

# Multiagent Scientific Reasoning Systems

Yilun Du  
May 2026



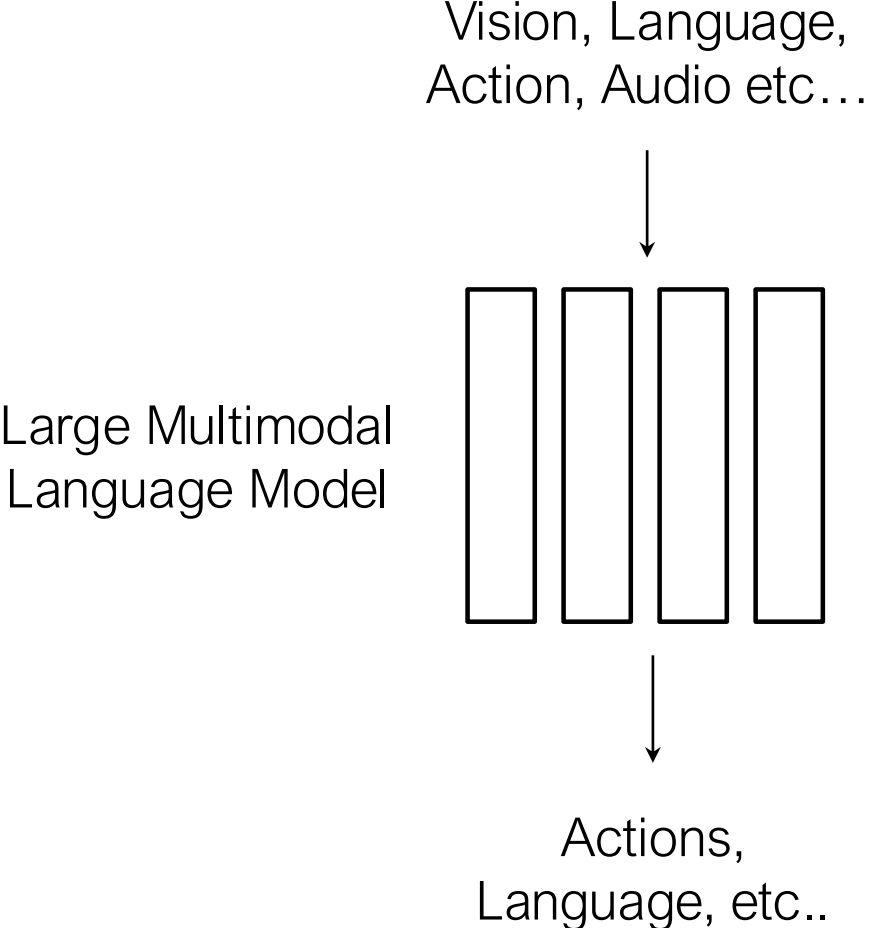
**HARVARD**  
UNIVERSITY



**Kempner**  
INSTITUTE



# Is One Large Monolithic LLM All You Need?

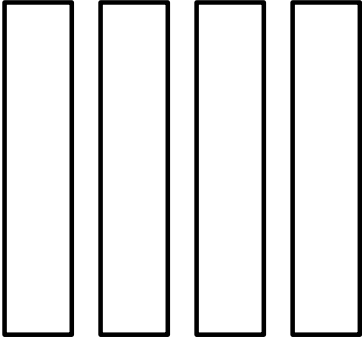


# Is One Large Monolithic LLM All You Need?

Vision, Language,  
Action, Audio etc...



Large Multimodal  
Language Model

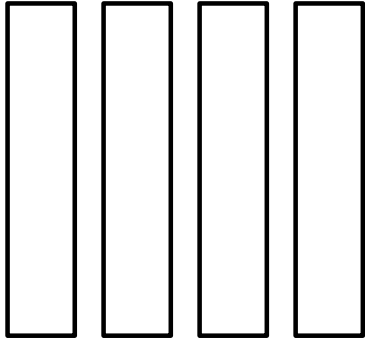


Actions,  
Language, etc..

Difficult to interpret  
and a single model  
is in charge of  
executing  
everything.

# Towards Compositional AI Systems

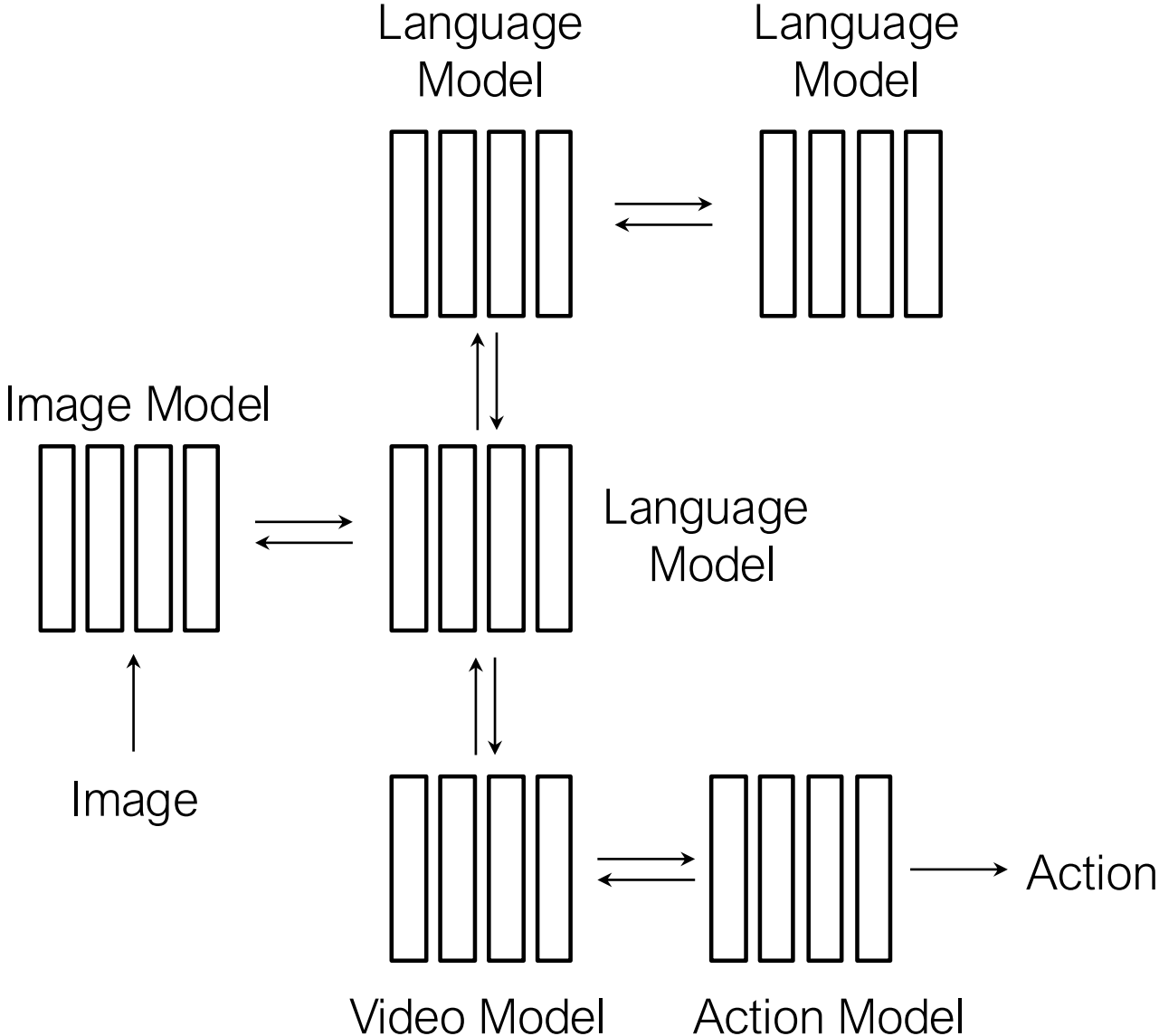
Vision, Language,  
Action, Audio etc...



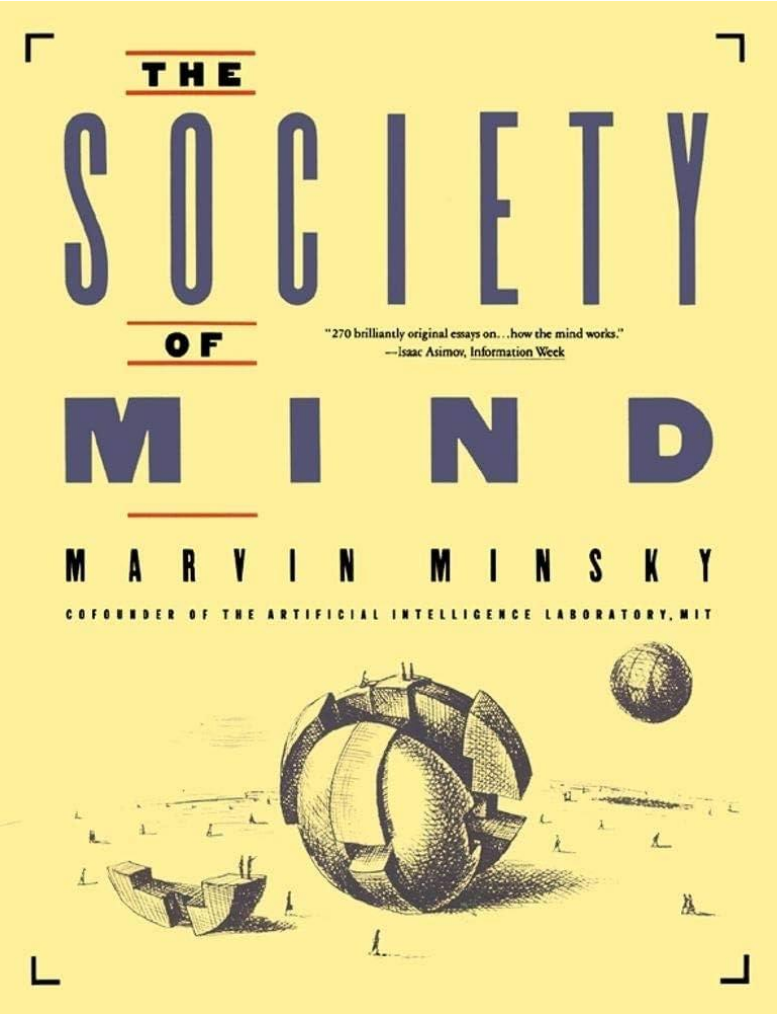
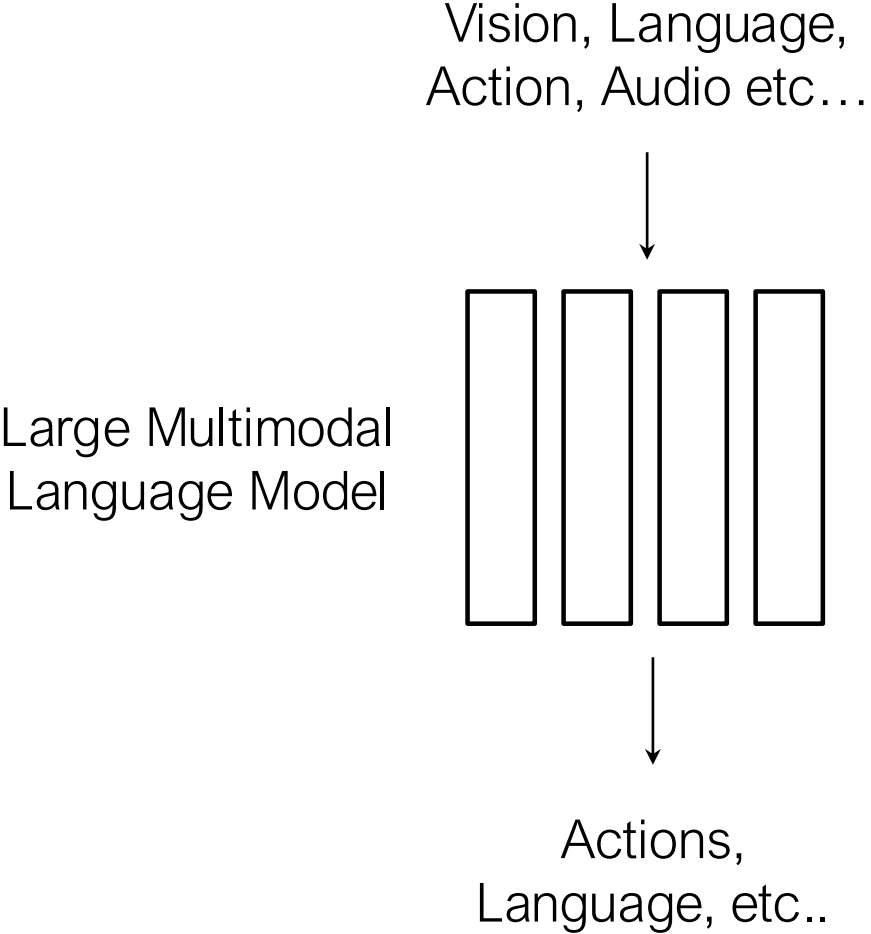
Large Multimodal  
Language Model

Use multiagent  
communication  
between LLMs/other  
foundation models  
to solve complex  
tasks!

Actions,  
Language, etc..



# Towards Compositional AI Systems



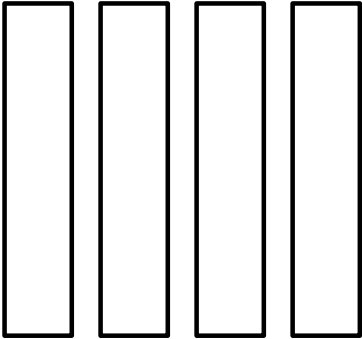
[1] Marvin Minsky. The Society of Mind.

# Towards Compositional AI Systems

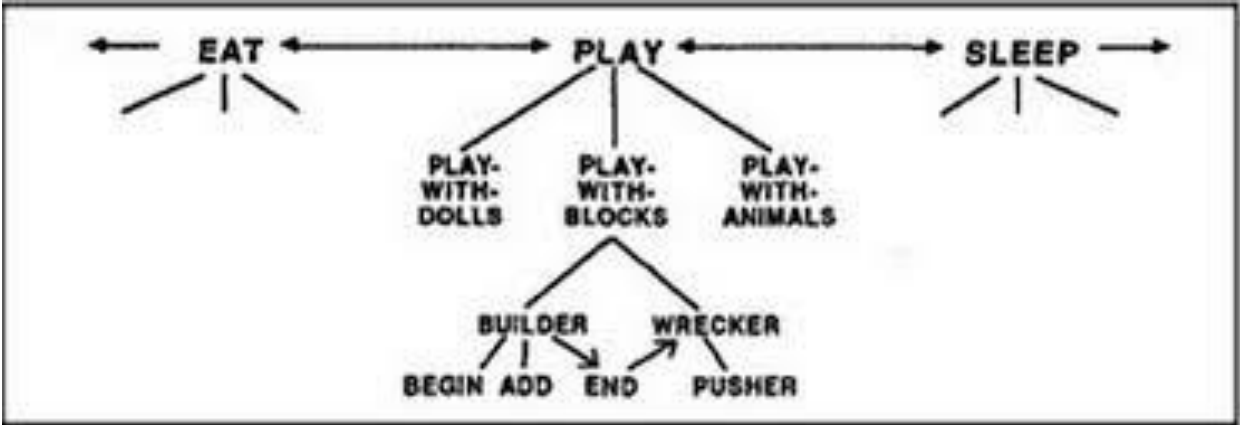
Vision, Language,  
Action, Audio etc...



Large Multimodal  
Language Model



Actions,  
Language, etc..

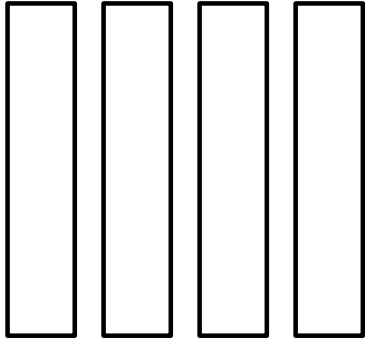


[1] Marvin Minsky. The Society of Mind.

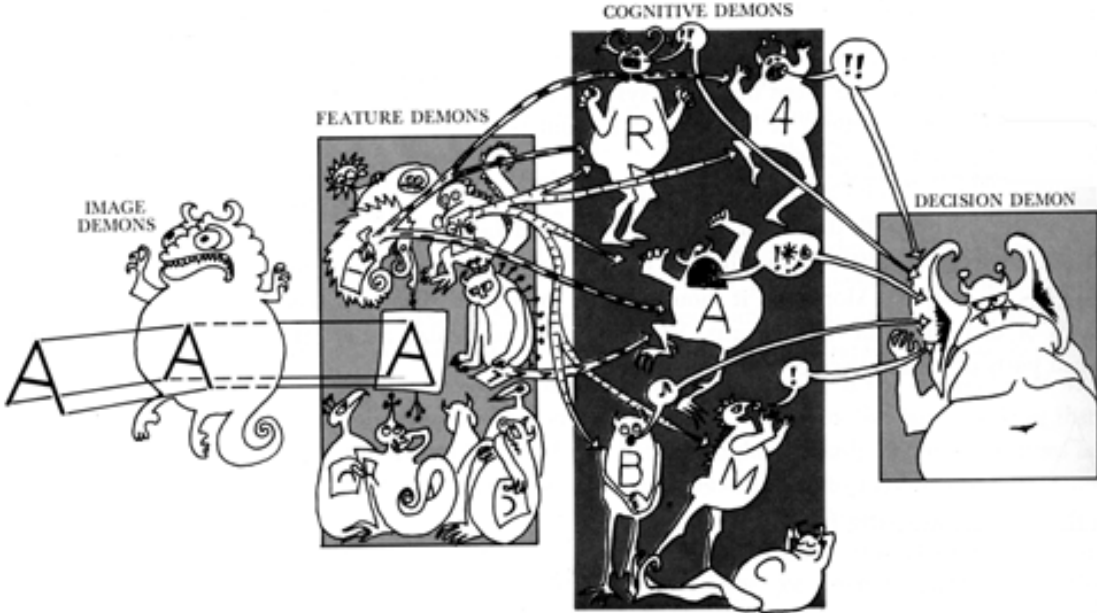
# Towards Compositional AI Systems

Vision, Language,  
Action, Audio etc...

Large Multimodal  
Language Model



Actions,  
Language, etc..



