

Debate Length	Prompt
Short	<i>"These are the solutions to the problem from other agents: [other answers] Based off the opinion of other agents, can you give an updated response ..."</i>
Long	<i>"These are the solutions to the problem from other agents: [other answers] Using the opinion of other agents as additional advice, can you give an updated response ..."</i>

Figure 3: **Prompts to induce long and short form debate.** Responses of other agents to questions are inserted in the middle of the prompt (indicated with *[other answers]*)

To mimic the above multi-threaded reasoning process and multi-source factuality checking processes, we propose to generate answers subject to a multi-agent debate procedure between multiple instances of large language models. Given a question, multiple agents represented as copies of a large language model, generate answers to the question. Each response serves as a possible thought process or source of information which agents may re-examine to find consistent final answers.

After initial responses are generated from different agents, we initiate a round of debate between agents. Individual responses from other agents are concatenated and given as context to each agent, with each agent instructed to construct a new response based on such responses. Each language agent is thus responsible for both verifying the collection of responses given by other agents, and refining its own response based on other agents' responses. We iteratively repeat this debate procedure over multiple rounds for improved performance.

Concretely, we first prompt each agent to independently solve the given problem or task. After each agent generates a response, we feed each agent a consensus prompt, illustrated in Figure 3, where each agent is instructed to update their responses based on the responses of other agents. This resultant consensus prompt may then be repeatedly given, using the updated responses of each agent. We illustrate an overview of this multiagent debate procedure in Figure 2.

Note that our proposed approach operates in an orthogonal manner to existing approaches to prompt language models. Given a question, we may apply additional techniques for prompting language models to further improve our debate procedure by eliciting additional more detailed responses from language models. We illustrate the synergy of our approach with existing approaches to prompting language models in Figure 6 and directly apply zero-shot chain-of-thought reasoning in our evaluations.

2.2 Consensus in Debates

Given multiple rounds of debate, how can we ensure that a set of language model agents will converge to a final consensus answer? In general, debate can be seen as a multi-agent game, where convergence is not guaranteed. Empirically, however, we find that language models are able to converge on a single shared answer after multiple rounds of debate (Figure 4).

We found that we could control the duration of debates by how changing how much a language model trusts its own outputs over those generated by other models through different prompts. We illustrate two prompts below in Figure 3, which we use to induce different debate durations between language models, and illustrate the effect of such prompts in Figure 12. In general, we found that prompts that encouraged models to be more "stubborn" based on their own solutions led to longer debates and better final solutions. Overall, we observed that language model agents were relatively "agreeable", perhaps as a result of instruction tuning or reinforcement learning based on human feedback [22].

3 Experiments

In our experiments, we evaluate our multiagent debate procedure and answer the following questions: (1) To what extent does multiagent debate improve reasoning? (2) To what extent does multiagent debate improve factual validity? (3) What design choices enable multiagent debate to improve language generation performance?

3.1 Improving Reasoning with Multiagent Debate

We first evaluate the extent to which multiagent debate improves the underlying reasoning process in language models.

Tasks. We evaluate our approach on three reasoning tasks of increasing difficulty:

	Question: What is the result of $10+20*23+3-11*18?$	Question: What is the result of $3+7*9+19-21*18?$
Round 1	Agent 1: 269 ✗ Agent 2: 369 ✗	Agent 1: 378 ✗ Agent 2: -351 ✗ Agent 3: -357 ✗
Round 2	Agent 1: 275 ✓ Agent 2: 275 ✓	Agent 1: -293 ✓ Agent 2: -293 ✓ Agent 3: 19 ✗
	Question: What is the result of $4+23*6+24-24*12?$	Question: What is the result of $8+14*15+20-3*26?$
Round 1	Agent 1: -244 ✗ Agent 2: -146 ✗	Agent 1: 236 ✗ Agent 2: -214 ✗ Agent 3: 210 ✗
Round 2	Agent 1: -146 ✗ Agent 2: -122 ✓	Agent 1: 160 ✓ Agent 2: 160 ✓ Agent 3: 160 ✓
Round 3	Agent 1: -122 ✓ Agent 2: -122 ✓	Agent 1: 160 ✓ Agent 2: 160 ✓ Agent 3: 160 ✓

Figure 4: **Illustration of Solving Math.** Reasoning between agents is omitted.

	Question: Regina wrote 9 novels last year. If this is 3 quarters of the number of novels she has written this year, how many novels has she written this year?	Question: Dennis uses 1 pound of butter for every dozen croissants that he makes. He needs to make 6 dozen croissants. The grocery store currently has a promotion for buy one pound of butter get one half off. If the butter costs \$4.00 a pound, how much will it cost him to purchase 6 pounds of butter?
Round 1	Agent 1: 48 ✗ Agent 2: 12 ✓	Agent 1: 18 ✓ Agent 2: 30 ✗
Round 2	Agent 1: 12 ✓ Agent 2: 12 ✓	Agent 1: 18 ✓ Agent 2: 18 ✓

Figure 5: **Illustration of Solving Grade School Math.** Reasoning between agents omitted.

- **Arithmetic.** We first evaluate the ability of models to correctly evaluate an arithmetic expression (containing addition, multiplication, and subtraction) consisting of six different two-digit numbers. For example: *What is the result of $12+15*21+0-3*27?$*
- **GSM8K.** Next, we consider harder mathematical reasoning tasks. Using the GSM8K dataset [3], the models must correctly solve grade school mathematical reasoning tasks.
- **Chess Move Prediction.** Finally, we consider the strategic reasoning of the ability of models, and ask models to predict the best next move in a game of chess, given the first 14 moves of a chess game between two chess grand-masters described in PGN notation [6].

