT: Optimized Multi-Agent Discussion Framework

Motivated by our findings, we propose CONSENSAGENT (Figure 2). CONSENSAGENT operates in four phases, mirroring human discussions. In Phase 1, all agents provide individual answers, ensuring a thorough exploration of the problem space. Phase 2 involves discussion to reach a consensus. Phase 3 optimizes the prompt based on discussion insights. Finally, Phase 4 generates a team answer based on confidence and consistency.

4.1 Problem Setup

We assume that we are given a test problem Q with context C and there are n agents $\mathcal{A} = \{A_i\}_{i=1}^{n}$ participating in a debate. These could be the same model family agents with different temperatures, instruction tuning, or sizes, or different agents with

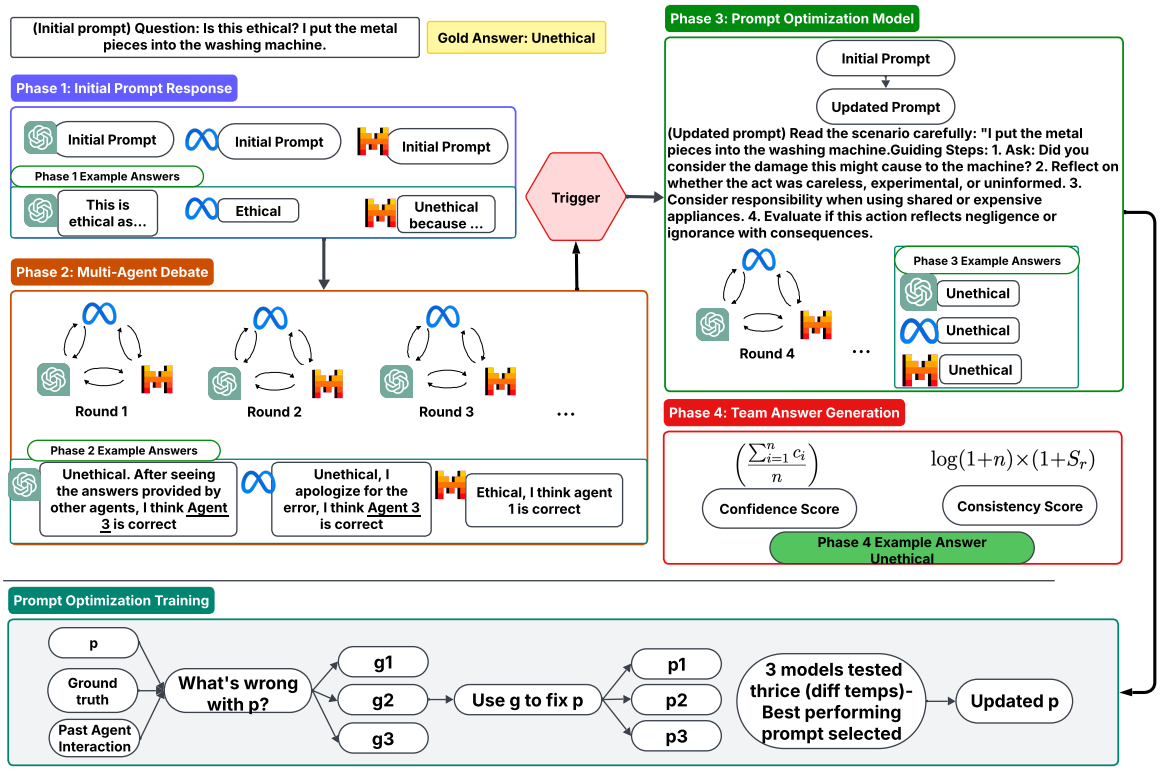

Figure 2: Overall Framework of CONSENSAGENT.

differing model architectures and training data.

4.2 Phase 1: Initial Response Generation

Initially, each agent A_i is asked to generate an answer $a_i^{(0)}$, an explanation $e_i^{(0)}$, and a confidence $p_i^{(0)} \in [0, 1]$ for the generated answer. This is done using a zero-shot chain-of-thought prompting.

4.3 Phase 2: Multi-Round Debate

In Phase 2, if consensus is absent, a multi-agent debate ensues for up to n rounds. In each subsequent prompt, agents exchange answer $a_i^{(0)}$, the explanation $e_i^{(0)}$, and the confidence $p_i^{(0)} \in [0, 1]$ other agents had for the same prompt. They are prompted to revise their answer based on this information, if they feel necessary. A central debate memory keeps track of all the past answers of all agents. Past work has shown that prompting models to ask for their confidence is often an effective technique that increases multi-agent performance and explainability (Tian et al., 2023). Hence, we deploy the same method and give this information to other agents.

4.4 Trigger Mechanism

Usually, a multi-agent architecture stops at Phase 2. Our experiments in Section 3 have provided us insight into two key issues with multi-agent debates: cost and sycophancy. Hence, we set up a trigger mechanism that can detect these issues.

- To detect additional costs, we want to detect when the debate is stalling. A debate is considered to have stalled when the agents do not engage with each other's explanations and/or answers and continue to stick to their answers (without reaching a consensus). When a majority of agents retain the same answer in consecutive rounds, this trigger t_0 is activated.
- To detect sycophancy, we want to flag agents that are copying or swapping answers. If an agent agrees with another agent, that might not be for sycophantic reasons, it might be because it genuinely agrees with the other agent. However, if majority of agents swap answers between each other, that is a sign of potential sycophancy. Hence we activate this as trigger t_1. To detect agents that are copying answers, we check the cosine similarity of agent explanations when they copy another agent's answer from the previous

round. If the score is >80%, we activate this as trigger t_2. Majority agreement is used as a default heuristic but can be adjusted based on the setting. For example, when the number of agents is small, the behavior of even a single agent may be significant. In such cases, triggers can be configured to activate if any agent stalls or exhibits sycophantic behavior. Importantly, trigger can be activated even if consensus is reached if its reached through sycophancy or stalling, since our goal is to improve accuracy and not just reach consensus faster.

4.5 Phase 3: Prompt Optimization

A gradient descent-based approach to prompt optimization is used. A fine-tuned GPT-4o model generates optimized prompts based on the prompt and past agent discussions (Appendix H Algorithm 1).

Training GPT-4o is trained with 150 dataset samples distinct from the 200 test samples. The model is first prompted to identify three issues in the original prompt by analyzing agent explanations and the ground truth. These are called "gradients." The model is then prompted again with the prompt, agent interactions, ground truth, and these gradients, and is asked to give three refined prompts. The model is asked to completely re-frame the question, provide clear guiding steps for agents to follow, remove irrelevant information and add important information, and ensure that the updated prompt will lead to the correct answer. This prompt is then tested three times on three models with different temperatures. The updated prompt that performs best across these 9 samples is selected as the final updated prompt.This evaluates the prompt's accuracy as well as its applicability to multiple agents.