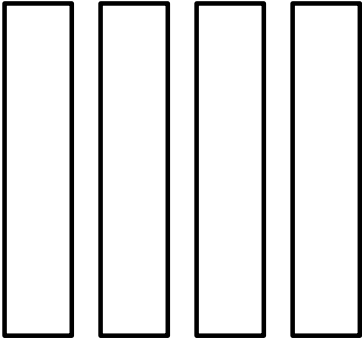
[1] Oliver Selfridge. Pandemonium: A Paradigm for Learning

# Towards Compositional AI Systems

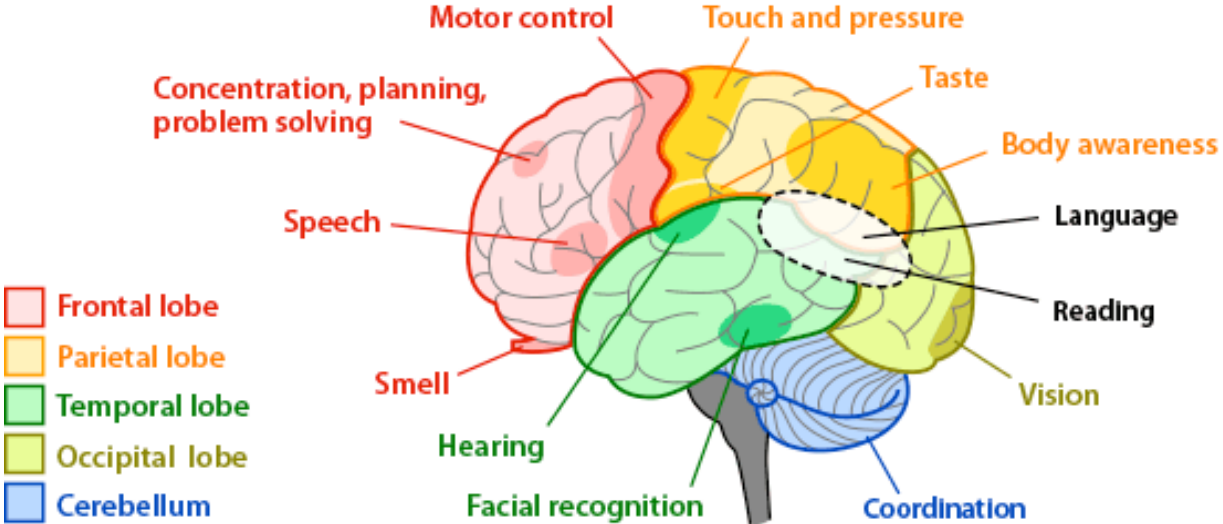
Vision, Language,  
Action, Audio etc...



Large Multimodal  
Language Model



Actions,  
Language, etc..



# Talk Outline

## Multiagent Debate

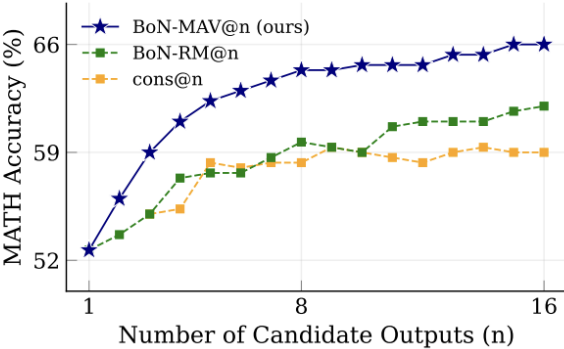
*Using the solutions from other agents, give an updated response* **Prompt**  
<copy Agent 1 and Agent 2 responses>

**Agent 1** Given the information provided in other agents' solutions ... + 245 = 560. Our final answer is **560**. ❌

**Agent 2** After reviewing the solutions provided by other agents ... my answer to the math problem is **595**. ✅

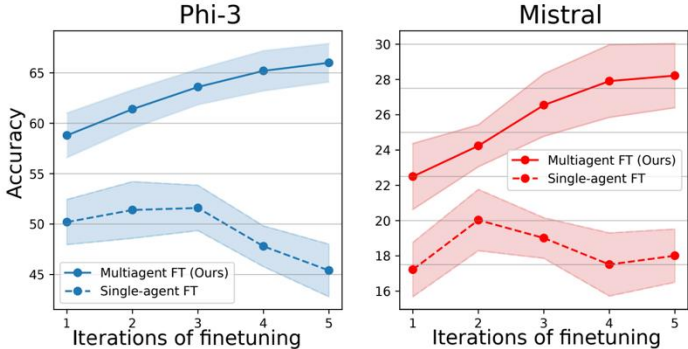
Iterative Communication  
Between Language  
Models

## Multiagent Verification



Combining Language Models  
through Verification

## Multiagent Self-Improvement



Finetuning Compositions of  
Language Models

# Talk Outline

## Multiagent Debate

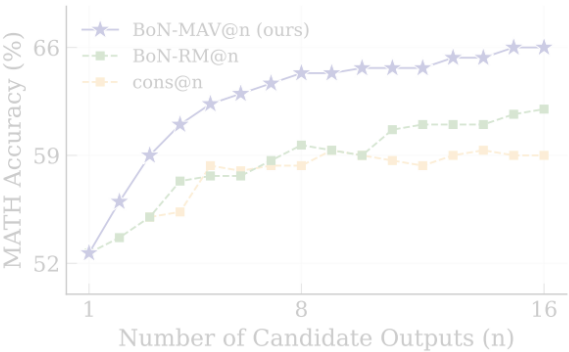
*Using the solutions from other agents, give an updated response* **Prompt**  
<copy Agent 1 and Agent 2 responses>

**Agent 1** Given the information provided in other agents' solutions ... + 245 = 560. Our final answer is **560**. ❌

**Agent 2** After reviewing the solutions provided by other agents ... my answer to the math problem is **595**. ✅

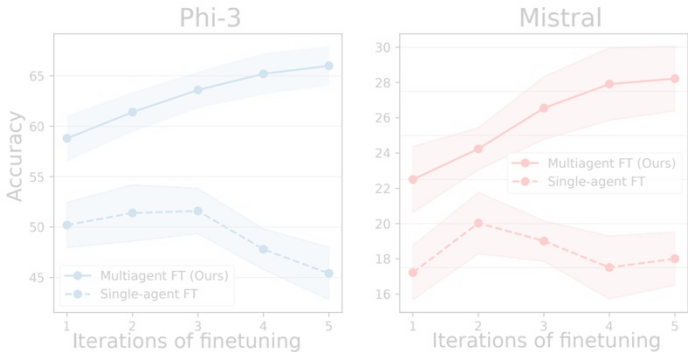
## Iterative Communication Between Language Models

## Multiagent Verification



## Combining Language Models through Verification

## Multiagent Self-Improvement



## Finetuning Compositions of Language Models

# Debate Example (GSM8K)

*... There were 175 diamonds, 35 fewer rubies ... How many of the gems were there in the chest?*

**Prompt**

# Debate Example (GSM8K)

... There were 175 diamonds, 35 fewer rubies ... **How many of the gems were there in the chest?**

**Prompt**

**Agent 1**

Let  $x$  be the number of rubies. Then ... we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



**Agent 2**

If there were 175 diamonds, then ...  $175 + 140 + 280 = 595$ . Therefore, there were **595** gems in the chest.



# Debate Example (GSM8K)

... There were 175 diamonds, 35 fewer rubies ... **How many of the gems were there in the chest?**

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Agent 1

Let  $x$  be the number of rubies. Then ... we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



Agent 2

If there were 175 diamonds, then ...  $175 + 140 + 280 = 595$ . Therefore, there were **595** gems in the chest.



# Debate Example (GSM8K)

... There were 175 diamonds, 35 fewer rubies ... **How many of the gems were there in the chest?**

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Agent 1

Let  $x$  be the number of rubies. Then ... we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



Agent 2

If there were 175 diamonds, then ...  $175 + 140 + 280 = 595$ . Therefore, there were **595** gems in the chest.



Agent 1

Given the information provided in other agents' solutions ...  $+ 245 = 560$ . Our final answer is **560**.



Agent 2

After reviewing the solutions provided by other agents ... my answer to the math problem is **595**.



# Debate Example (GSM8K)

... There were 175 diamonds, 35 fewer rubies ... **How many of the gems were there in the chest?**

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Agent 1

Let  $x$  be the number of rubies. Then ... we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



Agent 2

If there were 175 diamonds, then ...  $175 + 140 + 280 = 595$ . Therefore, there were **595** gems in the chest.



Agent 1

Given the information provided in other agents' solutions ...  $+ 245 = 560$ . Our final answer is **560**.



Agent 2

After reviewing the solutions provided by other agents ... my answer to the math problem is **595**.



# Debate Example (GSM8K)

... There were 175 diamonds, 35 fewer rubies ... **How many of the gems were there in the chest?**

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Using the solutions from other agents, give an updated response  
<copy Agent 1 and Agent 2 responses>

Prompt

Agent 1

Let  $x$  be the number of rubies. Then ... we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



Agent 2

If there were 175 diamonds, then ...  $175 + 140 + 280 = 595$ . Therefore, there were **595** gems in the chest.



Agent 1

Given the information provided in other agents' solutions ...  $+ 245 = 560$ . Our final answer is **560**.



Agent 2

After reviewing the solutions provided by other agents ... my answer to the math problem is **595**.



Agent 1

After reviewing the solutions ... my answer to the math problem is **595**.



Agent 2

After reviewing the solutions ... my answer to the math problem is **595**.