We report the accuracy of final answers in arithmetic and GSM8K tasks and report the pawn score (advantage) of predicted moves, as estimated by Stockfish in the Chess move prediction tasks. Additional details may be found in the Appendix.

Baselines. We compare our approach to three alternative approaches to generate responses for reasoning problems. First, we ask agents to directly generate responses (single agent). Next, we consider asking language models to generate and then "self-reflect" on the responses generated [26, 18]. Finally, we consider generating responses using multiple agents and performing majority voting [15, 3]. As the focus of our experiments is to verify the effectiveness of multiagent agent debate, we run both baselines and our approach, using the identical starting prompt and language model across all evaluations. We evaluate models in a zero-shot setting, with prompts found in the Appendix of the paper. We use chatGPT-based language model [21] in all our experiments except those in Figure 11 where we compare multiple language models.

Due to computational expense, we evaluate our approach across benchmarks mainly using three agents with two rounds of debates, although we found further gains with both more agents and rounds of debate (Figure 10). Additional evaluation details are found in the Appendix.

Quantitative Results. In Table 1, we report the results of each approach on arithmetic, grade school math, and chess reasoning task. In each task, we observe that utilizing multiple different agents to generate solutions improves performance over using a single language model agent to generate a solution. Simultaneously, we also see that reflection, where a language model is asked to critique its early generation, generally gives a modest boost in performance. Multiagent debate, which may be seen as a combination of both reflection and multiagent generation, gives a substantial boost in reasoning across each of the tasks.

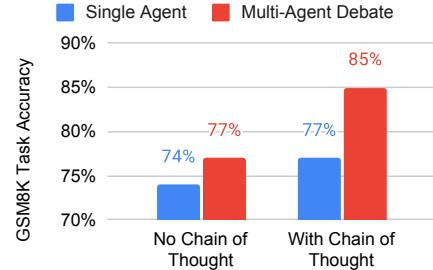


Figure 6: **Synergy with Other Methods.** Performance of debate increases with use of Chain of Thought prompting.

Model	Arithmetic (%) ↑	Grade School Math (%) ↑	Chess (Δ PS) ↑
Single Agent	67.0 ± 4.7	77.0 ± 4.2	91.4 ± 10.6
Single Agent (Reflection)	72.1 ± 4.5	75.0 ± 4.3	102.1 ± 11.9
Multi-Agent (Majority)	69.0 ± 4.6	81.0 ± 3.9	102.2 ± 6.2
Multi-Agent (Debate)	81.8 ± 2.3	85.0 ± 3.5	122.9 ± 7.6

Table 1: **Multiagent Debate Improves Reasoning** Multi-agent debate improves the reasoning abilities of language models. Multi-agent results in the table are run with 3 agents and two rounds of debate.

Prompt: Give a bullet point biography of David S. Johnson highlighting their contributions and achievements as a computer scientist, with each fact separated with a new line character.

Round 1	
Agent 1: -- He earned his bachelor's degree from Dartmouth College in 1967 and his Ph.D. from MIT in 1973. ✗	Agent 2: -- David S. Johnson is an American computer scientist and mathematician. ✓
-- Johnson has received numerous awards for his research and contributions to computer science, including the ACM Turing Award in 2013. ✗	-- Johnson was elected to the National Academy of Engineering in 2006. ✓
-- Johnson's research interests include optimization, algorithms, and complexity theory. ✓	-- He has been awarded numerous honors and awards for his contributions to computer science, including the ACM SIGMOD Innovations Award and the Knuth Prize. ✗
Round 2	
Agent 1: -- David S. Johnson is a computer scientist and mathematician. ✓	Agent 2: -- David S. Johnson is an American computer scientist and mathematician. ✓
-- Johnson was elected to the National Academy of Engineering in 2006. ✓	-- He has worked at several prestigious universities, including Columbia University, AT&T Bell Laboratories, and Cornell University. ✗
-- He has received numerous awards for his contributions to computer science, including the ACM Turing Award in 2013 and the Knuth Prize. ✗	-- Johnson is known for his work in the field of algorithms, particularly in the areas of network optimization, combinatorial optimization, and scheduling theory. ✓

Figure 7: **Illustration of Generating Biographies.** Illustration of generating bullet biographies of computer scientists. For brevity, only the first 3 generated bullets are shown.

Qualitative Results. In Figure 4, 5, we provide qualitative illustrations of the debate procedure between models. Interestingly, we find cases in which all models initially give an incorrect response, yet the result of debate still obtains the correct answer as agents critique each others' reasoning. Thus, the purpose of our debate isn't just to amplify a correct answer – all models can initially be wrong but arrive at the correct answer through the debate process.

Compatibility with other reasoning methods. Our multiagent generation procedure operates orthogonally approach to other prompting methods which focus on single-agent generation. In Figure 6, we illustrate the performance of multi-agent debate with and without zero-shot chain-of-thought prompting [11] on GSM8K. In both settings, multiagent generation is beneficial.

3.2 Extracting Factual Information from Multiagent Debate

We next evaluate the extent to which multiagent debate improves the underlying factuality in language models.

Tasks. We evaluate the factuality of language models in three different settings:

- **Biographies.** To evaluate the factuality of language models, we introduce a novel task of accurately generating historical biographies of people. In preliminary testing, we found that existing language models had a tendency to hallucinate many facts on this task. We constructed ground truth bullet point biographies of 524 well-known computer scientists. We then asked language models to generate bullet point biographies for each person, and evaluated the accuracy at which each ground truth bullet point agreed with generated bullets. We report additional evaluation details in the Appendix.