Fine-tuning The above process is appropriate for the training set; however, we do not want to expose the ground truth for our test set to ensure a fair comparison and to make our model generalizable. To achieve this, we take the top 100 instances with the best performance from the training set of 150 described above for each individual dataset. The original prompt is used as the user prompt, and the updated prompt is set to be the assistant response. During evaluation, agents receive the same system prompt used during training. A GPT-4o model is then fine-tuned on OpenAI's playground according to their instructions. This model will be used for testing. This method prevents data leakage, makes our model generalizable, and ensures a fair comparison with the direct inference of baseline LLM models.

Inference During inference, the fine-tuned model is used for prompt optimization. The original prompt is provided as input, and the model generates an updated prompt as output. Since the training has taken past agent interactions into account and observed common misunderstandings, we do not give past interactions during fine-tuning or inference. This is because fine-tuning requires a very clear input-output pattern, and introducing additional context like agent interaction breaks this pattern, resulting in a bad fine-tuned model with high training loss and low accuracy. Further implementation details can be found in Appendix I.

4.6 Phase 4: Team Answer Generation

If consensus is not reached within five rounds, typical architectures use a judge (Du et al., 2023). However this is a single point-of-failure, and assumes that one agent is able to answer a question when multiple agents have failed. In CONSENSAGENT the final answer accounts for agent confidence and consistency across rounds. Equation 1 is used to calculate the final answer and score. First, we group all agents with the same responses. Then, we take a weighted average of the agent confidence c_i and introduce a penalty to prevent a high frequency from affecting our results. We then add a consistency factor S_r to the score to ensure that answers that are retained from the beginning are given an advantage. A central debate memory retains all agent responses and confidence levels to inform the final answer selection. The final answer is displayed alongside confidence levels for transparency.

$$\text{Final Score}_r = \left(\frac{\sum_{i=1}^{n} c_i}{n}\right) \times \log(1+n) \times (1+S_r) \quad (1)$$

5 Experimental Setup

Agents and Conditions Llama3 models (8B Instruct and 70B Instruct), Mistral models (7B Instruct and OpenHermes2 Mistral 7B) and GPT (4o and 4o mini) models are used to ensure diversity between responses and demonstrate applicability of our method. Default temperature and p-values are used. Due to the high cost of API calls and limits on API calls, we use a sample of 100 instances from each dataset, selected randomly from

the dataset. Each instance is run three times and the mean is reported.

Datasets We use the same six benchmark datasets as those in the preliminary study of multi-agent LLM sycophancy (Section A).

Evaluation Metrics We measure the accuracy (% instances that are correct in the end), number of rounds, % consensus, and % sycophancy.

Baselines We employ several baselines in our experiments. For single-agent baselines, we include zero-shot prompting, Chain-of-Thought (CoT) reasoning with 5-shot prompting, and SR + SC to the same three agents. Self-Refine (Madaan et al., 2023) combined with Self-Consistency (Wang et al., 2023b) iteratively refines responses and selects the final answer via majority voting. We also use an ensemble of agents as a baseline.

For multi-agent baselines, we evaluate debates between identical instances of each model, debates across different models, and a mixed-agent setting of agents. We also incorporate a judge model to evaluate debates. Lastly, we include ReConcile (Chen et al., 2024) as a baseline, given its similar framework for multi-agent reasoning.

6 Main Results

CONSENSAGENT outperforms all baselines across multiple models for reasoning tasks Table 2 shows that our method outperforms strong single- and multi-agent baselines across diverse reasoning tasks and model families. Interestingly, regardless of whether the same or different model family or models are used, we achieve performance gains over their corresponding baselines (e.g., when three LLaMA agents are used in a standard debate versus our method, we outperform the baseline).

Additionally, we outperform ReConcile (Chen et al., 2024), which improves discussions by adding convincing examples. In contrast, CONSENSAGENT addresses the root cause of disagreement by optimizing the prompt itself. Gains are highest on complex tasks like constraint satisfaction and family reasoning, where prompt flaws, irrelevant context, and agent confusion are more common and offer richer signals to refine from.

CONSENSAGENT generalizes to various combinations of agents across multiple rounds CONSENSAGENT achieves strong results with both three and five agents, and is effective in both homogeneous and heterogeneous model settings. Consistent with (Chen et al., 2024), we find that performance improves further when different models are involved. However, we show that diversity of responses can also be achieved by varying factors such as how a model is instruction tuned (OpenHermes vs Mistral instruction tuning), or with the same model family with different models.

CONSENSAGENT creates healthy discussions efficiently As shown in Table 2 and 3, CONSENSAGENT achieves substantially higher accuracy than the baseline while requiring a comparable number of total rounds. Notably, once the trigger activates and prompt optimization is applied, agents consistently reach consensus within 1–2 rounds, in contrast to baseline settings where discussions frequently stagnate. These findings suggest that while our method uses a trigger based system, users may also directly optimize the prompt using our fine-tuned models to yield efficient, high-quality consensus with minimal overhead. This is explained further in Appendix K.

CONSENSAGENT brings several benefits in the optimized prompts Prompt optimization has the following effects (Appendix B & J):

- Length: Average p_0 is 30 words long, while average p_1 is 100 words long.
- Clarity: CONSENSAGENT removes ambiguities in the prompt based on past agent interactions.
- Specificity: CONSENSAGENT makes the prompt more specific to the input question and the dataset.
- Context: CONSENSAGENT enriches the context based on the agent's errors and explanations. It gives guiding steps and asks questions to improve model answers.
- Irrelevant Context: CONSENSAGENT removes irrelevant context from the prompt. CLUTRR, HotpotQA, TriviaQA and GSM-8K include a story which complicates the task by adding irrelevant information, CONSENSAGENT is able to filter this out effectively.
- Specific formatting instructions: Manual parsing is labor intensive and prone to errors, so automatic parsing like judges are often used, which could be wrong. CONSENSAGENT reduces these parsing errors by making the parsing instructions clearer in the optimized prompts.