# Multiagent Debate

Round 0



LLM 1



LLM 2



LLM 3

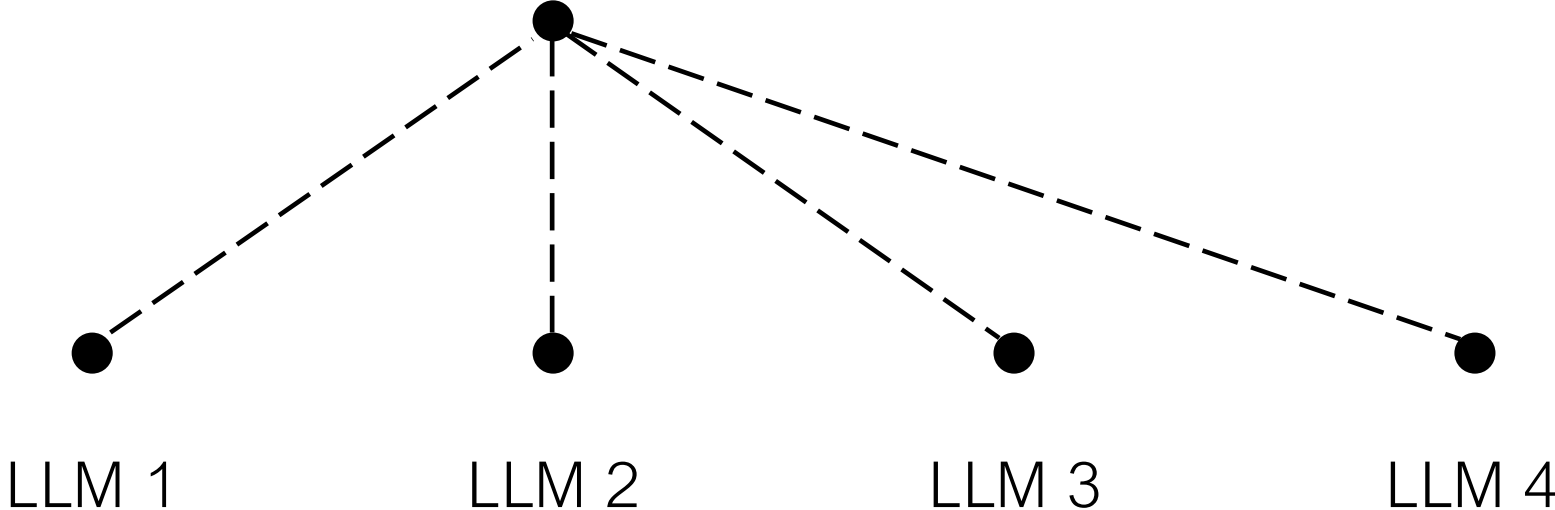


LLM 4

# Multiagent Debate

Round 1

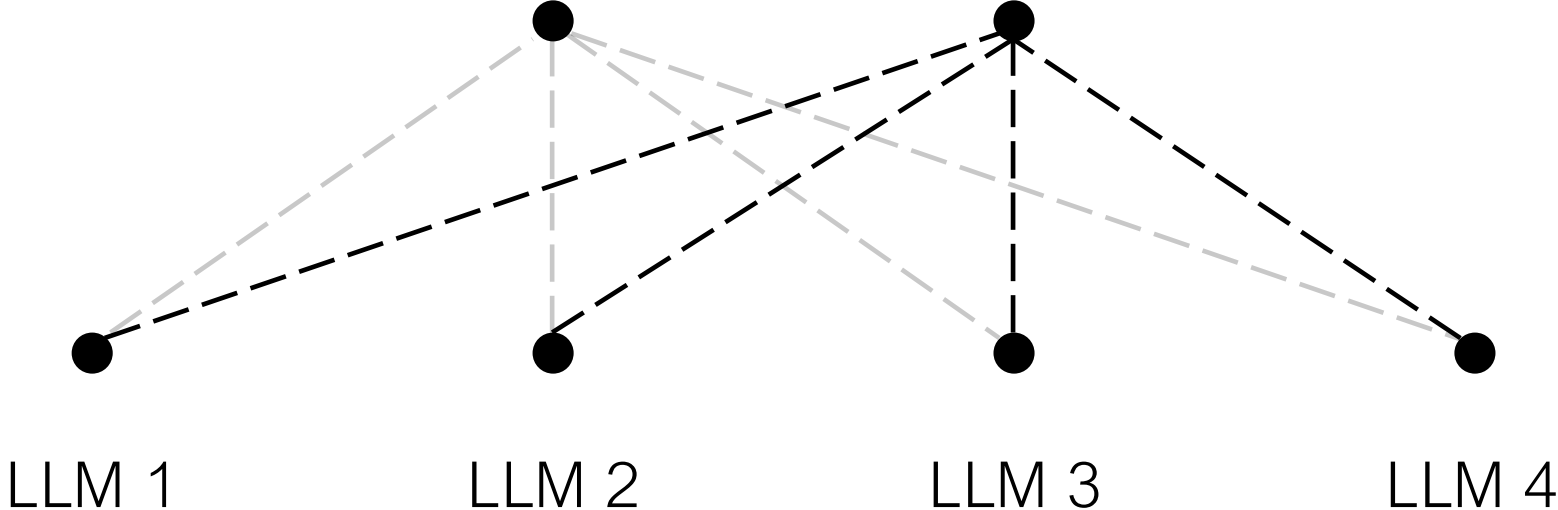
Round 0



# Multiagent Debate

Round 1

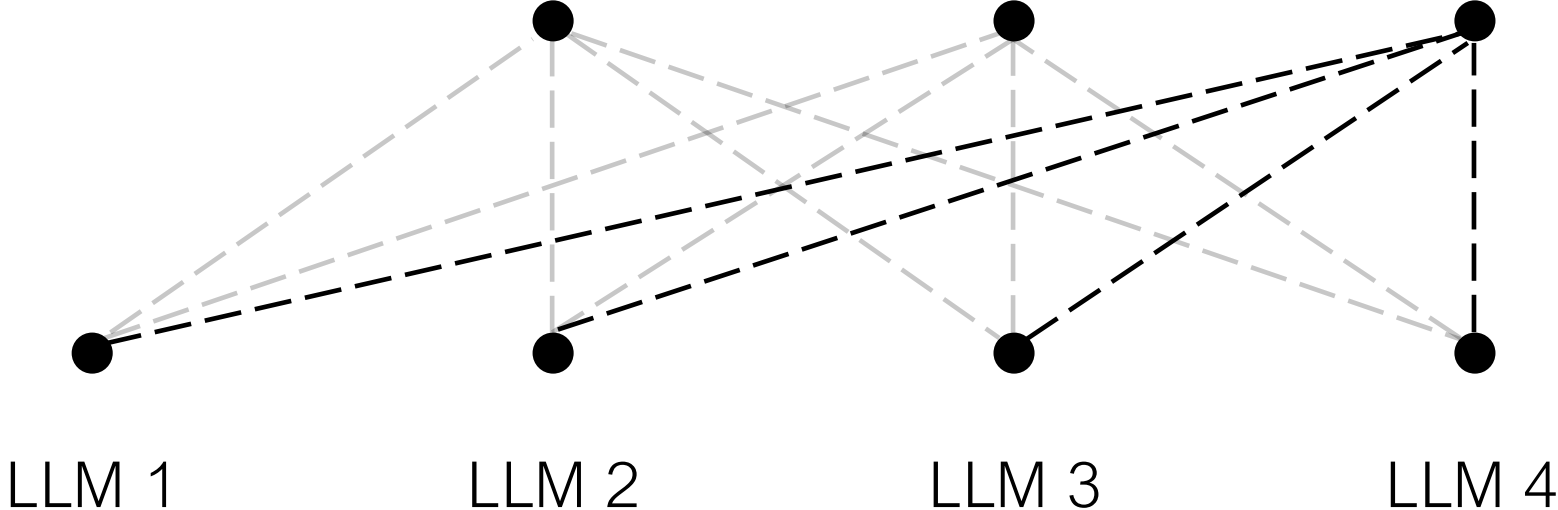
Round 0



# Multiagent Debate

Round 1

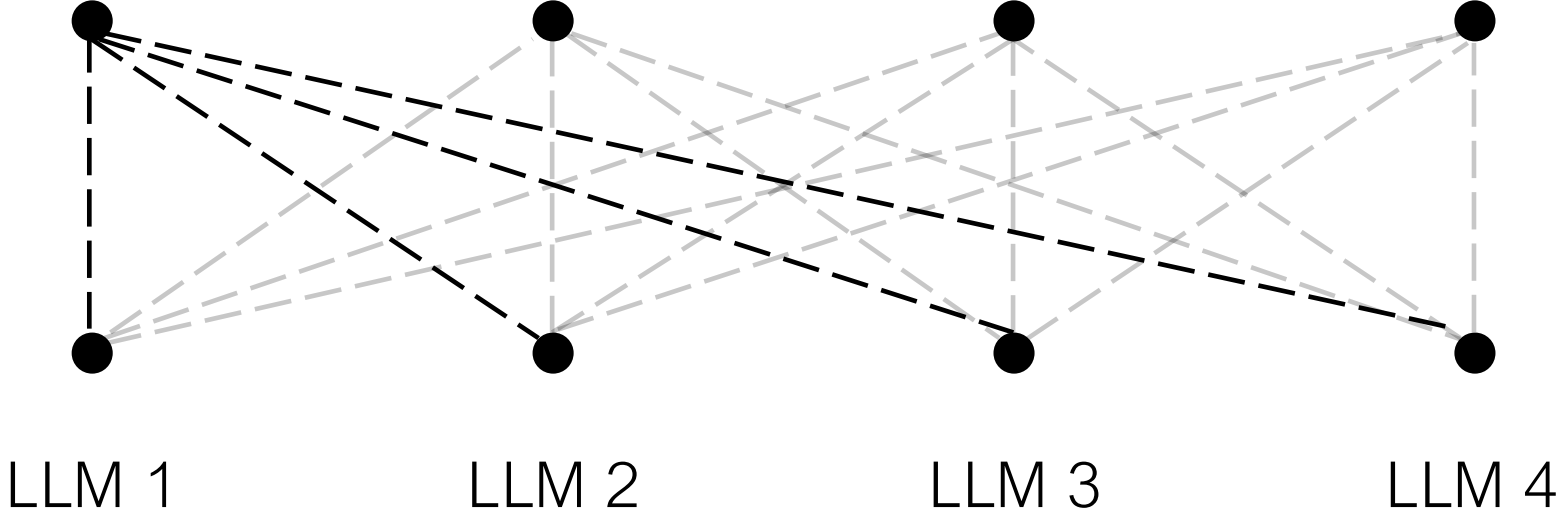
Round 0



# Multiagent Debate

Round 1

Round 0



# Multiagent Debate

Round 2

Round 1

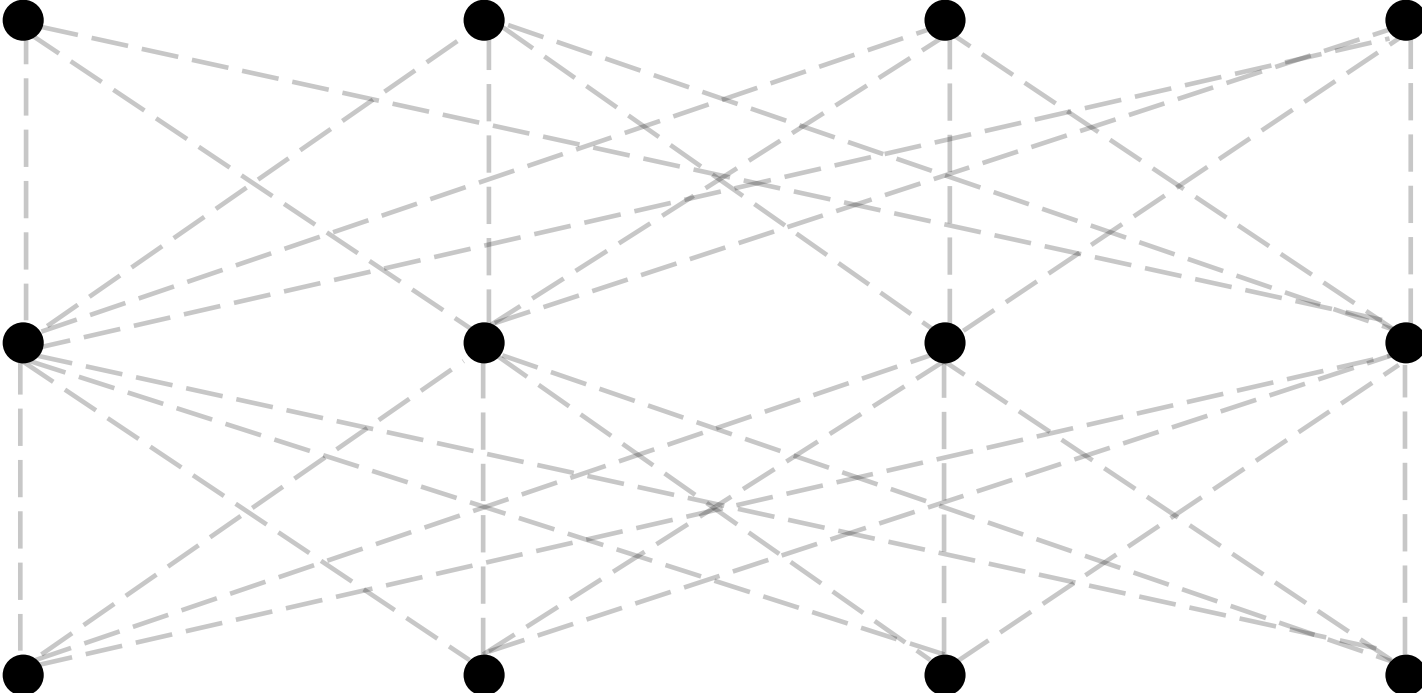
Round 0

LLM 1

LLM 2

LLM 3

LLM 4



# Multiagent Debate

Each agents can critique or assess the factuality of their response in light of other answers

Round 2

Round 1

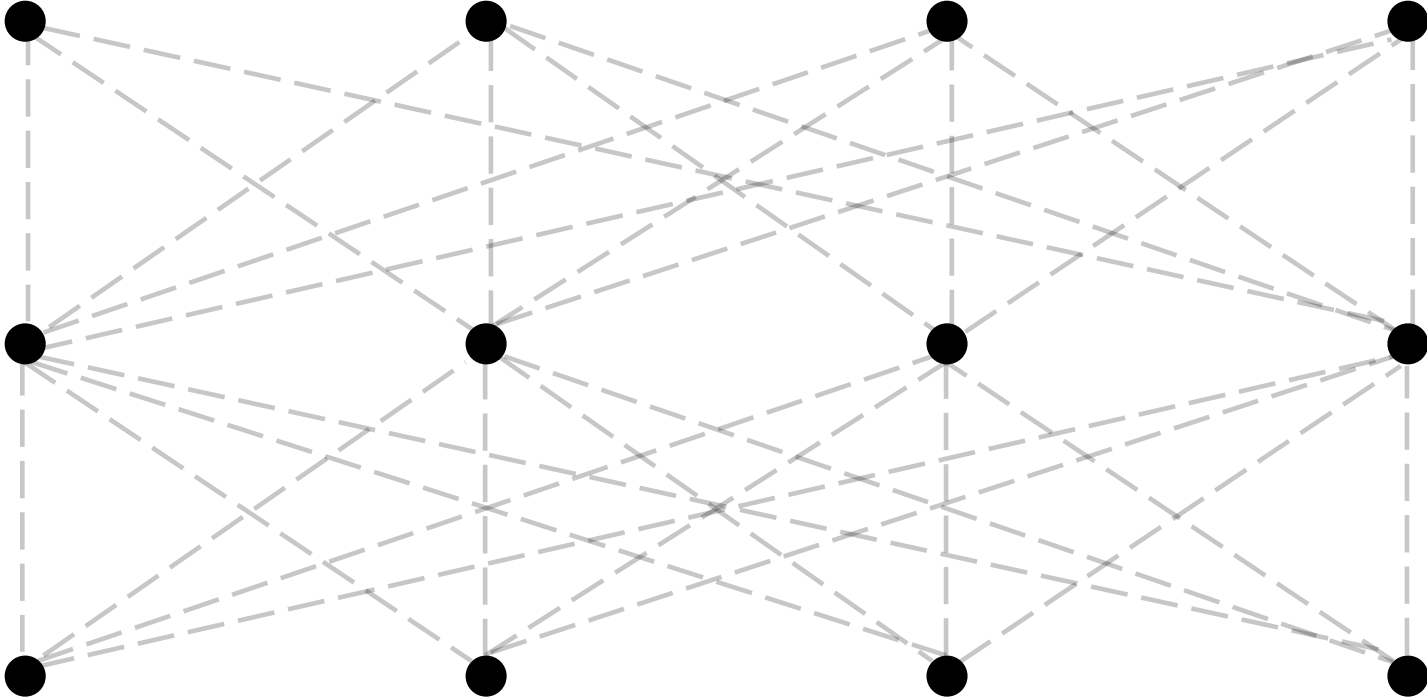
Round 0

LLM 1

LLM 2

LLM 3

LLM 4



# Improving Reasoning

**Instruction:** Can you solve the following math problem? Romeo boards a train with 120 people. At the first stop, 20 more people board the train. At the second stop, 50 people descended from the train while twice that number boarded the train. If 80 more people descended at the third station, calculate the total number of people on the train at the final stop.

# Improving Reasoning

**Instruction:** Can you solve the following math problem? Romeo boards a train with 120 people. At the first stop, 20 more people board the train. At the second stop, 50 people descended from the train while twice that number boarded the train. If 80 more people descended at the third station, calculate the total number of people on the train at the final stop.

## Round 1

**Agent 1:** At the first stop, the total number of people on the train became  $120+20 = 140$ . At the second stop, a total of  $50/2 = 25$  more people boarded the train than those who left, which means  $50-25 = 25$  people boarded the train. The number of people on the train became  $140+25 = 165$ . At the third station, the number of people on the train reduced to  $165-80 = 85$ .



**Agent 2:** After the first stop, there were  $120 + 20 = 140$  people on the train. At the second stop, 50 people descended, leaving  $140 - 50 = 90$  people. Twice that number boarded, meaning  $2 * 50 = 100$  people joined the train. The total number of people on the train became  $90 + 100 = 190$ . At the third stop, 80 people alighted from the train, leaving  $190 - 80 = 110$  people on the train.



# Improving Reasoning

**Instruction:** Can you solve the following math problem? Romeo boards a train with 120 people. At the first stop, 20 more people board the train. At the second stop, 50 people descended from the train while twice that number boarded the train. If 80 more people descended at the third station, calculate the total number of people on the train at the final stop.

## Round 1

**Agent 1:** At the first stop, the total number of people on the train became  $120+20 = 140$ . At the second stop, a total of  $50/2 = 25$  more people boarded the train than those who left, which means  $50-25 = 25$  people boarded the train. The number of people on the train became  $140+25 = 165$ . At the third station, the number of people on the train reduced to  $165-80 = 85$ . ❌

**Instruction:** Here are responses from other agents: (Copy Agent 2's reply) Using the solutions from other agents as additional information, can you give an updated response....

**Agent 2:** After the first stop, there were  $120 + 20 = 140$  people on the train. At the second stop, 50 people descended, leaving  $140 - 50 = 90$  people. Twice that number boarded, meaning  $2 * 50 = 100$  people joined the train. The total number of people on the train became  $90 + 100 = 190$ . At the third stop, 80 people alighted from the train, leaving  $190 - 80 = 110$  people on the train. ✅

**Instruction:** Here are responses from other agents: (Copy Agent 1's reply) Using the solutions from other agents as additional information, can you give an updated response....

# Improving Reasoning

**Instruction:** Can you solve the following math problem? Romeo boards a train with 120 people. At the first stop, 20 more people board the train. At the second stop, 50 people descended from the train while twice that number boarded the train. If 80 more people descended at the third station, calculate the total number of people on the train at the final stop.

## Round 1

**Agent 1:** At the first stop, the total number of people on the train became  $120+20 = 140$ . At the second stop, a total of  $50/2 = 25$  more people boarded the train than those who left, which means  $50-25 = 25$  people boarded the train. The number of people on the train became  $140+25 = 165$ . At the third station, the number of people on the train reduced to  $165-80 = 85$ . ❌

**Agent 2:** After the first stop, there were  $120 + 20 = 140$  people on the train. At the second stop, 50 people descended, leaving  $140 - 50 = 90$  people. Twice that number boarded, meaning  $2 * 50 = 100$  people joined the train. The total number of people on the train became  $90 + 100 = 190$ . At the third stop, 80 people alighted from the train, leaving  $190 - 80 = 110$  people on the train. ✅

**Instruction:** Here are responses from other agents: (Copy Agent 2's reply) Using the solutions from other agents as additional information, can you give an updated response....

**Instruction:** Here are responses from other agents: (Copy Agent 1's reply) Using the solutions from other agents as additional information, can you give an updated response....

## Round 2

**Agent 1:** From the information provided by the other agents, the correct answer to the problem is **110**. ✅

**Agent 2:** Yes, I can provide my answer. Based on the given information and the solutions provided by other agents, the correct answer to the problem is **110**. ✅

# Improving Reasoning

**Instruction:** What is the result of  $23+19*22+22-7*9$ ? Make sure to state your answer at the end of the response.

## Round 1

**Agent 1:** First, we multiply 19 by 22 to get 418. Then, we multiply 7 by 9 to get 63.

We can use the order of operations, also known as PEMDAS, to simplify the expression:

$$\begin{aligned} 23 + 418 + 22 - 63 \\ = 460 \end{aligned}$$

Therefore, the result of  $23+19*22+22-7*9$  is 460.



**Agent 2:** Using the order of operations (PEMDAS), we first multiply 19 and 22, resulting in 418. Then, we multiply 7 and 9, resulting in 63. Next, we subtract 63 from 22, which gives us -41. Then, we add together 23, 418, -41, and 22, which results in 422.

Therefore, the answer is 422.



# Improving Reasoning

**Instruction:** What is the result of  $23+19*22+22-7*9$ ? Make sure to state your answer at the end of the response.

## Round 1

**Agent 1:** First, we multiply 19 by 22 to get 418. Then, we multiply 7 by 9 to get 63.

We can use the order of operations, also known as PEMDAS, to simplify the expression:

$$\begin{aligned} 23 + 418 + 22 - 63 \\ = 460 \end{aligned}$$

Therefore, the result of  $23+19*22+22-7*9$  is 460.



**Agent 2:** Using the order of operations (PEMDAS), we first multiply 19 and 22, resulting in 418. Then, we multiply 7 and 9, resulting in 63. Next, we subtract 63 from 22, which gives us -41. Then, we add together 23, 418, -41, and 22, which results in 422.

Therefore, the answer is 422.



**Instruction:** Using the solutions from other agents as additional information. (Copy Agent 2's reply)

**Instruction:** Using the solutions from other agents as additional information. (Copy Agent 1's reply)

## Round 2

**Agent 1:** It seems that the first agent made an error by subtracting 63 from 22 instead of subtracting 63 from the previous result of the multiplication of 19 and 22.

Using the correct order of operations, the expression should be simplified as follows:

$$\begin{aligned} 19*22 &= 418 \\ 7*9 &= 63 \\ 23+418+22-63 &= 400 \end{aligned}$$

Therefore, the updated answer is 400.



**Agent 2:** The order of operations (PEMDAS) must be followed to correctly simplify the expression.

First, we multiply 19 and 22 to get 418.

Next, we multiply 7 and 9 to get 63.

Then, we add 23 to 418 to get 441.

After that, we add 22 to get 463.

Finally, we subtract the product of 7 and 9 (63) from 463, resulting in 400.

Therefore, the updated answer for  $23+19*22+22-7*9$  is 400.



# Improving Reasoning

**Question:** What is the result of  $10+20*23+3-11*18$ ?

**Round 1**    **Agent 1:** 269 ❌    **Agent 2:** 369 ❌

**Round 2**    **Agent 1:** 275 ✅    **Agent 2:** 275 ✅

**Question:** What is the result of  $3+7*9+19-21*18$ ?

**Agent 1:** 378 ❌    **Agent 2:** -351 ❌    **Agent 3:** -357 ❌

**Agent 1:** -293 ✅    **Agent 2:** -293 ✅    **Agent 3:** 19 ❌

**Question:** What is the result of  $4+23*6+24-24*12$ ?

**Round 1**    **Agent 1:** -244 ❌    **Agent 2:** -146 ❌

**Round 2**    **Agent 1:** -146 ❌    **Agent 2:** -122 ✅

**Round 3**    **Agent 1:** -122 ✅    **Agent 2:** -122 ✅

**Question:** What is the result of  $8+14*15+20-3*26$ ?

**Agent 1:** 236 ❌    **Agent 2:** -214 ❌    **Agent 3:** 210 ❌

**Agent 1:** 160 ✅    **Agent 2:** 160 ✅    **Agent 3:** 160 ✅

**Agent 1:** 160 ✅    **Agent 2:** 160 ✅    **Agent 3:** 160 ✅

# Improving Factuality

**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. ❌

-- Johnson has received numerous awards for his research and contributions to computer science, including the **ACM Turing Award** in 2013. ❌

-- Johnson's research interests include optimization, algorithms, and complexity theory. ✅

**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 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. ✅

-- Johnson was elected to the National Academy of Engineering in 2006. ✅

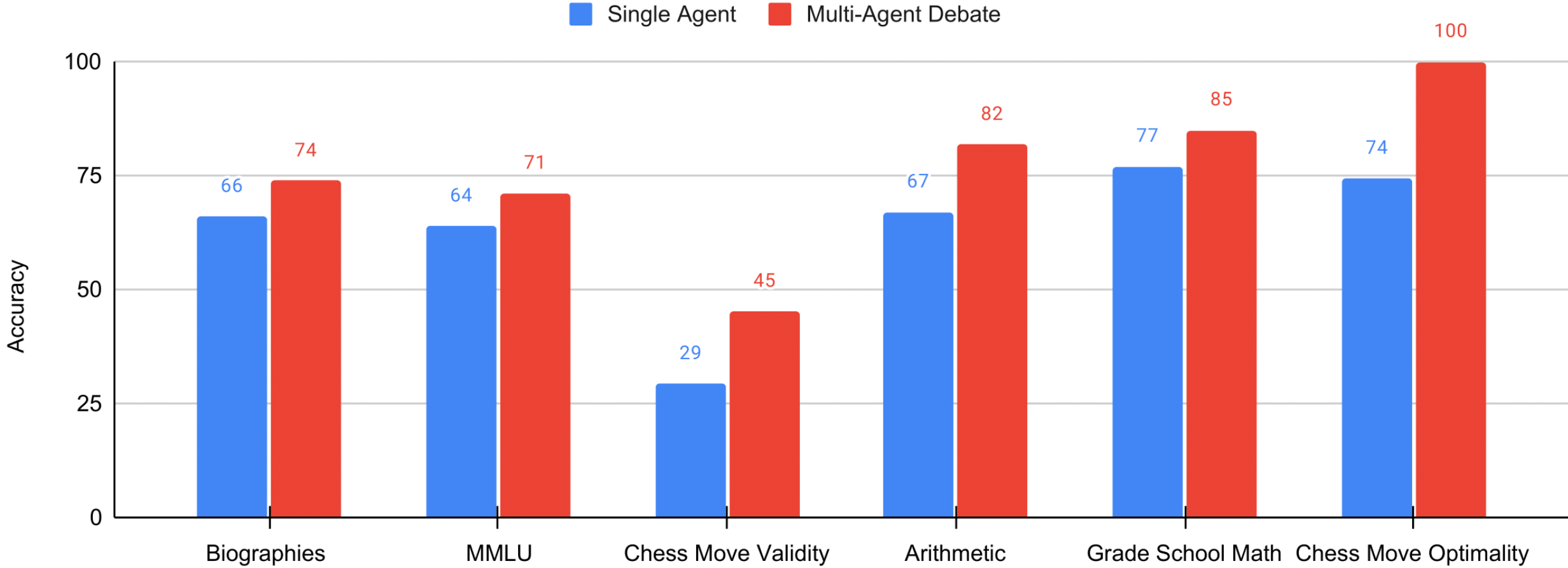
-- He has received numerous awards for his contributions to computer science, including the **ACM Turing Award** in 2013 and the Knuth Prize. ❌