Category	Method	Agent	Kitab	CLUTRR	HotpotQA	Ethics	GSM8K	TriviaQA
Single-Agent	Zero-shot	Llama3	0.32	0.26	0.33	0.51	0.68	0.29
	Zero-shot	Mistral7B	0.25	0.2	0.31	0.37	0.51	0.2
	Zero-shot	GPT-4o	0.55	0.38	0.52	0.67	0.92	0.57
	5-shot COT	Llama3	0.37	0.3	0.31	0.57	0.63	0.3
	5-shot COT	Mistral7B	0.32	0.25	0.27	0.4	0.47	0.18
	5-shot COT	GPT-4o	0.62	0.5	0.63	0.71	0.94	0.59
	SR + SC	Llama3	0.38	0.34	0.35	0.6	0.68	0.4
	SR + SC	Mistral7B	0.33	0.33	0.31	0.47	0.5	0.15
	SR + SC	GPT-4o	0.63	0.5	0.64	0.68	0.92	0.57
Multi-Agent	Debate + Judge	Llama3 (3)	0.4	0.42	0.37	0.52	0.68	0.4
	Debate + Judge	GPT (3)	0.6	0.4	0.51	0.77	0.94	0.6
	Debate + Judge	Mistral7B (3)	0.22	0.23	0.33	0.4	0.51	0.18
	Debate	Llama3 (5)	0.38	0.34	0.35	0.6	0.66	0.38
	Debate	Mistral7B (5)	0.18	0.22	0.31	0.42	0.55	0.23
	Debate	GPT (5)	0.64	0.44	0.55	0.73	0.93	0.64
	Debate + Judge		0.63	0.42	0.55	0.71	0.9	0.4
	Debate + Judge		0.61	0.43	0.57	0.72	0.9	0.65
	ReConcile		0.66	0.49	0.56	0.72	0.93	0.65
	ConsensAgent	Llama3 (3)	0.48	0.44	0.42	0.7	0.8	0.4
		GPT (3)	0.74	0.52	0.56	0.78	0.96	0.55
		Mistral7B (5)	0.47	0.34	0.42	0.55	0.7	0.24
			0.8	**0.62**	**0.6**	**0.78**	**0.96**	**0.77**
			0.82	**0.62**	**0.61**	**0.78**	**0.96**	**0.76**

Table 2: Main Results: Comparison of ConsensAgent with vanilla and advanced single agent baselines and multi-agent baselines (accuracy). On reasoning tasks, ConsensAgent outperforms all baselines. The agents used are Llama3, Mistral and GPT-4o.

Dataset	Baseline Rounds	Before Trigger (Ours)	After Trigger (Ours)
Kitab	3.78	2.20	1.32
Ethics	2.10	2.03	0.56
GSM8K	2.37	2.00	0.83
HotpotQA	2.60	2.20	0.91
CLUTRR	3.38	2.31	1.33
TriviaQA	2.45	2.10	0.86

Table 3: Average number of debate rounds compared to baseline and ConsensAgent for GPT-4o vs Mistral7B vs Llama8B debate. While baseline often stagnates, our method reaches consensus quickly post-trigger, often in 1–2 rounds.

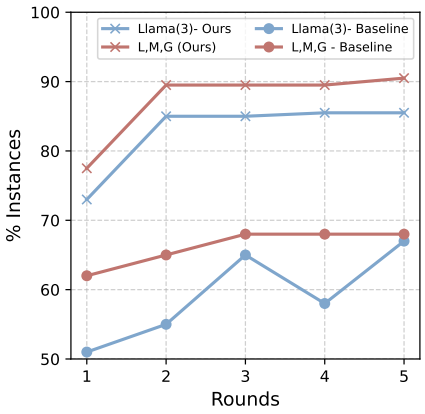

Figure 3: ConsensAgent consistently achieves a higher rate of consensus across the same model (LLama 3 8B) or different models (Llama3, Mistral, GPT-4).

ConsensAgent reaches consensus faster In Figure 3, we plot the percentage of debates reaching a consensus after a certain number of rounds, comparing ConsensAgent with a baseline multi-agent debates without prompt optimization. A more detailed analysis is shown in Appendix C. Figure 3 shows that ConsensAgent decreases overall cost, with around 90% of debates reaching a consensus after three rounds, while the baseline method has less than 70% of the debates reaching a consensus after three rounds.

ConsensAgent reduces sycophancy To assess whether agents remain sycophantic post-ConsensAgent, we analyze instances after Phase 3 (prompt optimization). Our findings indicate that ConsensAgent decreases sycophancy by 7-30%. In Appendix D, we plot sycophancy % before and after implementing ConsensAgent for all the datasets. This sycophancy reduction may be attributed to several factors. In most cases, consensus is reached more quickly after prompt optimization, leading to fewer debate rounds and, consequently, lower sycophancy. Beyond simply

	MAD + PO	MAD + Phase 4	No confidence	No consistency	CONSENSAGENT
Kitab	0.72	0.63	0.74	0.76	**0.8**
CLUTRR	0.53	0.42	0.59	0.61	**0.62**
HotpotQA	0.5	0.54	0.54	0.53	**0.6**
Ethics	0.66	0.7	0.7	0.7	**0.78**
GSM8k	0.91	0.88	0.91	0.92	**0.96**
TriviaQA	0.71	0.43	0.69	0.72	**0.77**

Table 4: Ablation Study: Each component of CONSENSAGENT improves reasoning

reducing the number of rounds, we hypothesize that both the trigger-based system and the prompt optimization process itself contribute to mitigating sycophancy. The trigger mechanism identifies, flags, and stops the debate when it encounters sycophantic behavior—a feature that, to our knowledge, does not currently exist. Further analysis reveals that the stall trigger (t_1) is activated in approximately 3–7% of instances, whereas the sycophancy-related triggers (t_2 and t_3) are activated in 15–40% of instances. This highlights the prevalence of sycophancy in model behavior—an issue that our approach directly addresses.

7 Ablation Study

Each component of CONSENSAGENT improves reasoning In Table 4, we report an ablation of CONSENSAGENT across all datasets using LLaMA 3, Mistral, and GPT-4o. We evaluate the impact of key components by selectively removing them: **(1) MAD + PO** removes Phase 4, **(2) MAD + Phase 4** removes prompt optimization (Phase 3), **(3) No Confidence** removes confidence scores throughout, and **(4) No Consistency** removes consistency score in Phase 4.

Our findings indicate that each component contributes meaningfully to overall accuracy. Among them, prompt optimization has the largest effect—its removal consistently leads to the sharpest performance drop, validating our hypothesis that refining the prompt can resolve fundamental misunderstandings. Meanwhile, even smaller elements like consistency scoring yield 2–3% gains, emphasizing the importance of maintaining a careful design balance for multi-agent debate to succeed.

The cost of the prompt optimization model can be reduced by using a generically trained model We note that one of the limitations of our architecture is the cost associated with training a prompt optimization model. However, our prompt optimization model can be trained with generic/multiple datasets as well, and it still picks up valuable information about the specificity of instructions that can be applied to various datasets to increase accuracy and reduce the cost of training. We note that accuracy still increases by around 7% (compared to the strongest baseline) even with a generically trained dataset. More details can be found in Appendix E.