**Agent 2:** -- David S. Johnson is an American computer scientist and mathematician. ✅

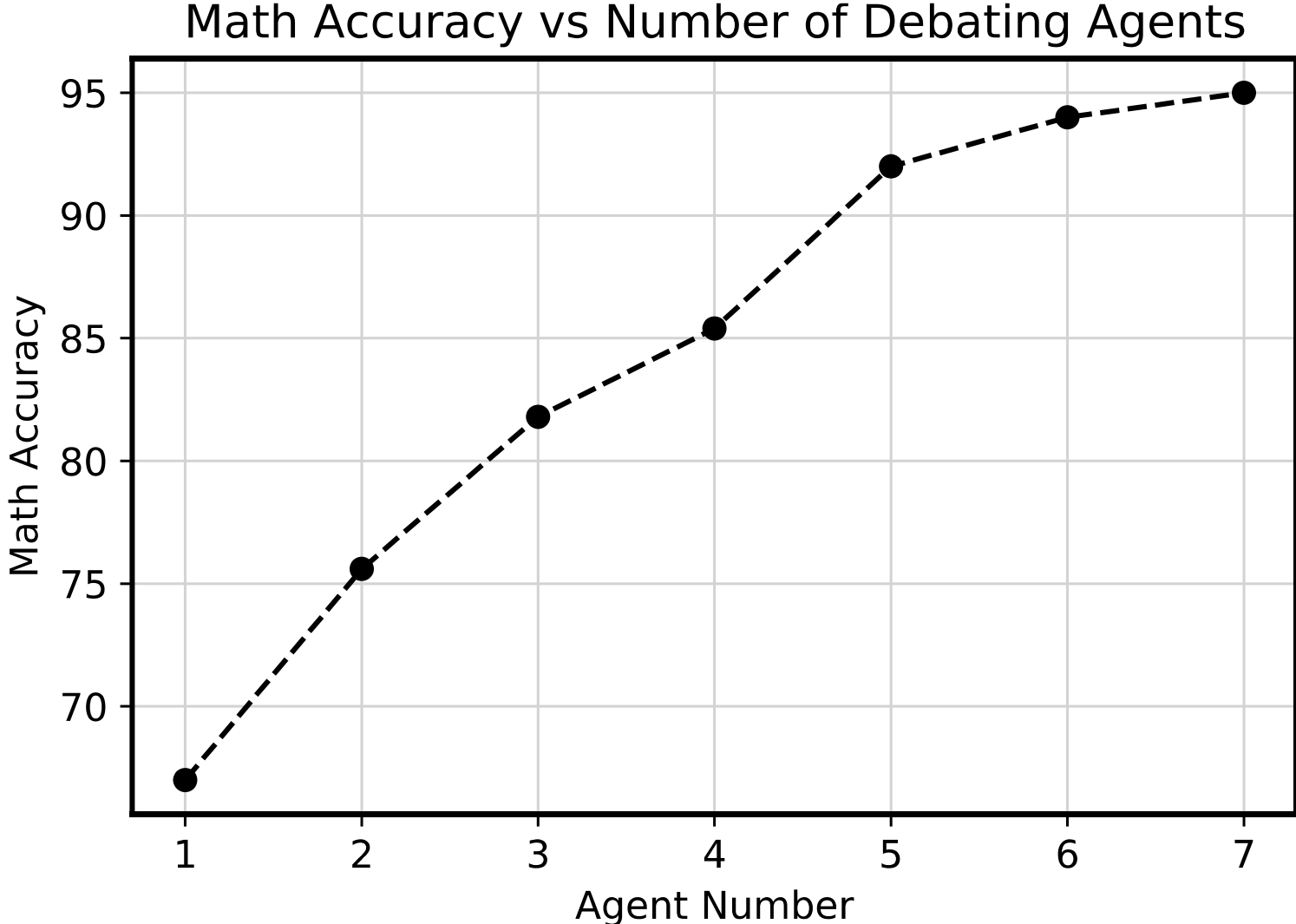
-- He has worked at several prestigious universities, including Columbia University, AT&T Bell Laboratories, and **Cornell University**. ❌

-- Johnson is known for his work in the field of algorithms, particularly in the areas of network optimization, combinatorial optimization, and scheduling theory. ✅

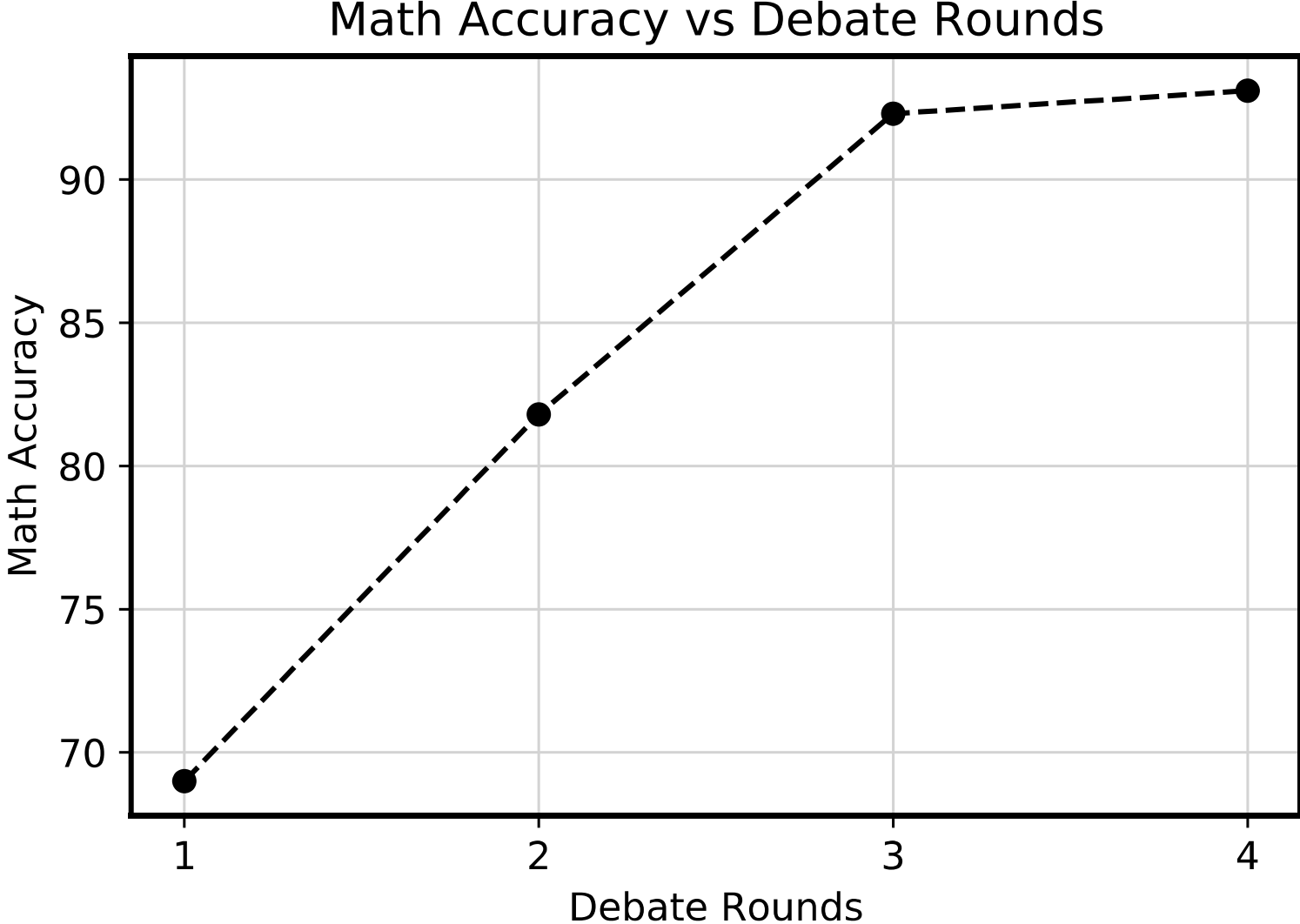
# Multiagent Debate Helps Accuracy and Reasoning



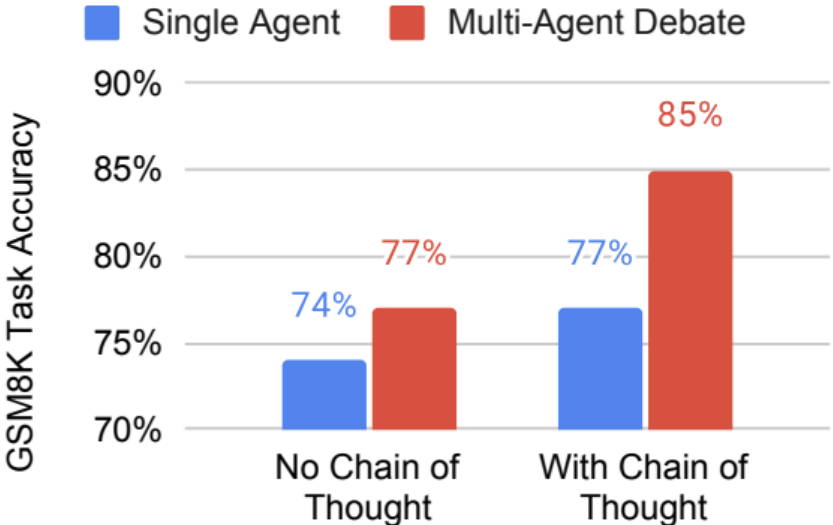
# Performance with More Agents



# Performance with More Rounds of Debate

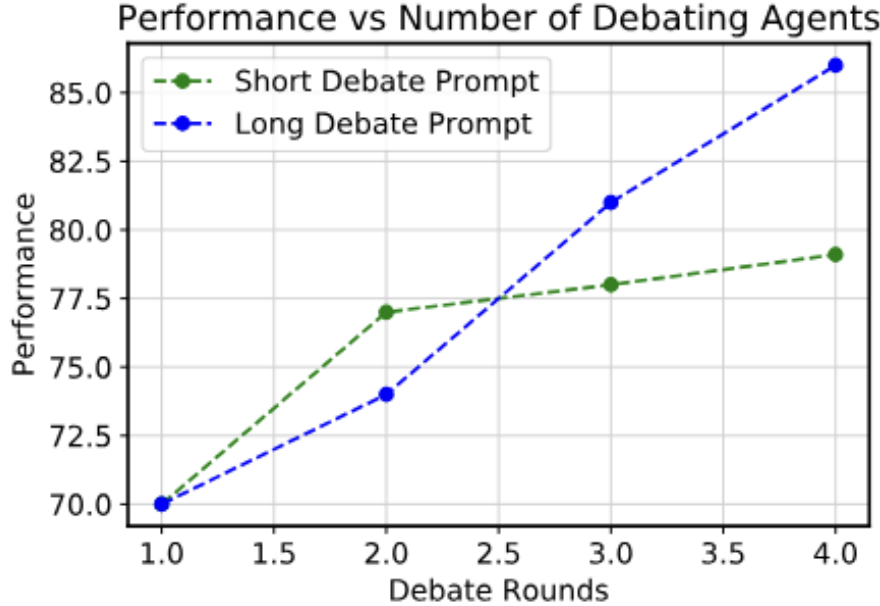


# Synergy with Other Prompting Methods



*Figure 6. Synergy with Other Methods.* Performance of debate increases with use of zero-shot Chain of Thought prompting on GSM8K. Results with few-shot Chain of Thought prompting can be found in Section [A.1](#).

# Performance with Debate Length



*Figure 12. Performance vs Debate Length.* Prompts which induce longer debate improve performance. Analysis on GSM8K.

# Talk Outline

## Multiagent Debate

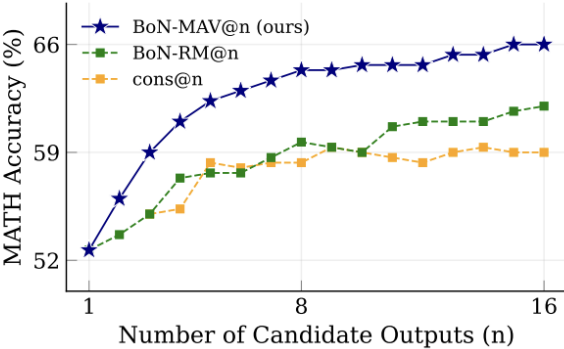
*Using the solutions from other agents, give an updated response* **Prompt**  
<copy Agent 1 and Agent 2 responses>

**Agent 1** Given the information provided in other agents' solutions ... + 245 = 560. Our final answer is **560**. ❌

**Agent 2** After reviewing the solutions provided by other agents ... my answer to the math problem is **595**. ✅

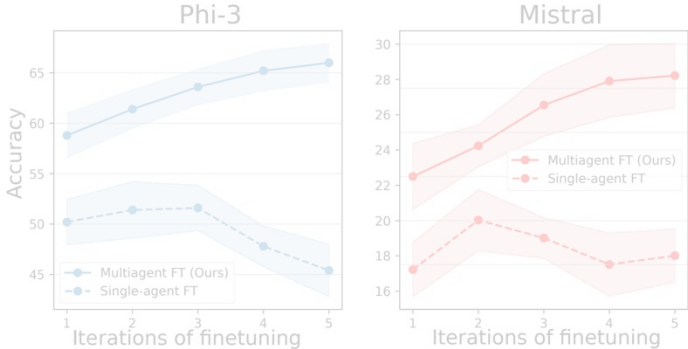
## Iterative Communication Between Language Models

## Multiagent Verification



## Combining Language Models through Verification

## Multiagent Self-Improvement



## Finetuning Compositions of Language Models

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

**Sampled Candidate Output: (INCORRECT)**

Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \underline{411/333}$

[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

**Sampled Candidate Output: (INCORRECT)**

Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \mathbf{411/333}$

**Aspect Verifier #1**  
Base model: Gemini-1.5-Flash  
Aspect: General Correctness  
Strategy: Direct Approval

APPROVAL = TRUE

[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111  
**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

**Sampled Candidate Output: (INCORRECT)**  
Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \underline{411/333}$

**Aspect Verifier #1**  
Base model: Gemini-1.5-Flash  
Aspect: General Correctness  
Strategy: Direct Approval  
  
**APPROVAL = TRUE**

**Aspect Verifier #2**  
Base model: GPT-4o-mini  
Aspect: Unit Conversions  
Strategy: Step-by-Step  
  
The problem involves converting a decimal to a fraction, which doesn't involve any units. Therefore, there are no units to check.  
  
**APPROVAL = TRUE**

[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

**Sampled Candidate Output: (INCORRECT)**

Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \mathbf{411/333}$

**Aspect Verifier #1**  
Base model: Gemini-1.5-Flash  
Aspect: General Correctness  
Strategy: Direct Approval

APPROVAL = TRUE

**Aspect Verifier #2**  
Base model: GPT-4o-mini  
Aspect: Unit Conversions  
Strategy: Step-by-Step

The problem involves converting a decimal to a fraction, which doesn't involve any units. Therefore, there are no units to check.

APPROVAL = TRUE

**Aspect Verifier #3**  
Base model: Gemini-1.5-Flash  
Aspect: Logical Soundness  
Strategy: Step-by-Step

Let's analyze the proposed solution step by step.  
1. Let  $x$  equal 1.234234...: This is correctly stated. We define  $x = 1.234234\dots$   
...  
Now, let's check if 411/333 can be simplified further. The greatest common divisor (GCD) of 411 and 333 is 3, so we can simplify it further:  $411 \div 3 / 333 \div 3 = 137/111$   
Thus, the final simplified form of 1.234234... as an improper fraction is 137/111, not 411/333.  
Since the proposed solution contains an error in the final simplification step, we conclude that the solution is not entirely correct.

APPROVAL = FALSE

[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

## Sampled Candidate Output: (INCORRECT)

Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \underline{411/333}$

### Aspect Verifier #1

Base model: Gemini-1.5-Flash  
Aspect: General Correctness  
Strategy: Direct Approval

APPROVAL = TRUE

### Aspect Verifier #2

Base model: GPT-4o-mini  
Aspect: Unit Conversions  
Strategy: Step-by-Step

The problem involves converting a decimal to a fraction, which doesn't involve any units. Therefore, there are no units to check.

APPROVAL = TRUE

### Aspect Verifier #3

Base model: Gemini-1.5-Flash  
Aspect: Logical Soundness  
Strategy: Step-by-Step

Let's analyze the proposed solution step by step.  
1. Let  $x$  equal 1.234234...: This is correctly stated. We define  $x = 1.234234\dots$   
...  
Now, let's check if 411/333 can be simplified further. The greatest common divisor (GCD) of 411 and 333 is 3, so we can simplify it further:  $411 \div 3 / 333 \div 3 = 137/111$   
Thus, the final simplified form of 1.234234... as an improper fraction is 137/111, not 411/333.

Since the proposed solution contains an error in the final simplification step, we conclude that the solution is not entirely correct.

APPROVAL = FALSE

### Aspect Verifier #4

Base model: GPT-4o-mini  
Aspect: General Correctness  
Strategy: Summarize Solution

The proposed solution begins by letting  $x$  equal the repeating decimal 1.234234..., which is correctly represented as  $x = 1.234234\dots$   
...  
Thus, the fraction simplifies correctly to:  $x = 411/333$ .  
However, the solution does not check if 411/333 can be simplified further. Both 411 and 333 can be divided by 3 again:  $411 \div 3 = 137$  and  $333 \div 3 = 111$ . Thus, the fraction can be simplified further to:  $x = 137/111$

The final boxed answer 411/333 is not in its simplest form. Therefore, the proposed solution contains an error in the simplification step.

APPROVAL = FALSE

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

## Sampled Candidate Output: (INCORRECT)

Let  $x = 1.234234\dots$   
Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
Subtract the first equation from the second equation:  $999x = 1233$   
Solve for  $x$ :  $x = 1233/999$   
Simplify by dividing the numerator and denominator by 3:  $x = \underline{411/333}$

### Aspect Verifier #1

Base model: Gemini-1.5-Flash  
Aspect: General Correctness  
Strategy: Direct Approval

APPROVAL = TRUE

### Aspect Verifier #2

Base model: GPT-4o-mini  
Aspect: Unit Conversions  
Strategy: Step-by-Step

The problem involves converting a decimal to a fraction, which doesn't involve any units. Therefore, there are no units to check.

APPROVAL = TRUE

### Aspect Verifier #3

Base model: Gemini-1.5-Flash  
Aspect: Logical Soundness  
Strategy: Step-by-Step

Let's analyze the proposed solution step by step.  
1. Let  $x$  equal 1.234234...: This is correctly stated. We define  $x = 1.234234\dots$   
...  
Now, let's check if 411/333 can be simplified further. The greatest common divisor (GCD) of 411 and 333 is 3, so we can simplify it further:  $411 \div 3 / 333 \div 3 = 137/111$   
Thus, the final simplified form of 1.234234... as an improper fraction is 137/111, not 411/333.  
Since the proposed solution contains an error in the final simplification step, we conclude that the solution is not entirely correct.

APPROVAL = FALSE

### Aspect Verifier #4

Base model: GPT-4o-mini  
Aspect: General Correctness  
Strategy: Summarize Solution

The proposed solution begins by letting  $x$  equal the repeating decimal 1.234234..., which is correctly represented as  $x = 1.234234\dots$   
...  
Thus, the fraction simplifies correctly to:  $x = 411/333$ .  
However, the solution does not check if 411/333 can be simplified further. Both 411 and 333 can be divided by 3 again:  $411 \div 3 = 137$  and  $333 \div 3 = 111$ . Thus, the fraction can be simplified further to:  $x = 137/111$   
The final boxed answer 411/333 is not in its simplest form. Therefore, the proposed solution contains an error in the simplification step.

APPROVAL = FALSE

### Aspect Verifier #5

Base model: Gemini-1.5-Flash  
Aspect: Math Correctness  
Strategy: Step-by-Step

Okay, let's go through the solution step-by-step.  
• Step 1: Let  $x$  equal 1.234234... This is a valid way to start, assigning a variable to the repeating decimal.  
...  
• Step 5: Solve for  $x$ :  $x = 1233/999$ . This is correct, dividing both sides by 999 isolates  $x$ .  
...  
• Step 6: Simplify by dividing the numerator and denominator by 3:  $x = 411/333$ . This is where the error occurs. While both 1233 and 999 are divisible by 3, the fraction can be simplified further.

APPROVAL = FALSE

# Multi-Agent Verification

**Question:** Express 1.234234... as an improper fraction.  
**Correct Answer:** 137/111

**Generated Answer:** 411/333  
**Aggregated Verification Score:** 2/5

**Sampled Candidate Output: (INCORRECT)**

Let  $x = 1.234234\dots$   
 Multiply both sides of the equation by 1000:  $1000x = 1234.234234\dots$   
 Subtract the first equation from the second equation:  $999x = 1233$   
 Solve for  $x$ :  $x = 1233/999$   
 Simplify by dividing the numerator and denominator by 3:  $x = \mathbf{411/333}$

**Aspect Verifier #1**  
 Base model: Gemini-1.5-Flash  
 Aspect: General Correctness  
 Strategy: Direct Approval

**Aspect Verifier #2**  
 Base model: GPT-4o-mini  
 Aspect: Unit Conversions  
 Strategy: Step-by-Step

**Aspect Verifier #3**  
 Base model: Gemini-1.5-Flash  
 Aspect: Logical Soundness  
 Strategy: Step-by-Step

**Aspect Verifier #4**  
 Base model: GPT-4o-mini  
 Aspect: General Correctness  
 Strategy: Summarize Solution

**Aspect Verifier #5**  
 Base model: Gemini-1.5-Flash  
 Aspect: Math Correctness  
 Strategy: Step-by-Step

The problem involves converting a decimal to a fraction, which doesn't involve any units. Therefore, there are no units to check.

Let's analyze the proposed solution step by step.  
 1. Let  $x$  equal 1.234234...: This is correctly stated. We define  $x = 1.234234\dots$

The proposed solution begins by letting  $x$  equal the repeating decimal 1.234234..., which is correctly represented as  $x = 1.234234\dots$

Okay, let's go through the solution step-by-step.  
 • Step 1: Let  $x$  equal 1.234234... This is a valid way to start, assigning a variable to the repeating decimal.

...  
 Now, let's check if 411/333 can be simplified further. The greatest common divisor (GCD) of 411 and 333 is 3, so we can simplify it further:  $411 \div 3 / 333 \div 3 = 137/111$

...  
 Thus, the fraction simplifies correctly to:  $x = 411/333$ .

...  
 • Step 5: Solve for  $x$ :  $x = 1233/999$ . This is correct, dividing both sides by 999 isolates  $x$ .

Thus, the final simplified form of 1.234234... as an improper fraction is 137/111, not 411/333.

However, the solution does not check if 411/333 can be simplified further. Both 411 and 333 can be divided by 3 again:  $411 \div 3 = 137$  and  $333 \div 3 = 111$ . Thus, the fraction can be simplified further to:  $x = 137/111$

• Step 6: Simplify by dividing the numerator and denominator by 3:  $x = 411/333$ . This is where the error occurs. While both 1233 and 999 are divisible by 3, the fraction can be simplified further.

Since the proposed solution contains an error in the final simplification step, we conclude that the solution is not entirely correct.

The final boxed answer 411/333 is not in its simplest form. Therefore, the proposed solution contains an error in the simplification step.

APPROVAL = TRUE

APPROVAL = TRUE

APPROVAL = FALSE

APPROVAL = FALSE

APPROVAL = FALSE

[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

**Question:**

```
def eat(number, need, remaining):
```

```
    """
```

```
    You're a hungry rabbit, and you already have eaten a certain number of carrots,
    but now you need to eat more carrots to complete the day's meals.
    you should return an array of [ total number of eaten carrots after your meals,
    the number of carrots left after your meals ]
    if there are not enough remaining carrots, you will eat all remaining carrots, but will still be hungry.
```

```
    Example:
```

```
* eat(5, 6, 10) -> [11, 4]
* eat(4, 8, 9) -> [12, 1]
* eat(1, 10, 10) -> [11, 0]
* eat(2, 11, 5) -> [7, 0]
```

```
    Variables:
```

```
@number : integer
    the number of carrots that you have eaten.
@need : integer
    the number of carrots that you need to eat.
@remaining : integer
    the number of remaining carrots that exist in stock
    """
```

**Correct Answer:**

```
def eat(number, need, remaining):
    if(need <= remaining):
        return [ number + need , remaining-need ]
    else:
        return [ number + remaining , 0]
```

**Sampled Candidate Output: (INCORRECT)**

```
def eat(number, need, remaining):
    eaten = min(number + remaining, need)
    return [eaten, remaining - (eaten - number)]
```

**Aggregated Verification Score: 1/3**

**Aspect Verifier #1**

Base model: **GPT-4o-mini**

Aspect: **Logical Soundness**

Strategy: **Step-by-Step**

Let's analyze the proposed solution step by step.

...

**2. Calculating Total Eaten Carrots:**

- The line `eaten = min(number + remaining, need)` is intended to calculate how many carrots the rabbit will have eaten after the meal.
- Here, `number + remaining` gives the total carrots the rabbit could potentially eat (the carrots already eaten plus the remaining ones).
- The `min` function is used to ensure that the total eaten does not exceed the need. This is correct because if the rabbit needs more carrots than it can eat, it will only eat what it can.

**3. Calculating Remaining Carrots:**

- The return statement is `return [eaten, remaining - (eaten - number)]`.
- The expression `remaining - (eaten - number)` is meant to calculate how many carrots are left after the meal.
- However, this expression is incorrect. The correct calculation should simply be `remaining - (eaten - number)` only if `eaten` is greater than `number`. If `eaten` is less than or equal to `number`, the remaining carrots should not change, as the rabbit has not eaten any additional carrots.

**4. Final Calculation:**

- If the rabbit eats all the remaining carrots, it should still be hungry, which means the total eaten should be `number + remaining`, and the remaining should be 0.
- If the rabbit eats less than it needs, the remaining should be calculated correctly based on how many it actually ate.

Given the above analysis, the calculation for remaining carrots is flawed. The logic does not correctly account for the scenario where the rabbit eats all remaining carrots and still needs more.

Therefore, the proposed solution is incorrect.

**APPROVAL = FALSE**

**Aspect Verifier #3**

Base model: **Gemini-1.5-Flash**

Aspect: **General Correctness**

Strategy: **Edge Cases**

Let's analyze the proposed solution step by step to ensure it handles all edge cases and boundary conditions correctly.

...

**3. Identifying Issues:**

- The calculation of remaining carrots is incorrect. The expression `remaining - (eaten - number)` does not correctly reflect the number of remaining carrots after eating. Instead, it should be `remaining - (eaten - number)` only if `eaten` is greater than `number`. If `eaten` is less than or equal to `number`, the remaining carrots should simply be `remaining`.
- The logic does not handle the case where the rabbit eats all remaining carrots but still needs more. In this case, the total eaten should be `number + remaining`, and the remaining should be 0.

**4. Testing Edge Cases:**

- Case 1: `eat(0, 0, 0)` should return `[0, 0]`.
- Case 2: `eat(0, 1, 0)` should return `[0, 0]` (not enough to eat).
- Case 3: `eat(0, 1, 1)` should return `[1, 0]` (eats the only remaining carrot).
- Case 4: `eat(5, 6, 10)` should return `[11, 4]` (correct).
- Case 5: `eat(4, 8, 9)` should return `[12, 1]` (correct).
- Case 6: `eat(1, 10, 10)` should return `[11, 0]` (correct).
- Case 7: `eat(2, 11, 5)` should return `[7, 0]` (correct).
- Case 8: `eat(1000, 1000, 1000)` should return `[1000, 1000]` (already satisfied).
- Case 9: `eat(1000, 1000, 0)` should return `[1000, 0]` (already satisfied, no remaining).

**5. Conclusion:**

- The proposed solution has a flaw in calculating the remaining carrots and does not correctly handle the case where the rabbit eats all remaining carrots but still needs more.
- Therefore, the solution does not handle all edge cases and boundary conditions correctly.

**APPROVAL = FALSE**

**Aspect Verifier #2**

Base model: **Gemini-1.5-Flash**

Aspect: **Math Correctness**

Strategy: **Step-by-Step**

Okay, let's analyze the code step by step.

1. `eaten = min(number + remaining, need)`: This line calculates the total number of carrots eaten. It takes the minimum value between the sum of the carrots already eaten (`number`) and the remaining carrots (`remaining`) and the total number of carrots needed (`need`). This is correct because the rabbit can't eat more than it needs or more than what's available.

2. `return [eaten, remaining - (eaten - number)]`: This line calculates the remaining carrots. It subtracts the difference between the total eaten (`eaten`) and the carrots already eaten (`number`) from the initial remaining carrots (`remaining`). This is also correct because it accurately reflects the number of carrots left after the rabbit eats.

**APPROVAL = TRUE**

# Multi-Agent Verification

**Question:** Two stars (Star\_1 and Star\_2) each have masses 1.5 and 1.2 times that of our Sun, respectively. Assuming LTE and using the EW method, astronomers have determined the elemental abundances of these two stars:  $[\text{Si}/\text{Fe}]_1 = 0.3$  dex,  $[\text{Mg}/\text{Si}]_2 = 0.3$  dex,  $[\text{Fe}/\text{H}]_1 = 0$  dex, and  $[\text{Mg}/\text{H}]_2 = 0$  dex. Consider the following photospheric composition for the Sun:  $12 + \log_{10}(n_{\text{Fe}}/n_{\text{H}}) = 7.5$  and  $12 + \log_{10}(n_{\text{Mg}}/n_{\text{H}}) = 7$ . Calculate the ratio of silicon atoms in the photospheres of Star\_1 and Star\_2.

**Options:** (A) ~3.9, (B) ~12.6, (C) ~0.8, (D) ~1.2

**Correct Answer:** (B)

**Output #1:** Here's how we can solve this problem step-by-step: ... The answer is (A)

AV #1: ✓	AV #2: ✗	AV #3: ✗
AV #4: ✗	AV #5: ✗	AV #6: ✓
AV #7: ✗	AV #8: ✗	AV #9: ✗

Aggregated Score: 2/9

**Output #2:** Here's how to break down the problem step-by-step: ... The answer is (B)

AV #1: ✓	AV #2: ✓	AV #3: ✗
AV #4: ✓	AV #5: ✓	AV #6: ✓
AV #7: ✗	AV #8: ✓	AV #9: ✓

Aggregated Score: 7/9 (SELECTED)

**Output #3:** Here's how to solve the problem step-by-step: ... The answer is (C)

AV #1: ✗	AV #2: ✗	AV #3: ✗
AV #4: ✗	AV #5: ✗	AV #6: ✗
AV #7: ✗	AV #8: ✓	AV #9: ✗

Aggregated Score: 1/9

**Output #4:** Here's how to solve the problem step-by-step: ... The answer is (A)

AV #1: ✓	AV #2: ✗	AV #3: ✗
AV #4: ✓	AV #5: ✗	AV #6: ✓
AV #7: ✓	AV #8: ✗	AV #9: ✗

Aggregated Score: 4/9

# Multi-Agent Verification

Generator LLM	MATH			MMLU-Pro			GPQA (diamond)			HumanEval		
	B-MAV	Cons	RM	B-MAV	Cons	RM	B-MAV	Cons	RM	B-MAV	Cons	RM
<b>Gemini-1.5-Flash</b>	<b><u>66.0</u></b>	59.0	61.7	<b><u>66.7</u></b>	63.3	60.7	42.0	40.0	<b><u>46.0</u></b>	<b><u>80.0</u></b>	79.0	79.0
<b>Gemini-1.5-Pro</b>	<b><u>72.7</u></b>	70.3	71.0	<b><u>72.3</u></b>	71.7	69.3	<b><u>49.0</u></b>	45.0	<b><u>49.0</u></b>	<b><u>88.0</u></b>	84.0	<b><u>88.0</u></b>
<b>GPT-4o-mini</b>	73.0	<b><u>74.7</u></b>	72.3	<b><u>67.0</u></b>	63.7	62.7	<b><u>50.0</u></b>	48.0	44.0	84.0	<b><u>87.0</u></b>	85.0
<b>GPT-4o</b>	76.3	77.3	<b><u>80.7</u></b>	75.7	<b><u>76.3</u></b>	72.7	<b><u>59.0</u></b>	<b><u>59.0</u></b>	58.0	92.0	<b><u>95.0</u></b>	92.0
<b>Mistral-7B</b>	<b><u>26.0</u></b>	22.0	21.7	<b><u>36.7</u></b>	25.7	31.0	36.0	32.0	<b><u>37.0</u></b>	<b><u>59.0</u></b>	46.0	52.0
<b>Llama-3.1-8B</b>	<b><u>61.7</u></b>	61.0	54.7	<b><u>59.3</u></b>	55.3	51.3	<b><u>43.0</u></b>	36.0	41.0	<b><u>75.0</u></b>	62.0	64.0
<b>Gemma-2-9B</b>	<b><u>58.7</u></b>	51.7	55.0	<b><u>57.7</u></b>	54.3	54.7	34.0	36.0	<b><u>38.0</u></b>	32.0	25.0	<b><u>51.0</u></b>
<b>Gemma-2-27B</b>	<b><u>62.3</u></b>	55.7	59.3	<b><u>62.0</u></b>	58.3	60.0	<b><u>41.0</u></b>	40.0	<b><u>41.0</u></b>	<b><u>76.0</u></b>	66.0	<b><u>76.0</u></b>

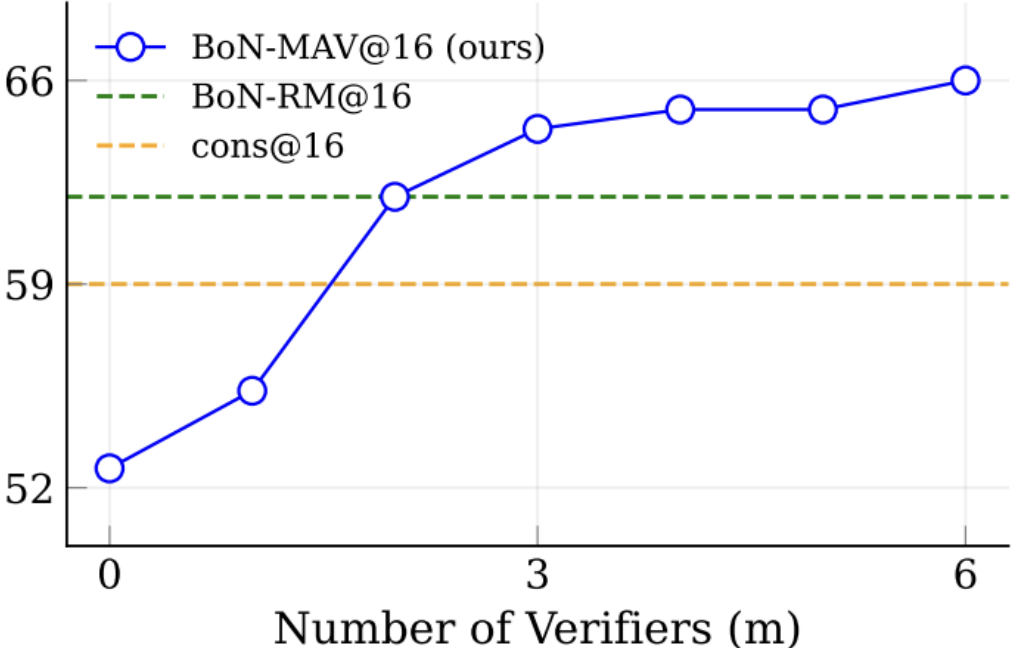
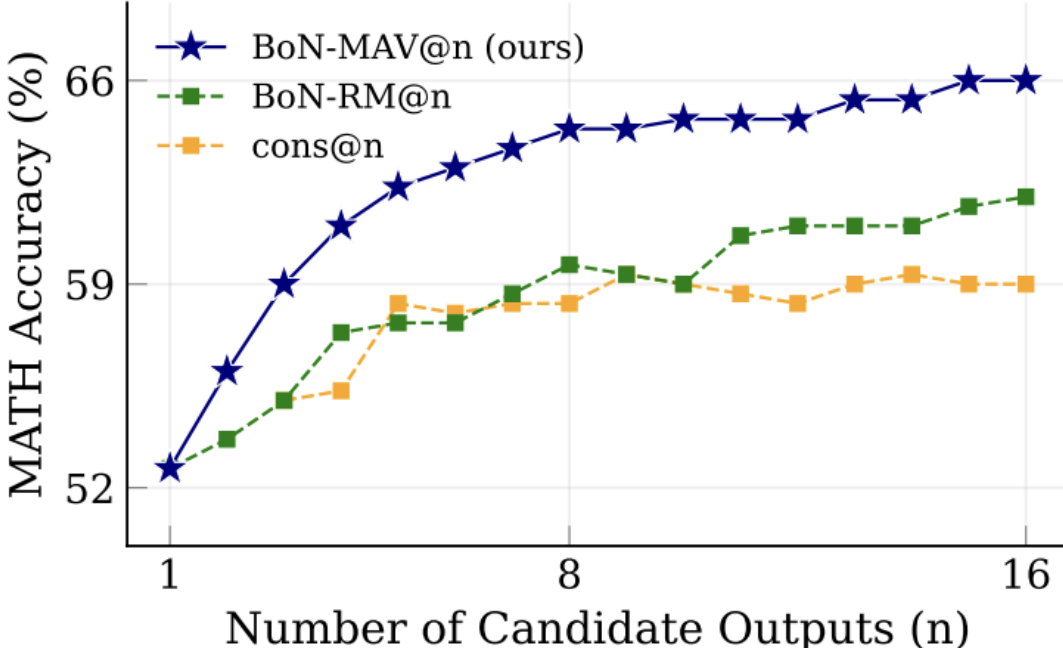
Table 1: **Best-of- $n$  with Multi-Agent Verification (BoN-MAV) across models and domains.** Performance (accuracy %) comparison of three test-time verification methods using  $n = 16$  candidate outputs: the BoN-MAV algorithm (labeled as B-MAV in the table), reward model verification (RM), and self-consistency (Cons). Results are shown for eight generator LLMs across four domains, with BoN-MAV on each domain using the domain-specific aspect verifier subset  $\mathcal{M}^d$ . BoN-MAV outperforms self-consistency in nearly all cases, and generally outperforms RM except on GPQA (diamond) and HumanEval, where BoN-MAV and RM achieve comparable results.