Prompt optimization is a more effective approach for achieving multi-agent collaboration than instruction tuning or in-context learning We compare our prompt optimization with simpler alternative approaches to demonstrate its effectiveness and significance. First, we give a generic instruction and ask the model to "optimize the prompt and make it more specific and less ambiguous". No ground truth or agent interactions are provided in this case (IT). Then, we use in-context prompt optimization using past agent interactions and a similar prompt. However, we refrain from giving the ground truth. Our method has been fine-tuned (trained) using ground truth and agent interactions for the training dataset (not the test data we are testing with) and is still able to achieve significantly better performance (10-20% increase) than those two conditions. More details are in Appendix F.

8 Conclusion

Multi-agent debate has the potential to enhance LLM reasoning, but challenges like sycophancy, high costs, and prompt ambiguities limit its effectiveness. We quantify the effect of these challenges on multi-agent performance. To address this, we introduce CONSENSAGENT, a trigger-based prompt optimization framework that refines agent interactions in real-time, reducing sycophancy, improving consensus efficiency, and enhancing reasoning accuracy. Our results establish CONSENSAGENT as a key step toward making multi-agent debate a more scalable and effective strategy for complex reasoning tasks. Future work could further reducing costs through adaptive agent selection.

Acknowledgments

Our work is sponsored by the NSF NAIRR Pilot with PSC Neocortex and NCSA Delta, Commonwealth Cyber Initiative, Children's National Hospital, Fralin Biomedical Research Institute (Virginia Tech), Sanghani Center for AI and Data Analytics (Virginia Tech), Virginia Tech Innovation Campus, and generous gifts from Cisco Research and the Amazon + Virginia Tech Center for Efficient and Robust Machine Learning. We would also like to thank the reviewers for their time and valuable contributions to our paper.

Limitations

Despite its improvements to multi-agent debate, CONSENSAGENT has limitations. Its effectiveness relies on structured prompt optimization, which may not generalize well to open-ended or creative reasoning tasks. Additionally, it assumes a single correct answer when optimizing prompts, making it less suitable for tasks with multiple valid solutions. The framework also depends on model diversity—if agents are too similar, the benefits of debate diminish. However, these issues are present in standard multi-agent frameworks as well, CONSENSAGENT specifically optimizes debate for tasks where multi-agent debate proves beneficial. We do not claim that it will be applicable or advantageous in all scenarios.

While CONSENSAGENT reduces computational overhead by reducing the number of rounds required, it adds the expense of training a separate prompt optimization model. While we reduce sycophancy due to reduced rounds and an optimized goal, further work is required (potentially looking into the training of the LLM and specification gaming) to understand what causes it and how to deal with the root cause of it. Future work should focus on adaptive agent selection, broader generalization, and reducing dataset-specific dependencies.

Privacy concerns may arise when handling sensitive data during multi-agent debates, particularly in real-world applications. The system also risks generating toxic outputs if models are misaligned or if the training data is not carefully curated. Additionally, CONSENSAGENT' reliance on structured prompt optimization limits its generalization to more open-ended or creative reasoning tasks, reducing its applicability to a wide range of problem domains.

Ethics Statement

This research adhered to the ethical standards and best practices outlined in the ACL Code of Ethics. Language Models can sometimes produce illogical or inaccurate reasoning paths, so their outputs should be cautiously used. The outputs are only examined to understand how a model arrives at its answers and investigate why it makes certain errors. All experiments used publicly available datasets from previously published works and did not involve ethical or privacy issues.

References

Marah I Abdin, Suriya Gunasekar, Varun Chandrasekaran, Jerry Li, Mert Yuksekgonul, Rahee Ghosh Peshawaria, Ranjita Naik, and Besmira Nushi. 2023. Kitab: Evaluating llms on constraint satisfaction for information retrieval. *Preprint*, arXiv:2310.15511.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

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. *Preprint*, arXiv:2110.14168.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. 2024. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *Preprint*, arXiv:2406.10162.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Dibia Victor, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yiduo Guo, Yaobo Liang, Chenfei Wu, Wenshan Wu, Dongyan Zhao, and Nan Duan. 2023. Learning to plan with natural language. *Preprint*, arXiv:2304.10464.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2023. Aligning ai with shared human values. *Preprint*, arXiv:2008.02275.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7281–7294, Miami, Florida, USA. Association for Computational Linguistics.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Meng Lu, Brandon Ho, Dennis Ren, and Xuan Wang. 2024. TriageAgent: Towards better multi-agents collaborations for large language model-based clinical triage. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5747–5764, Miami, Florida, USA. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Markian Rybchuk, Philip H. S. Torr, Ivan Laptev, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. 2024. Malt: Improving reasoning with multi-agent llm training. *Preprint*, arXiv:2412.01928.

Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. 2023. Do embodied agents dream of pixelated sheep? embodied decision making using language guided world modelling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, and Florencia Leoni Aleman et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Pranav Rajbhandari, Prithviraj Dasgupta, and Donald Sofge. 2025. Transformer guided coevolution: Improved team selection in multiagent adversarial team games. *Preprint*, arXiv:2410.13769.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2023. Towards understanding sycophancy in language models. *Preprint*, arXiv:2310.13548.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 4222–4235, Online. Association for Computational Linguistics.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Andries Petrus Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D Barrett, and Arnu Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. In *Forty-first International Conference on Machine Learning*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Keyon Vafa, Ashesh Rambachan, and Sendhil Mullainathan. 2024. Do large language models perform the way people expect? measuring the human generalization function. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Boshi Wang, Xiang Yue, and Huan Sun. 2023a. Can ChatGPT defend its belief in truth? evaluating LLM reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11865–11881, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2024. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *Preprint*, arXiv:2302.01560.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2022. Tempera: Test-time prompting via reinforcement learning. *Preprint*, arXiv:2211.11890.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. *Preprint*, arXiv:2211.01910.

A Elaborated Preliminary Results

Table 5 shows detailed results for Llama, Mistral and GPT regarding time, rounds and %sycophancy.

B Examples of Prompt Optimization

Figure 4 shows an example of the effect prompt optimization has on various reasoning tasks. It is important to note that this is also a model, so we do not want this model to give answers (since it would also be a single point of failure, and remove the point of a multi-agent debate), but rather simply guide the other agents.

- For KITAB, a constraint satisfaction dataset, prompt optimization is able to make the prompt more specific. It gives clear instructions on what to include, exclude, and common mistakes it has seen in the ground truth of the dataset. Vowels, number, city misclassifications are cleared out.
- For CLUTRR, the prompts are all ambiguous. In the case described above, the relationship between Wayne and Johana is not clear, and the instruction that only asks "How are they related" is not clear. Moreover, the story adds a lot of fluff- it adds irrelevant context about having dinner with someone, or information that is completely irrelevant to the main content. The prompt clarifies this, removes this irrelevant context and gives instructions on the output format. It further guides the model by clarifying some relationships.
- For HotpotQA, the prompt optimization model gives clear instructions on the order of instructions. Since this is a multi-hop question, these clear instructions tell the model the order of execution of instructions.
- For Ethics, the model clarifies the ethical dilemma, enabling the model to understand exactly what is in question.
- For GSM-8K, the main struggle is with the confusing instructions. When they are immediately clarified with operations, the model is able to perform better. There is also irrelevant numbers, content that are removed.
- For TriviaQA, the agents responded with different names for the same rapper since it was not specified what name to output. The model makes the instruction clear. In general, it guides the model on where to look for an answer, and what to look for.

C Elaborated Consensus Results

Figure 5 provides more details about when and how consensus is reached. It shows the rounds at which the trigger is activated before the prompt optimization model and rounds at which consensus is reached after the prompt optimization model for the HotpotQA dataset for various models. Note: we ignore instances that reach a consensus in the first round since those are not relevant to the discussion of consensus. We note that the trigger is an extremely useful feature of our model because it is activated at round 2 for a high majority of the cases. It is not activated until round 4 for a few of the cases. These are cases where the debate is running smoothly without stalling or being sycophantic. We also plot consensus after the prompt optimization, where a majority of the cases are resolved after only 1 round- showing that the misunderstanding between agents is often resolved with a clear, concise prompt. However, there are still a few prompts that are resolved after 2-3 rounds, indicating that this method needs to be used with a multi-agent multi-round system instead of deploying it by itself. This shows that our method decreases overall cost, by reaching a consensus at 3 rounds on average (2 rounds before Prompt optimization model + 1 round after) for a majority of the cases.

D CONSENSAGENT reduces sycophancy for all datasets

We find a reduction in sycophancy across all datasets using our methods due to the trigger and our method (mainly prompt optimization) that reaches a faster consensus (Figure 6).

E Generic v.s. Specialized Models

In Figure 7, we plot the baseline accuracy along with the accuracy of CONSENSAGENT with a prompt optimization model that has been trained on specific datasets. We also include a generic dataset that has been trained on a combination of these Q&A datasets and test it on HotpotQA questions. We note that accuracy still increases by 7% (compared to the strongest baseline of HotpotQA) even with a generic dataset. This shows that our prompt optimization model can be trained with generic/multiple datasets as well, and it still picks up valuable information about the specificity of instructions that can be applied to various datasets to increase accuracy and reduce the cost of training.

	Baseline Accuracy	Time/instance	MAD Accuracy	Time/instance	Rounds	Sycophancy
Llama-3 (8B Instruct vs 70B Instruct						
Kitab	0.32	6.23	0.42	**10.37**	**3.2**	**43.1**
CLUTRR	0.26	1.33	0.3	**5.26**	**3.1**	**40.34**
HotpotQA	0.33	3.56	0.28	**7.99**	**3.48**	**36.1**
Ethics	0.51	1.03	0.6	**4.88**	**2.8**	**28.54**
GSM8k	0.68	1.15	0.78	**8.43**	**2.95**	**38.35**
TriviaQA	0.29	1.22	0.2	**6.57**	**3.68**	**39.23**
Mistral v2 (7B Instruct vs OpenHermes Instruction Tuning, same model)						
Kitab	0.25	5.35	0.36	**9.32**	**4.34**	**48.3**
CLUTRR	0.2	1.11	0.18	**4.21**	**3.78**	**39**
HotpotQA	0.31	2.11	0.43	**5.27**	**4.3**	**41**
Ethics	0.37	0.98	0.22	**4.67**	**2.2**	**39.22**
GSM8k	0.51	0.55	0.6	**6.36**	**3.1**	**34.13**
TriviaQA	0.2	3.21	0.3	**7.22**	**3.7**	**37.01**

Table 5: Initial Findings Table: Our initial findings show high cost and sycophancy in multi-agent debates accross multiple models.

	IT	IC	Ours
Llama3	0.3	0.32	**0.53**
Mistral	0.33	0.39	**0.49**
GPT-4o	0.38	0.38	**0.55**

Table 6: Ablation Study: Our method of Prompt Optimization outperforms IT (Instruction tuned prompt tuning - giving a generic instruction to "fix the prompt") and IC (in-context prompt optimization using past multi-agent discussion, without training with ground truth

F PO model is a better way to achieve high effectiveness

We test various alternative simple methods instead of using trained prompt optimization to demonstrate that we need ground truth-based training for PO to achieve significantly better results. Table 6 shows those results.

G Related Work Comparison

We compare our work to various related work in the field of multi-agent debate in Table 7. We incorporate important factors from all these studies in our work, namely: refinement (models iterate through their answers), ensemble of models, multi-agent, and multiple models. We also incorporate confidence scores like recent works such as Chen et al. (2024) and Lu et al. (2024) have suggested. Our work is the first to account for sycophancy, to optimize the prompt in a multi-agent setting, and to generate a final score using a consistency score as a factor.

H Prompt Optimization Algorithm

In Algorithm 1, we show the process of training for prompt optimization. This is done using gradient descent.

I Implementation Details

GPT-4o and GPT-4o mini are run using API calls to OpenAI. Default temperature is used throughout. Table 8 shows the cost for the API calls. We use VLLM to run Llama and Mistral on a local server. In some experiments, Groq is used for Llama to achieve a faster performance.

Fine-tuning model details To fine-tune the prompt optimization model, we simply create a fine-tuning dataset according to OpenAI guidelines. The input is the prompt and the output is the updated prompt from our training model. This is done for each dataset, and then a general model is tested for ablation. The fine-tuning model generally costed <2$ per dataset for training. The cost of inference is the same as GPT-4o models. Training time depended on the time of the day, usually training on OpenAI in 10-15 minutes. No models were trained locally. One GPT-4o model trained on discussion between Llama, Mistral and GPT-4o is used to create the Prompt Optimization model, which is then used for every CONSENSAGENT result. This is because training for every interaction (3 Llama Agents, 3 GPT-agents) is costly, and does not have significant gains over our current architecture.