# Multi-Agent Verification

Generator LLM	MATH			MMLU-Pro			GPQA (diamond)			HumanEval		
	B-MAV	Cons	RM	B-MAV	Cons	RM	B-MAV	Cons	RM	B-MAV	Cons	RM
<b>Gemini-1.5-Flash</b>	<b><u>66.0</u></b>	59.0	61.7	<b><u>66.7</u></b>	63.3	60.7	42.0	40.0	<b><u>46.0</u></b>	<b><u>80.0</u></b>	79.0	79.0
<b>Gemini-1.5-Pro</b>	<b><u>72.7</u></b>	70.3	71.0	<b><u>72.3</u></b>	71.7	69.3	<b><u>49.0</u></b>	45.0	<b><u>49.0</u></b>	<b><u>88.0</u></b>	84.0	<b><u>88.0</u></b>
<b>GPT-4o-mini</b>	73.0	<b><u>74.7</u></b>	72.3	<b><u>67.0</u></b>	63.7	62.7	<b><u>50.0</u></b>	48.0	44.0	84.0	<b><u>87.0</u></b>	85.0
<b>GPT-4o</b>	76.3	77.3	<b><u>80.7</u></b>	75.7	<b><u>76.3</u></b>	72.7	<b><u>59.0</u></b>	<b><u>59.0</u></b>	58.0	92.0	<b><u>95.0</u></b>	92.0
<b>Mistral-7B</b>	<b><u>26.0</u></b>	22.0	21.7	<b><u>36.7</u></b>	25.7	31.0	36.0	32.0	<b><u>37.0</u></b>	<b><u>59.0</u></b>	46.0	52.0
<b>Llama-3.1-8B</b>	<b><u>61.7</u></b>	61.0	54.7	<b><u>59.3</u></b>	55.3	51.3	<b><u>43.0</u></b>	36.0	41.0	<b><u>75.0</u></b>	62.0	64.0
<b>Gemma-2-9B</b>	<b><u>58.7</u></b>	51.7	55.0	<b><u>57.7</u></b>	54.3	54.7	34.0	36.0	<b><u>38.0</u></b>	32.0	25.0	<b><u>51.0</u></b>
<b>Gemma-2-27B</b>	<b><u>62.3</u></b>	55.7	59.3	<b><u>62.0</u></b>	58.3	60.0	<b><u>41.0</u></b>	40.0	<b><u>41.0</u></b>	<b><u>76.0</u></b>	66.0	<b><u>76.0</u></b>

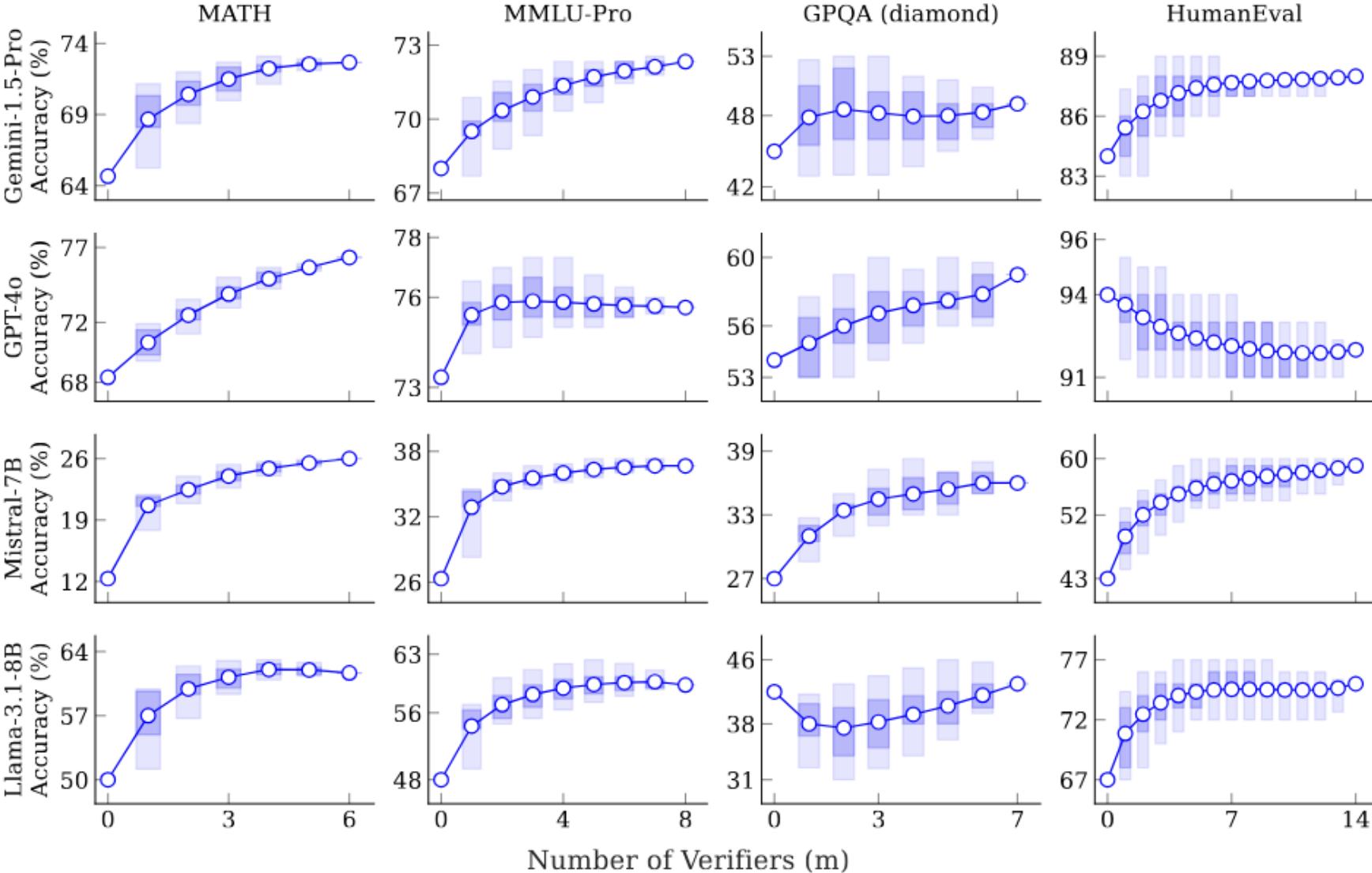
Table 1: **Best-of- $n$  with Multi-Agent Verification (BoN-MAV) across models and domains.** Performance (accuracy %) comparison of three test-time verification methods using  $n = 16$  candidate outputs: the BoN-MAV algorithm (labeled as B-MAV in the table), reward model verification (RM), and self-consistency (Cons). Results are shown for eight generator LLMs across four domains, with BoN-MAV on each domain using the domain-specific aspect verifier subset  $\mathcal{M}^d$ . BoN-MAV outperforms self-consistency in nearly all cases, and generally outperforms RM except on GPQA (diamond) and HumanEval, where BoN-MAV and RM achieve comparable results.

# Multi-Agent Verification



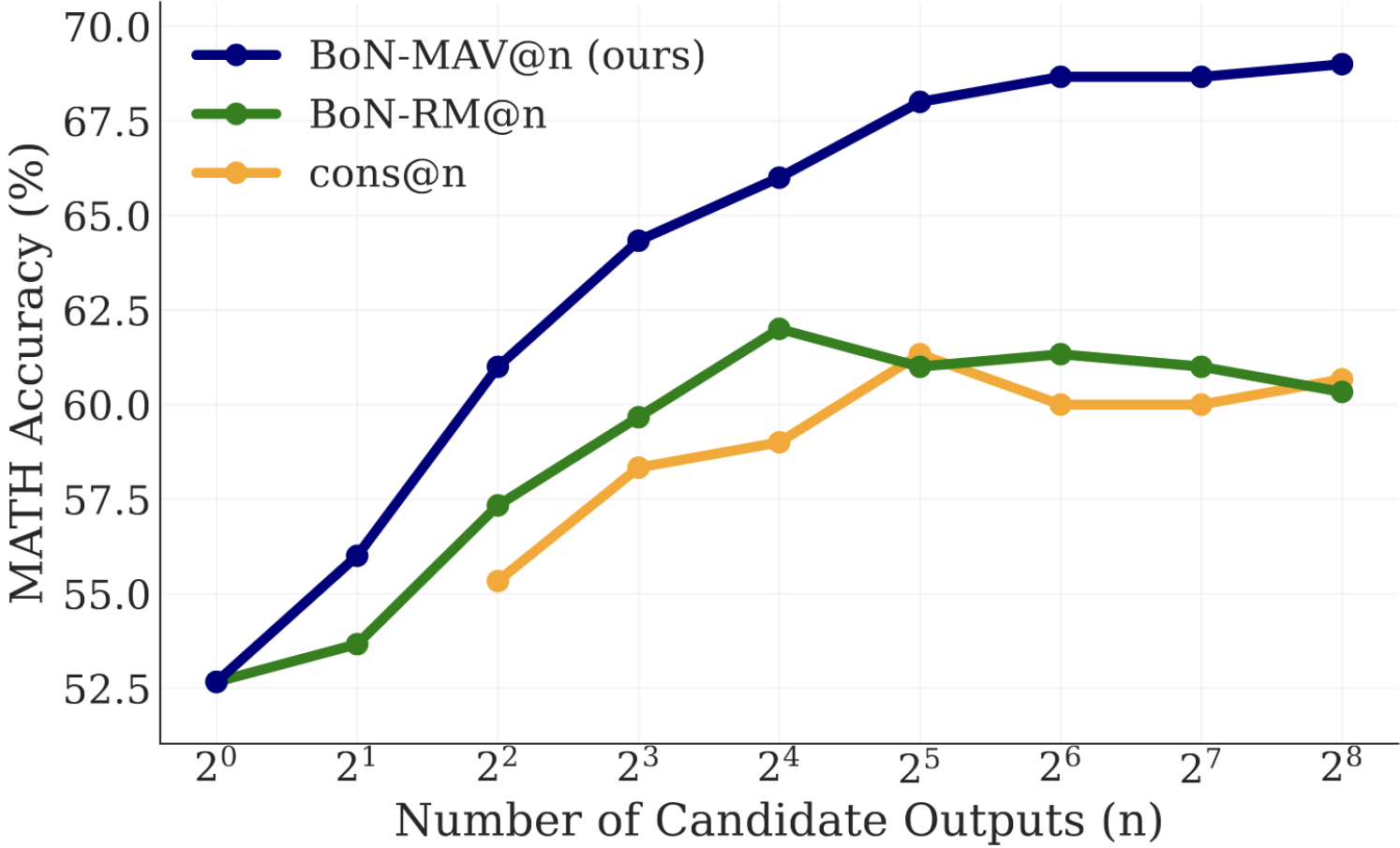
[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Scaling with More Composed Verifiers



[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Better Test-Time Scaling with Multi-Agent Verification



[1] Lifshitz et al. Multi-Agent Verification: Scaling Test-Time Computer with Multiple Verifiers. CoLM 2025

# Multi-Agent Verification with Pairwise Tournaments

**Question (HMMT)**  
Compute the number of ways to arrange 1,2,3,4,5,6,7 around a circle such that the product of every pair of adjacent numbers is at most 20. (Rotations and reflections are different.)

**Solution A (INCORRECT)**  
Generated answer: 008

- Determine adjacencies: 7 can only be adjacent to 1 and 2. Forbidden pairs: (3,7), (4,6), (4,7), (5,6), (5,7), (6,7).
- Fix  $a_1 = 7$ . Neighbors must be {1,2}. Case A: 1-7-2, Case B: 2-7-1.
- Arrange {3,4,5,6} in middle slots. 6 must be at an end, adjacent to 3. 4 valid orderings per case.
- Total:  $4 + 4 = 8$ .

**Solution B (CORRECT)**  
Generated answer: 056

- Determine adjacencies: 7 can only be adjacent to 1 and 2. Forbidden pairs: (3,7), (4,6), (4,7), (5,6), (5,7), (6,7).
- Place 7 in any of 7 positions. Its neighbors must be {1,2}: 2 orderings. So far:  $7 \times 2 = 14$  ways.
- Arrange {3,4,5,6} in remaining slots. 6 must be at an end, adjacent to 3. 4 valid orderings of the block.
- Total:  $14 \times 4 = 56$ .

Pairwise verification: 5 aspect verifiers compare both solutions

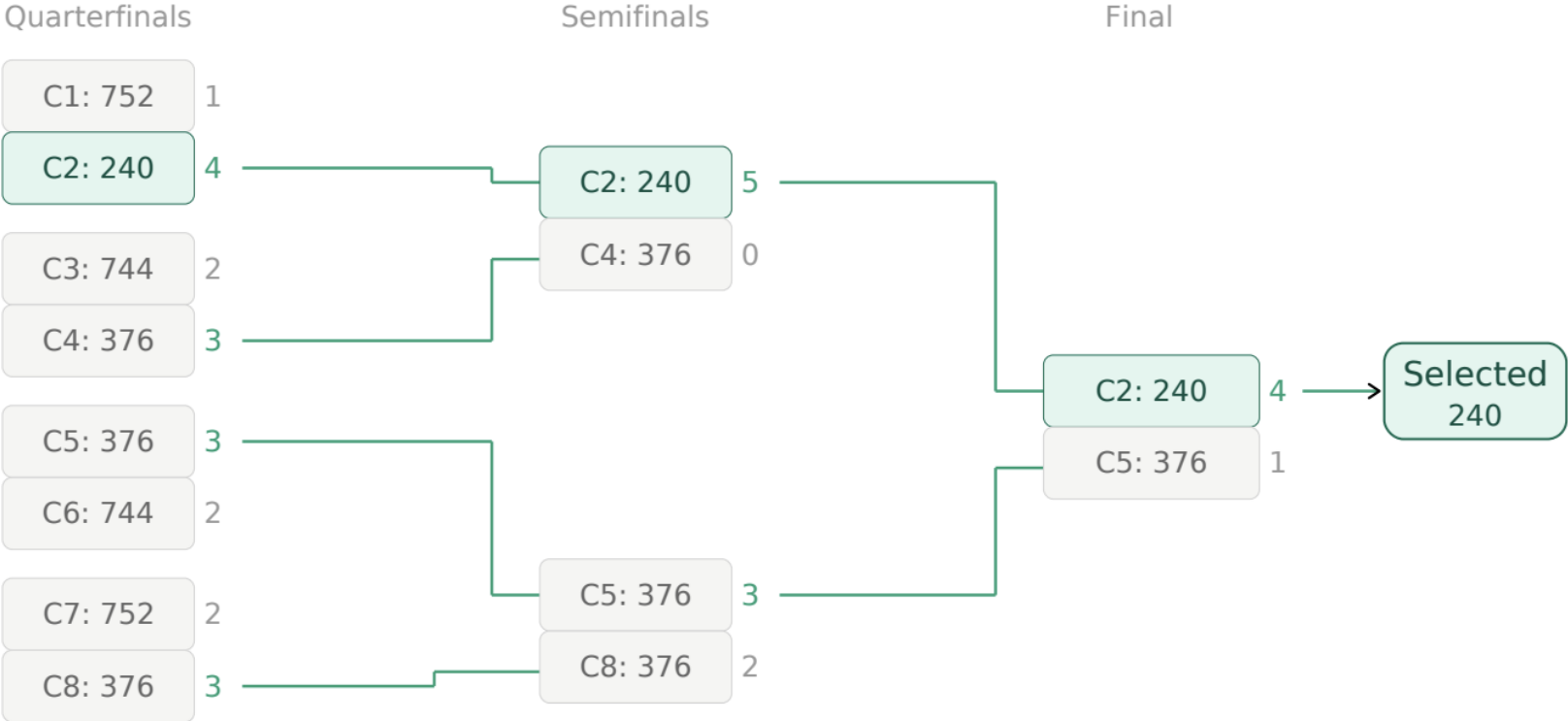
Verifier 1 GPT-5-mini Units check	Verifier 2 GPT-5-mini Summarize	Verifier 3 GPT-5-mini Mistakes	Verifier 4 GPT-5-mini Domain know.	Verifier 5 GPT-5-mini Edge cases
Neither solution introduces physical units. Both apply the product inequality $ab \leq 20$ consistently to dimensionless numbers. No unit errors in either.	Solution A fixes $a_1=7$ and finds 8 arrangements but makes a conceptual counting error: it does not multiply by 7 for distinct rotations. B correctly gets 56.	Both correctly find adjacencies and the 4 valid chains. But A counts only 8, which is off by factor of 7. A treated rotations as equivalent when they are distinct.	The error is in the final counting step. A omits the factor for distinct rotations. B correctly places 7 in any of 7 positions and multiplies to get 56.	B handles the rotational boundary correctly. A misses the key counting factor and so mishandles the rotation boundary condition.
NO PREF.	PREFER B	PREFER B	PREFER B	PREFER B

[1] Guan et al. Pairwise Verification: Scaling Test-Time Compute with Preference Tournaments. In Submission.

# Multi-Agent Verification with Pairwise Tournaments

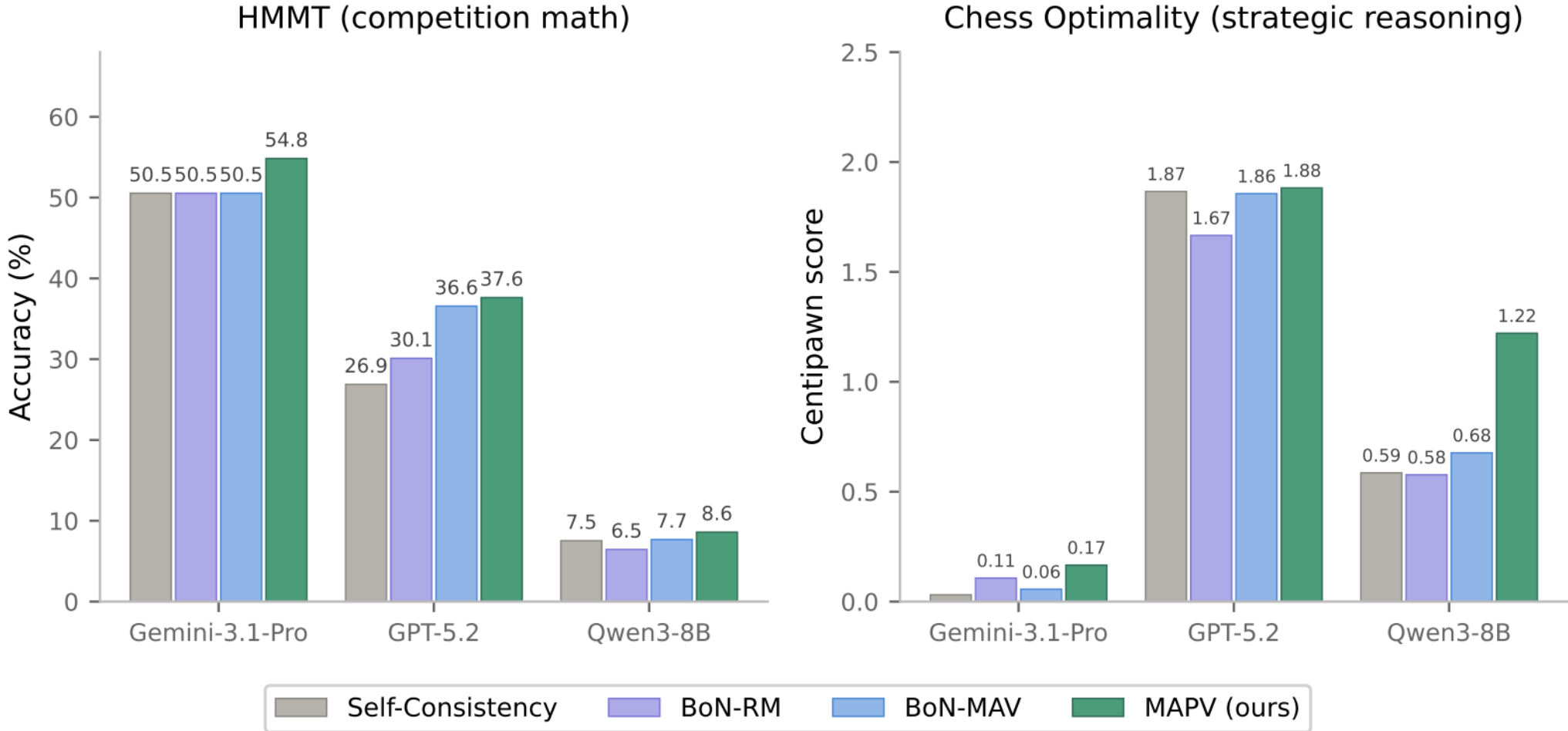
**Question**  
 Find the sum of three values of  $k$  such that  $f(x) = (x-18)(x-72)(x-98)(x-k)/x$  achieves its minimum at exactly two positive real numbers. Correct answer: 240

**Tournament aggregation (single elimination)**  
 Matchup winners advance. The process repeats until one candidate remains.