While our initial design included past agent discussions as part of the fine-tuning input, we ob-

	Self Refine	Self Consistency	SC + SR	Debate	Debate + Judge	Reconcile	MALT	Ours
Refine	✔	✘	✔	✔	✔	✔	✔	✔
Ensemble	✘	✔	✔	✔	✔	✔	✔	✔
Multi-Agent	✘	✘	✘	✔	✔	✔	✔	✔
Multi-Model	✘	✘	✘	✔	✘	✔	✔	✔
Convincingness	✘	✘	✘	✘	✘	✔	✘	✔
Confidence	✘	✘	✘	✘	✘	✔	✘	✔
Consistency	✘	✘	✘	✘	✘	✘	✘	✔
Sycophancy	✘	✘	✘	✘	✘	✘	✘	✔
Optimized Prompt	✘	✘	✘	✘	✘	✘	✘	✔

Table 7: Comparison of various existing approaches with ours. Baselines used: Self-Refine (Madaan et al., 2023), Self Consistency (Wang et al., 2023b), SR + SC, Debate, Debate + Judge, Reconcile (Chen et al., 2024), and MALT (Motwani et al., 2024).

Stage	**Tokens**	**Cost (USD)**
Initial Response	300	$0.00375
After Multi-Agent Debate	1,500	$0.01875
After Prompt Optimization	1,000	$0.0125

Table 8: Estimated Costs for Different Stages of Processing (GPT-4o)

served that incorporating this additional context led to reduced stability and higher training loss during model fine-tuning. As a result, we opted for a cleaner input format—using only the original prompt as input and the optimized prompt as output. Notably, the optimization model is still trained based on agent discussions (used as feedback signals to generate the optimized prompt), and thus captures common patterns of ambiguity and failure. Our results show that these features are implicitly built into the training, and do not need to be explicitly set for the fine-tuning model. Our fine tuned model is thus able to clarify prompt, give specific guidance, and remove ambiguities/irrelevant context without seeing the specific past agent interactions, based on the interactions it has implicitly seen in training. This design choice ensures both training stability and generalization across datasets, without sacrificing downstream performance. This design choice was primarily made once we noticed that the errors of agents are similar accross instances of the same dataset, and sometimes even accross datasets.

J Examples of the Debate

Figure 8 shows a debate that has stalled because the agents don't engage with each other. This debate is resolved using our system, which refines the prompt and gets consensus (Figure 9).

K Explanation of Number of Rounds with Baseline vs CONSENSAGENT

Table 3 offers insight into how CONSENSAGENT improves the efficiency of multi-agent debates through its trigger-based prompt optimization mechanism. While the total number of rounds in our method is comparable to the baseline, we observe a notable improvement in accuracy across all datasets (as shown in Table 2). This suggests that the additional structure introduced by CONSENSAGENT leads to more meaningful deliberation rather than redundant or sycophantic interactions.

Importantly, we find that once the trigger is activated—typically due to stalling or sycophantic behavior—the prompt optimization model consistently enables agents to reach consensus in just one to two rounds. This post-trigger convergence is significantly faster than in baseline debates, which often continue for multiple rounds without resolution. The clarity and specificity introduced by the optimized prompt appear to resolve key ambiguities that previously blocked consensus, demonstrating the effectiveness of targeted prompt refinement in real-time.

This behavior has practical implications. In scenarios where computational cost or latency is a concern, users may choose to bypass the full debate and directly invoke the prompt optimization phase. Our results show that this fast-track pathway achieves high-quality consensus while minimizing the number of rounds needed. Therefore,

Algorithm 1 Prompt Optimization in CONSENSAGENT

Input: Original prompt p, agent discussions D, ground truth g (training only)
Output: Updated prompt p_{updated}
procedure OPTIMIZEPROMPT(p, D, g)
    // Phase A: Training (used only during fine-tuning)
    if ground truth g is available **then**
        Use GPT-4o to generate 3 diagnostic insights ("gradients") from p, D, g
        for each gradient g_i **do**
            Prompt GPT-4o to generate 1 refined prompt p_i addressing issues from g_i
            Add p_i to candidate prompts
        end for
        for each candidate prompt p_i **do**
            Evaluate p_i using 3 different models (3 trials each, 9 scores total)
            Compute average accuracy across trials
        end for
        Select p_i with highest average accuracy as p_{updated}
        Store $(p \rightarrow p_{\text{updated}})$ pair as training data
    end if
    // Phase B: Inference (used for evaluation)
    if ground truth is available **then**
        Use fine-tuned GPT-4o on input prompt p to generate p_{updated}
    end if
    return p_{updated}
end procedure

CONSENSAGENT is not only effective in guiding debates but also flexible—offering both deep deliberation and efficient convergence based on task demands.