[1] Guan et al. Pairwise Verification: Scaling Test-Time Compute with Preference Tournaments. In Submission.

# Multi-Agent Verification with Pairwise Tournaments



[1] Guan et al. Pairwise Verification: Scaling Test-Time Compute with Preference Tournaments. In Submission.

# Talk Outline

## Multiagent Debate

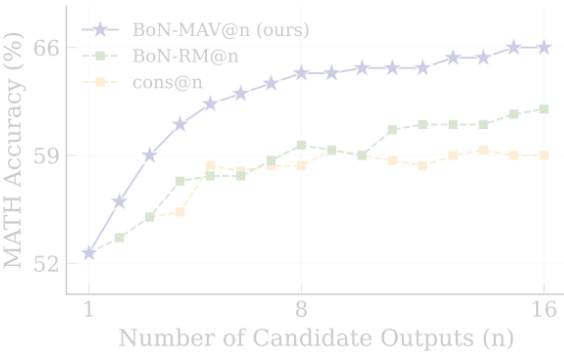
*Using the solutions from other agents, give an updated response* **Prompt**  
<copy Agent 1 and Agent 2 responses>

**Agent 1** Given the information provided in other agents' solutions ... + 245 = 560. Our final answer is **560**. ❌

**Agent 2** After reviewing the solutions provided by other agents ... my answer to the math problem is **595**. ✅

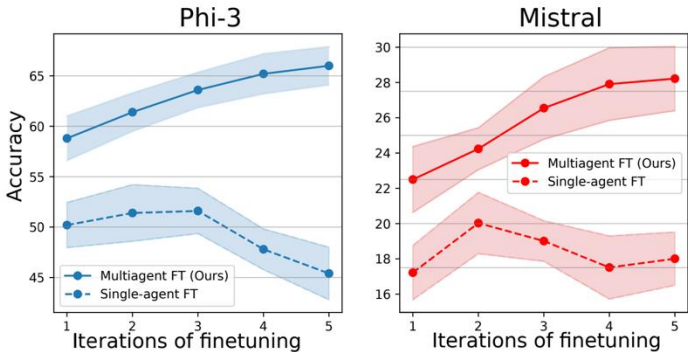
## Iterative Communication Between Language Models

## Multiagent Verification



## Combining Language Models through Verification

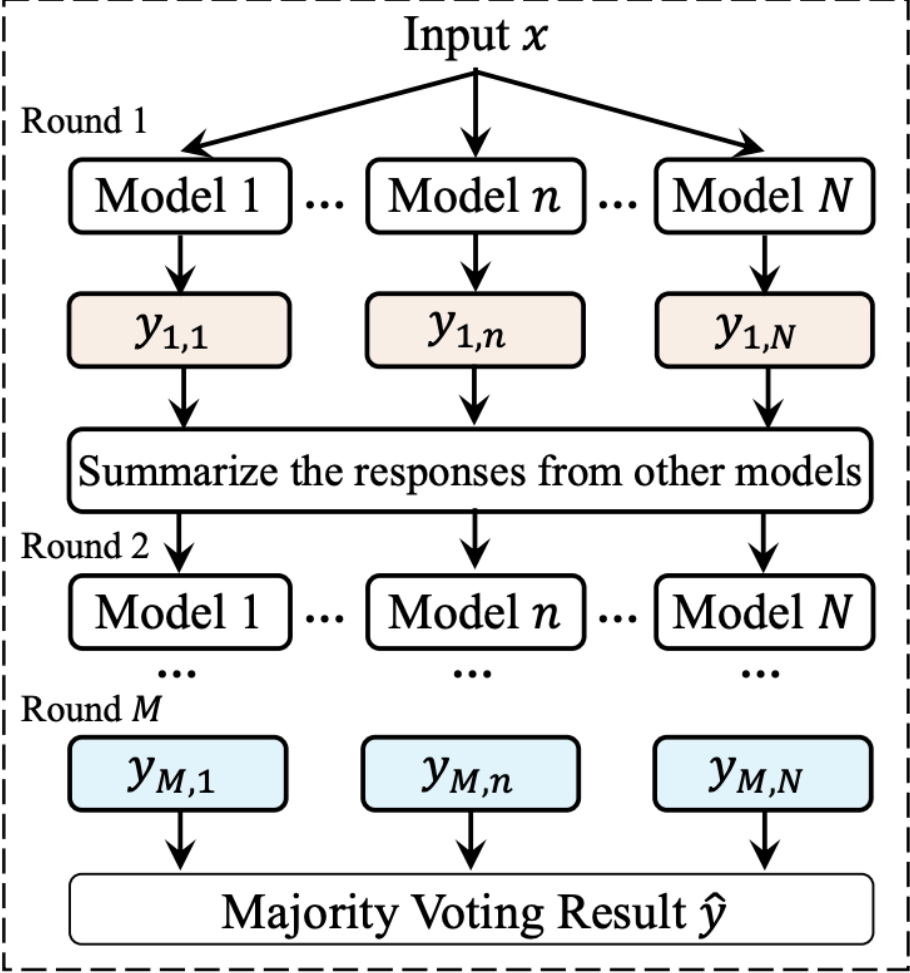
## Multiagent Self-Improvement



## Finetuning Compositions of Language Models

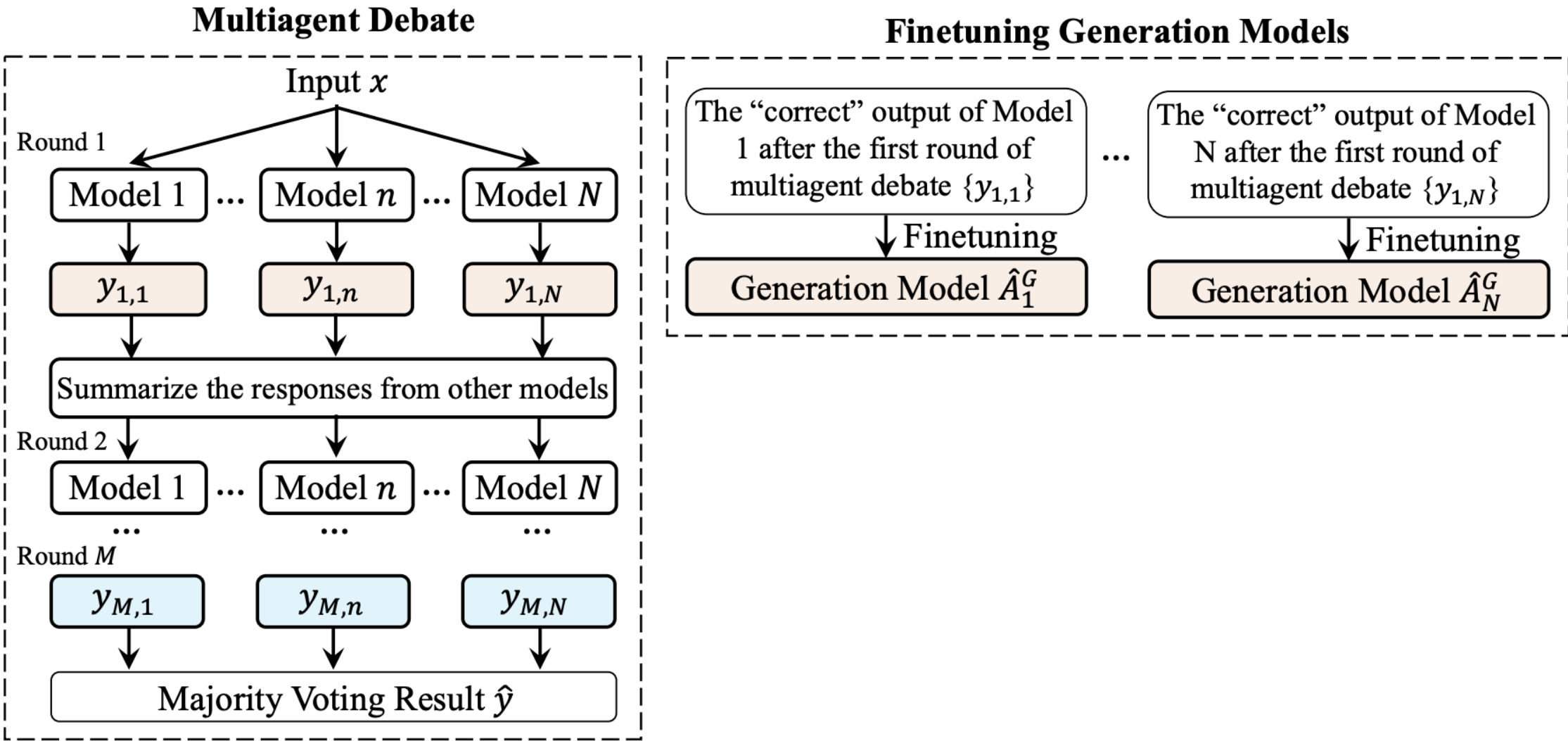
# Finetuning and Specializing Debate Models

## Multiagent Debate



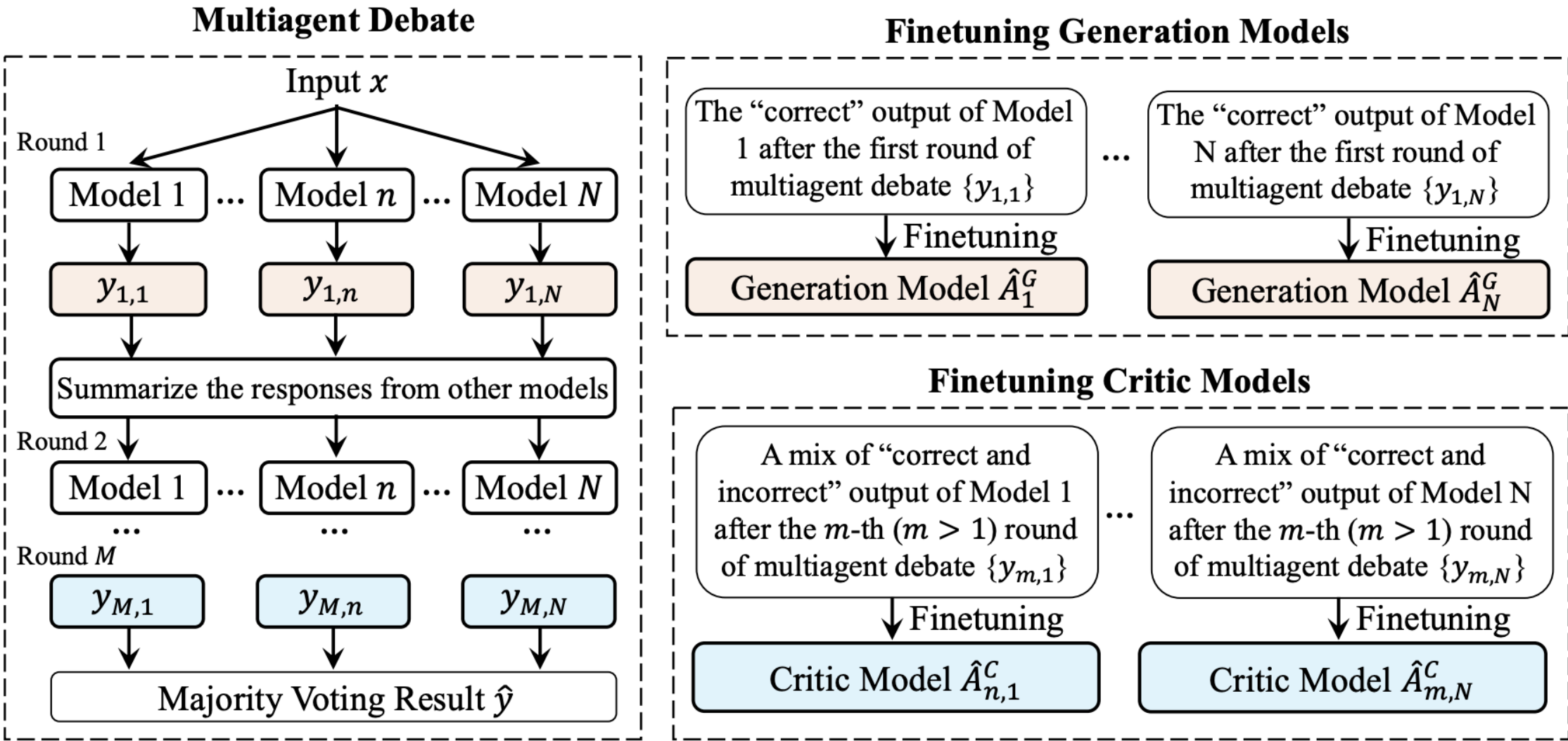
[1] Subramaniam et al. Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains. ICLR 2025

# Finetuning and Specializing Debate Models



[1] Subramaniam et al. Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains. ICLR 2025

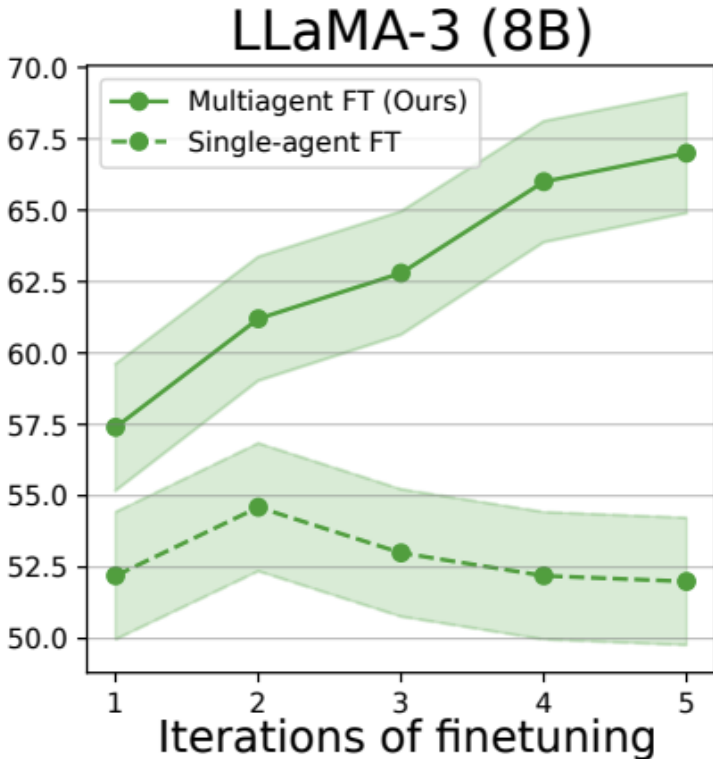
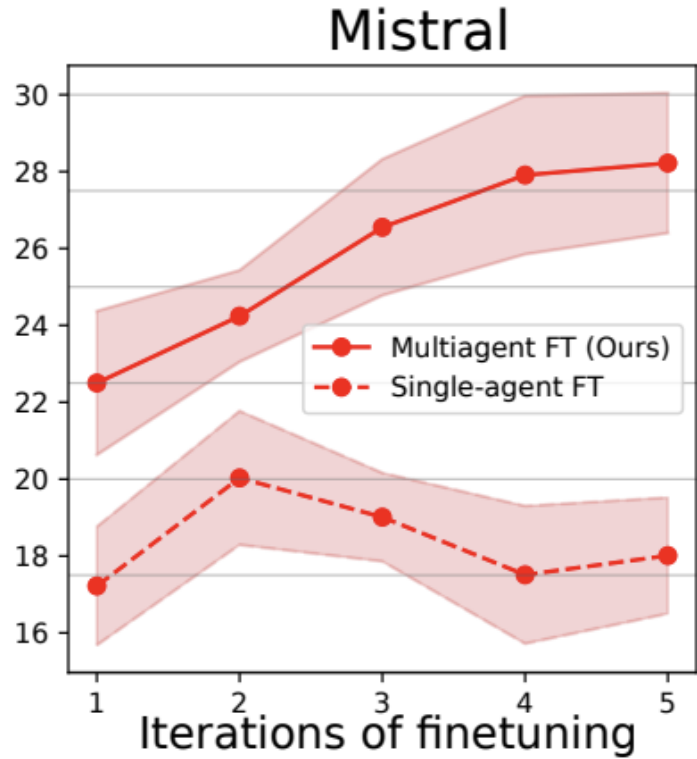
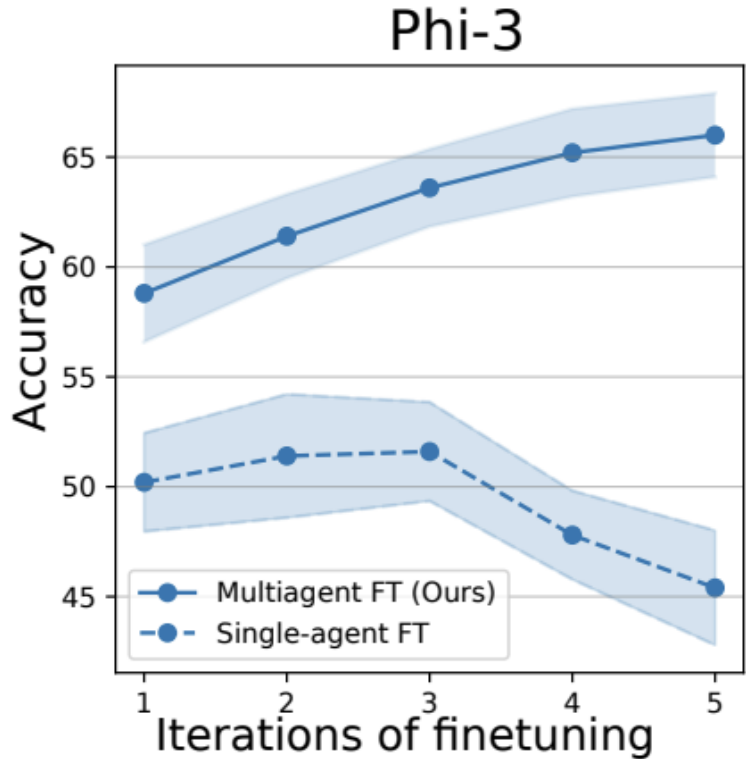
# Finetuning and Specializing Debate Models



[1] Subramaniam et al. Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains. ICLR 2025