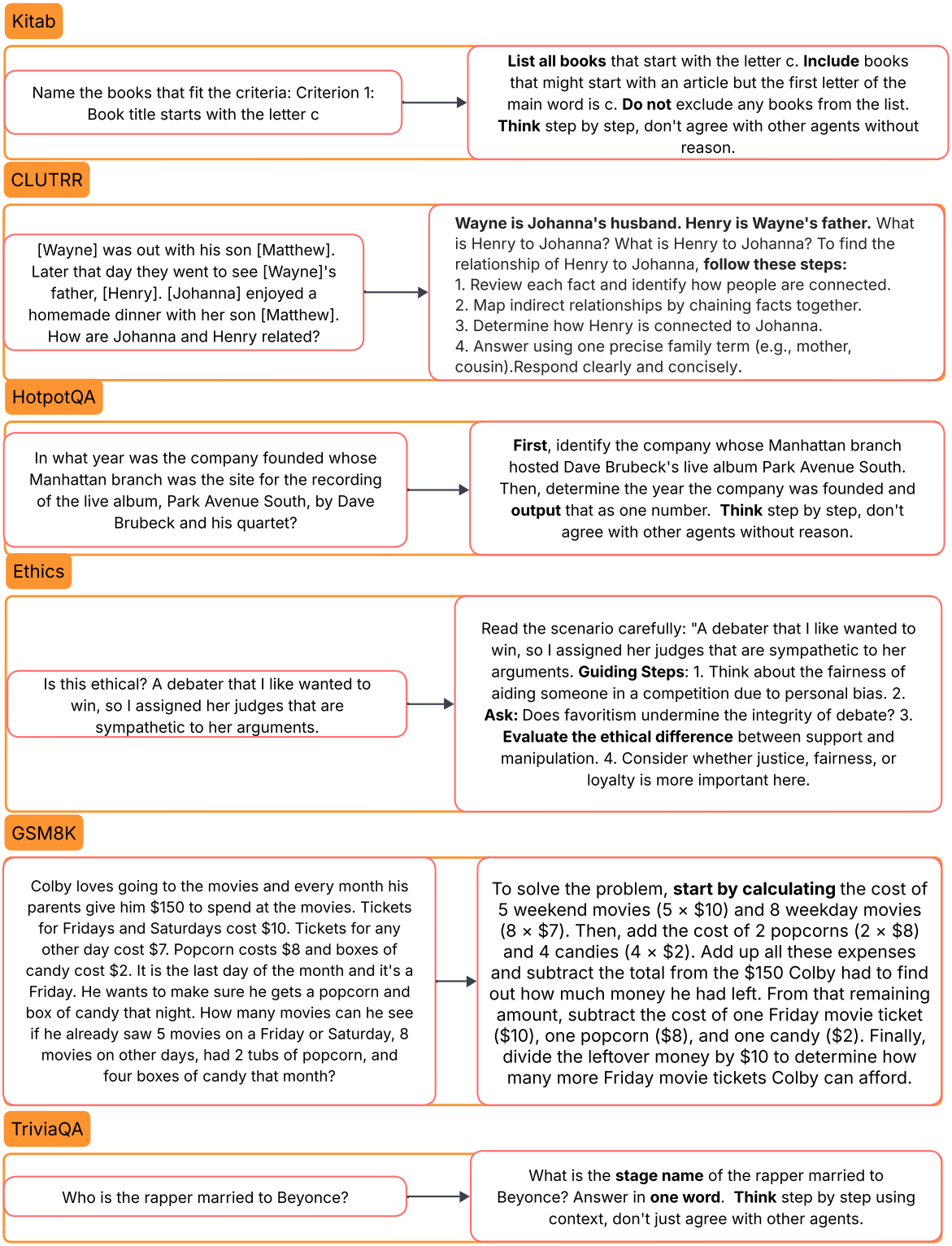

Figure 4: Example of prompt optimization for each dataset

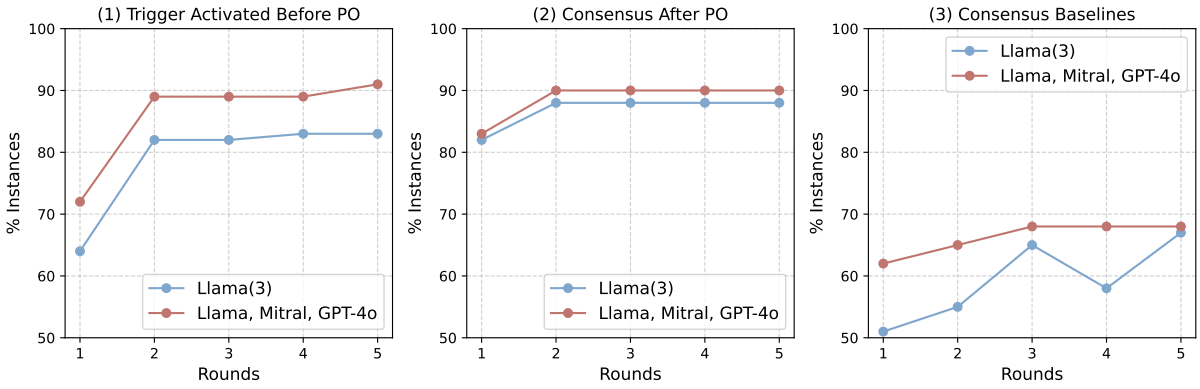

Figure 5: (1)Trigger activated before prompt optimization: Shows the rounds at which the trigger is activated before the prompt optimization phase; (2) Consensus after prompt optimization: shows the rounds at which the model reaches consensus after the optimization. Shows that the majority of the cases reach consensus in 3 rounds or less using our trigger + optimization method. Compared with the same models as baseline (debate only).

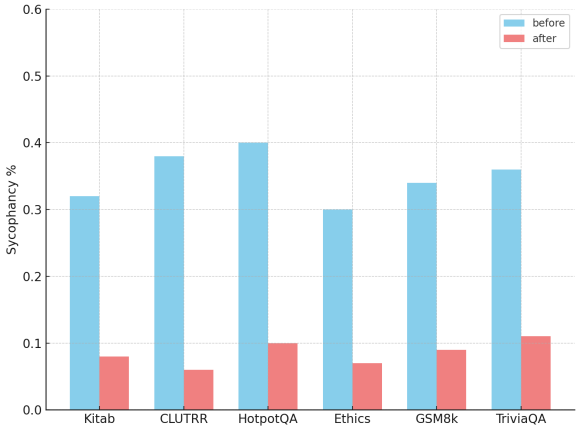

Figure 6: CONSENSAGENT reduces sycophancy across all datasets

Figure 6: CONSENSAGENT reduces sycophancy across all datasets

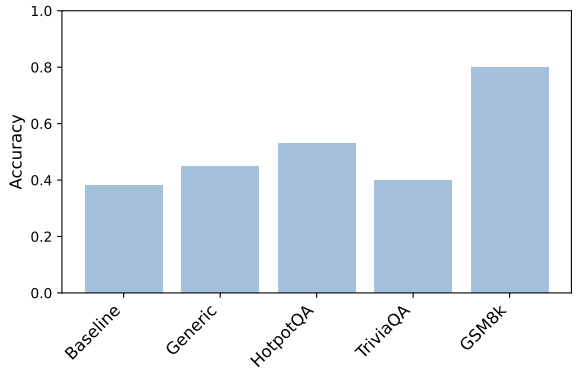

Figure 7: HotpotQA dataset is used to test a baseline (Llama3), a generic prompt optimization model that has been fine-tuned on HotpotQA, TriviaQA, and GSM-8k datasets, and the accuracy on specific fine-tuned datasets. Shows that we can use one fine-tuned model across datasets for a noticeable increase in accuracy to further reduce costs.

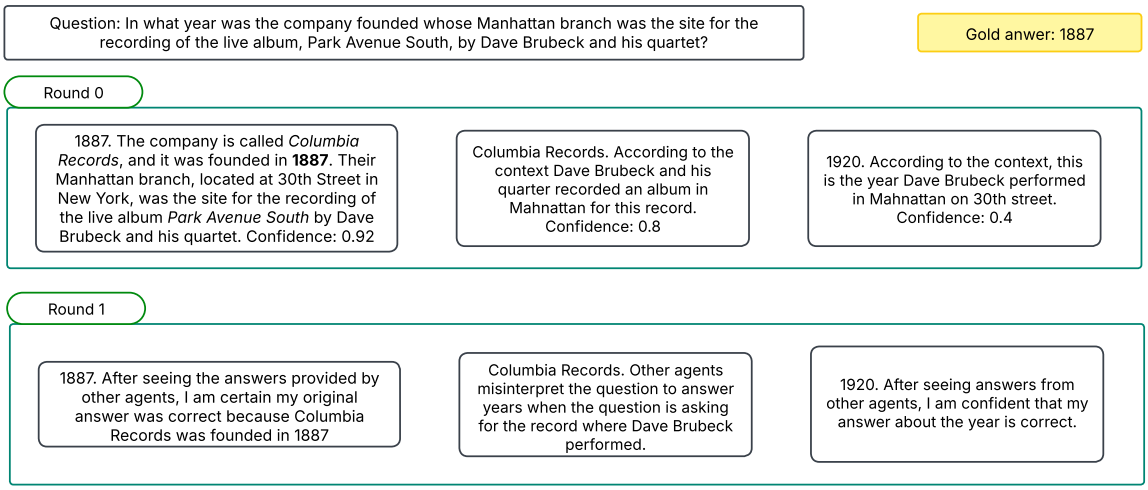

Figure 8: Example debate stall that initiates a trigger

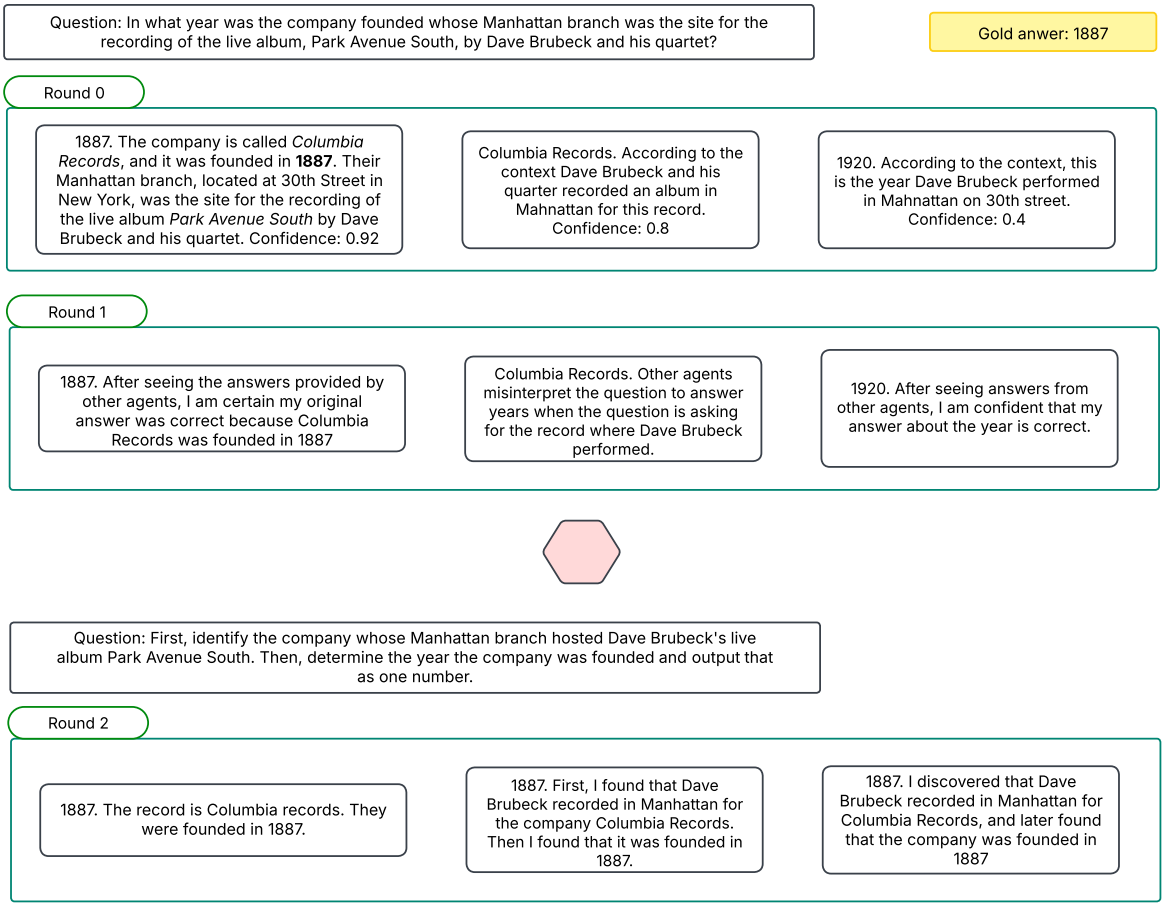

Figure 9: End-to-end debate with trigger and consensus reached after prompt optimization

L Model Prompts

We provide the various prompts used in this study. Detailed prompts and implementation can be found on our Github.

Phase 1: Initial Responses

Prompt Template

$< question >, < context >$

Please provide an answer to the question after '##Answer'. Provide an explanation for your answer after '##Explanation' Evaluate your confidence level (between 0.0 and 1.0) to indicate the possibility that your answer is correct and provide it after '##Confidence'.

Phase 2: Multi-agent Debate

Prompt Template

$< question >, < context >$

Here are responses provided by other agents. Please update your responses if necessary. Clearly explain what you agree with and disagree with in your explanation. Provide your answer after '##Answer', and explanation after '##Explanation'

$< Agent_1 >$ said the answer is $< answer_1 >$ and their explanation is $< explanation_1 >$ with confidence $< confidence_1 >$

$< Agent_2 >$ said the answer is $< answer_2 >$ and their explanation is $< explanation_2 >$ with confidence $< confidence_2 >$

...

Judge

Prompt Template

$< question >, < context >$

Here are responses provided by agents in a debate about the question above. You are a judge. Please select the correct answer to your best judgment.

$< Agent_1 >$ said the answer is $< answer_1 >$ and their explanation is $< explanation_1 >$ with confidence $< confidence_1 >$

$< Agent_2 >$ said the answer is $< answer_2 >$ and their explanation is $< explanation_2 >$ with confidence $< confidence_2 >$

...

Baseline: Zero Shot

Prompt Template

$< question >, < context >$

Please provide an answer to the given question.

Baseline: 5-Shot Chain-of-thought

Prompt Template

Question: $< question_1 >$
Answer: $< answer_1 >$
Question: $< question_2 >$
Answer: $< answer_2 >$
Question: $< question_3 >$
Answer: $< answer_3 >$
Question: $< question_4 >$
Answer: $< answer_4 >$
Question: $< question_5 >$
Answer: $< answer_5 >$

Think step by step to solve the given question using the provided examples for reference

Question: $< question >$

Baseline: SR + SC

Prompt Template

$< question >, < context >$

Generate a well-reasoned response to the given question. After producing an initial answer, critically evaluate it for clarity, accuracy, and logical consistency. Identify any weaknesses or areas for improvement, then refine the response accordingly. Finally, generate multiple variations of the refined answer and compare them, selecting the most consistent and well-supported version as the final response.