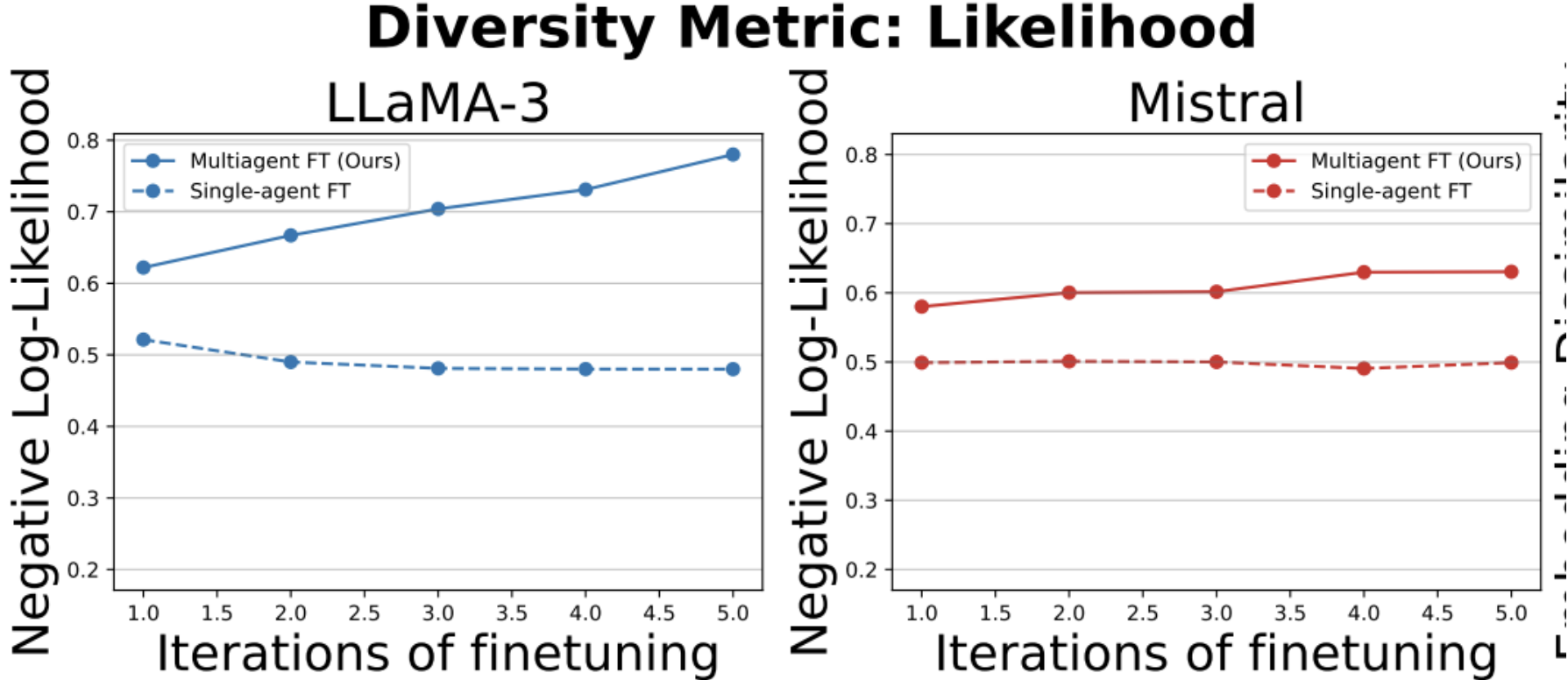


# Self-Improvement over Rounds of Finetuning



[1] Subramaniam et al. Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains. ICLR 2025

# Self-Improvement over Rounds of Finetuning



[1] Subramaniam et al. Multiagent Finetuning: Self Improvement with Diverse Reasoning Chains. ICLR 2025

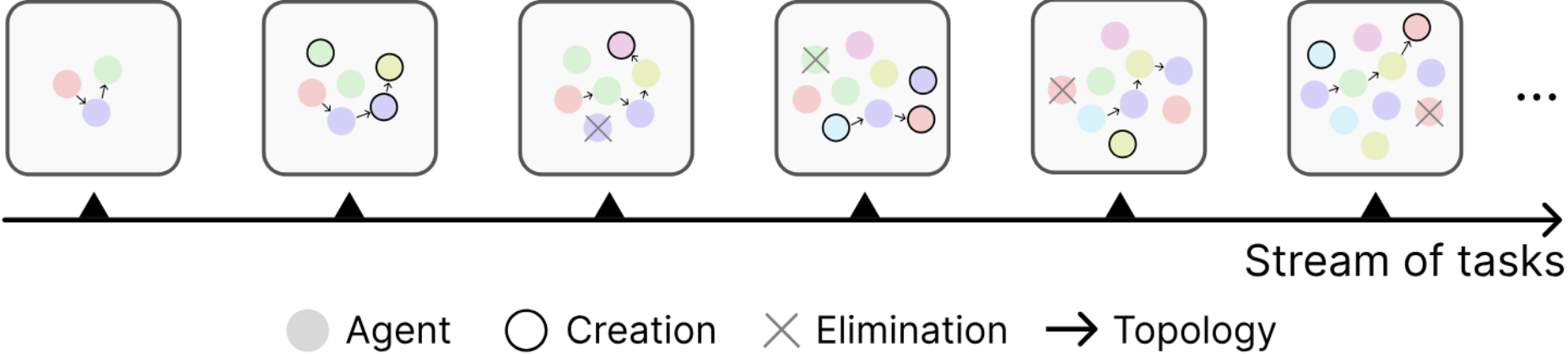
# Self-Improvement with Multiagent Finetuning

LLM	Methods	Arithmetic	GSM	MATH
GPT-3.5 (OpenAI, 2022)	Base	81.99 ± 0.99	75.60 ± 1.36	46.83 ± 2.25
	Majority	94.40 ± 1.03	81.20 ± 1.24	51.40 ± 2.23
	Debate	98.21 ± 0.54	83.30 ± 1.18	55.73 ± 2.21
	STaR	98.38 ± 0.57	83.60 ± 1.17	53.00 ± 2.23
	Majority FT	98.40 ± 0.56	83.70 ± 1.17	53.40 ± 2.23
	Ours	<b>99.62 ± 0.28</b>	<b>85.60 ± 1.11</b>	<b>60.60 ± 2.18</b>
Phi-3 (Abdin et al., 2024)	Base	88.30 ± 1.44	81.20 ± 1.74	45.60 ± 2.10
	Majority	91.80 ± 1.23	81.80 ± 1.72	47.20 ± 1.82
	Debate	96.20 ± 0.86	84.40 ± 1.58	53.40 ± 2.28
	STaR	94.80 ± 0.99	85.80 ± 1.21	51.80 ± 2.06
	Majority FT	93.80 ± 1.08	82.20 ± 1.71	48.60 ± 2.16
	Ours	<b>99.40 ± 0.34</b>	<b>88.60 ± 1.42</b>	<b>58.80 ± 2.22</b>
Mistral (Jiang et al., 2023)	Base	10.80 ± 0.51	35.60 ± 1.92	16.60 ± 1.21
	Majority	14.80 ± 1.17	41.80 ± 0.88	16.80 ± 1.25
	Debate	19.60 ± 1.12	52.60 ± 1.26	18.20 ± 1.37
	STaR	17.40 ± 0.97	45.50 ± 1.54	17.84 ± 1.23
	Majority FT	16.40 ± 0.73	44.60 ± 1.65	18.91 ± 1.37
	Ours	<b>22.60 ± 0.97</b>	<b>58.40 ± 2.11</b>	<b>22.50 ± 1.87</b>
LLaMA-3 (Dubey et al., 2024)	Base	43.20 ± 2.22	75.00 ± 1.94	46.80 ± 2.23
	Majority	45.80 ± 2.23	76.40 ± 1.90	47.20 ± 2.23
	Debate	48.40 ± 2.24	78.40 ± 1.44	51.60 ± 2.23
	Majority FT	49.20 ± 2.24	77.20 ± 1.87	52.20 ± 2.23
	Ours	<b>52.00 ± 2.24</b>	<b>88.60 ± 1.77</b>	<b>57.40 ± 2.21</b>

# Self-Improvement over Rounds of Finetuning

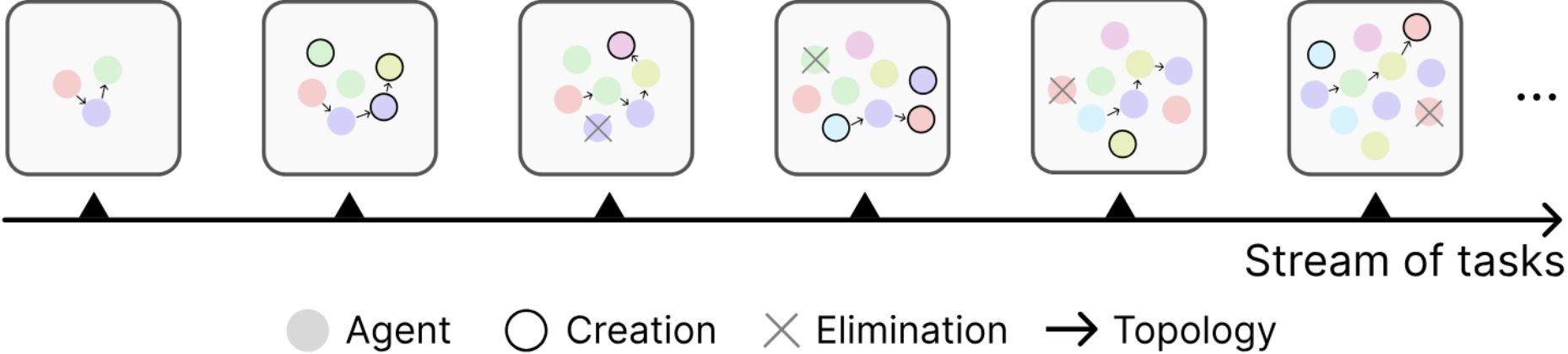
LLM	Methods	Arithmetic	GSM	MATH
GPT-3.5 (OpenAI, 2022)	Base	81.99 ± 0.99	75.60 ± 1.36	46.83 ± 2.25
	Majority	94.40 ± 1.03	81.20 ± 1.24	51.40 ± 2.23
	Debate	98.21 ± 0.54	83.30 ± 1.18	55.73 ± 2.21
	STaR	98.38 ± 0.57	83.60 ± 1.17	53.00 ± 2.23
	Majority FT	98.40 ± 0.56	83.70 ± 1.17	53.40 ± 2.23
	Ours	<b>99.62 ± 0.28</b>	<b>85.60 ± 1.11</b>	<b>60.60 ± 2.18</b>
Phi-3 (Abdin et al., 2024)	Base	88.30 ± 1.44	81.20 ± 1.74	45.60 ± 2.10
	Majority	91.80 ± 1.23	81.80 ± 1.72	47.20 ± 1.82
	Debate	96.20 ± 0.86	84.40 ± 1.58	53.40 ± 2.28
	STaR	94.80 ± 0.99	85.80 ± 1.21	51.80 ± 2.06
	Majority FT	93.80 ± 1.08	82.20 ± 1.71	48.60 ± 2.16
	Ours	<b>99.40 ± 0.34</b>	<b>88.60 ± 1.42</b>	<b>58.80 ± 2.22</b>
Mistral (Jiang et al., 2023)	Base	10.80 ± 0.51	35.60 ± 1.92	16.60 ± 1.21
	Majority	14.80 ± 1.17	41.80 ± 0.88	16.80 ± 1.25
	Debate	19.60 ± 1.12	52.60 ± 1.26	18.20 ± 1.37
	STaR	17.40 ± 0.97	45.50 ± 1.54	17.84 ± 1.23
	Majority FT	16.40 ± 0.73	44.60 ± 1.65	18.91 ± 1.37
	Ours	<b>22.60 ± 0.97</b>	<b>58.40 ± 2.11</b>	<b>22.50 ± 1.87</b>
LLaMA-3 (Dubey et al., 2024)	Base	43.20 ± 2.22	75.00 ± 1.94	46.80 ± 2.23
	Majority	45.80 ± 2.23	76.40 ± 1.90	47.20 ± 2.23
	Debate	48.40 ± 2.24	78.40 ± 1.44	51.60 ± 2.23
	Majority FT	49.20 ± 2.24	77.20 ± 1.87	52.20 ± 2.23
	Ours	<b>52.00 ± 2.24</b>	<b>88.60 ± 1.77</b>	<b>57.40 ± 2.21</b>

# Self-Improvement through Economic Interactions



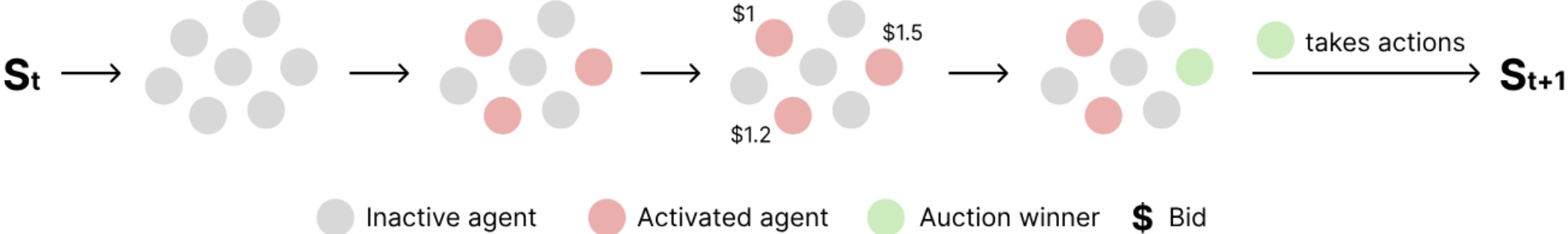
[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preparation

# Self-Improvement through Economic Interactions



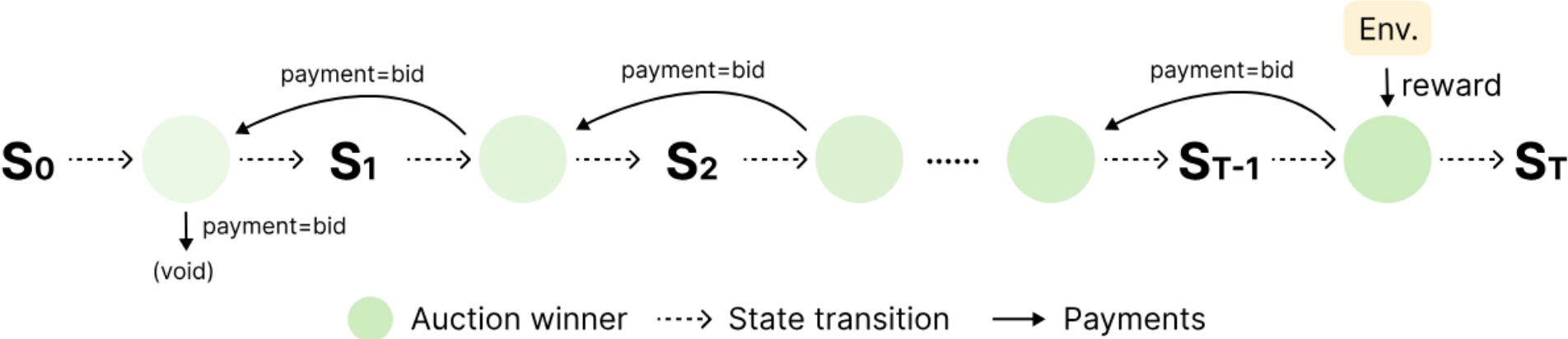
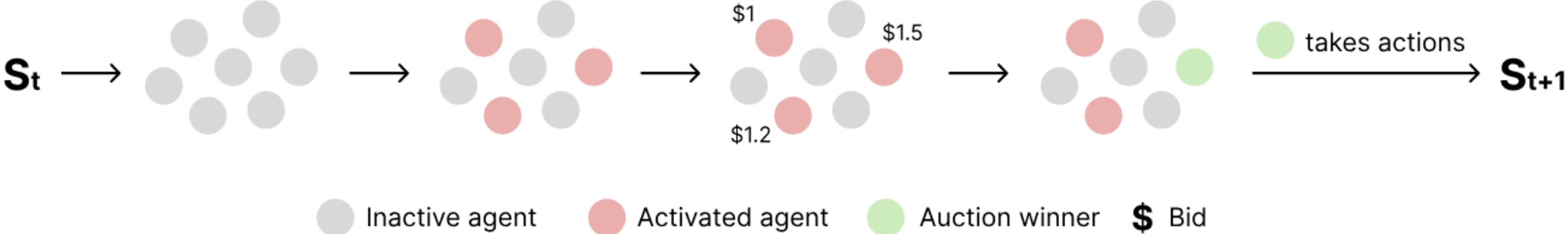
[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preparation

# Self-Improvement through Economic Interactions



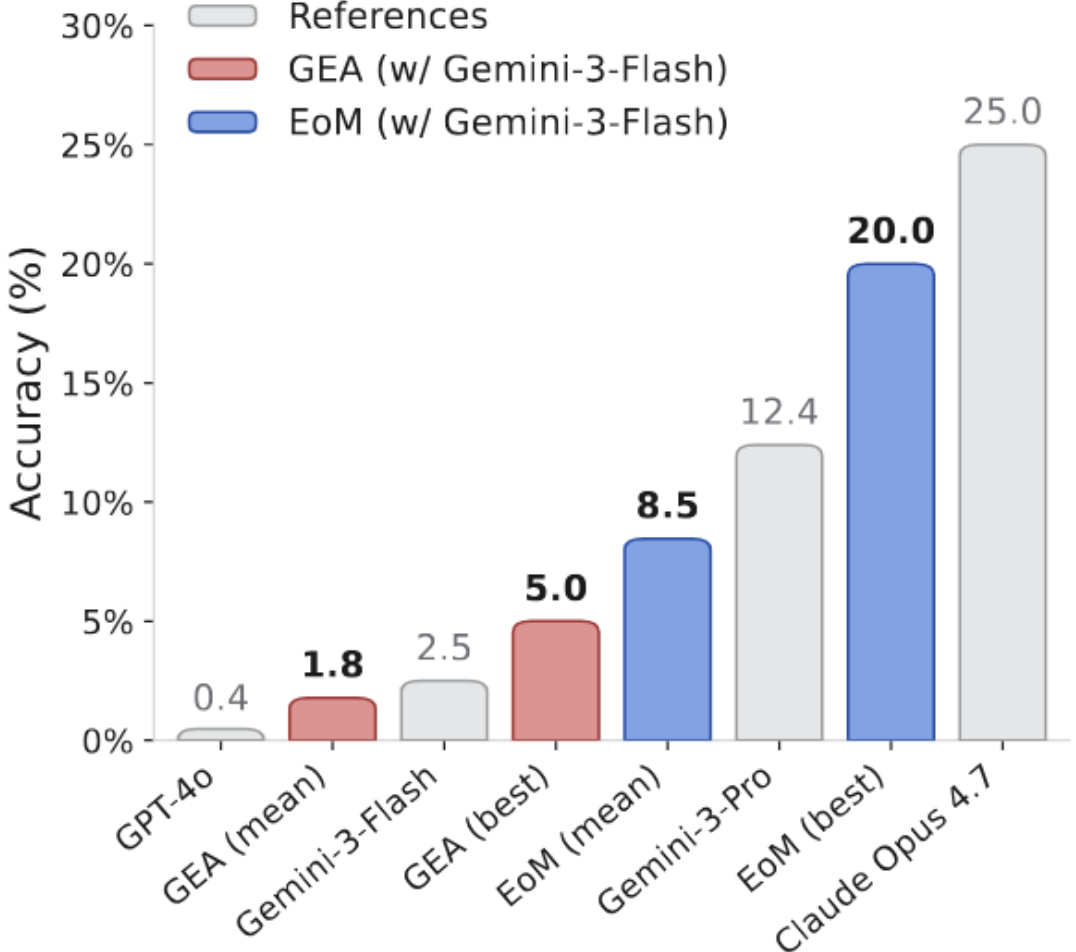
[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preparation

# Self-Improvement through Economic Interactions



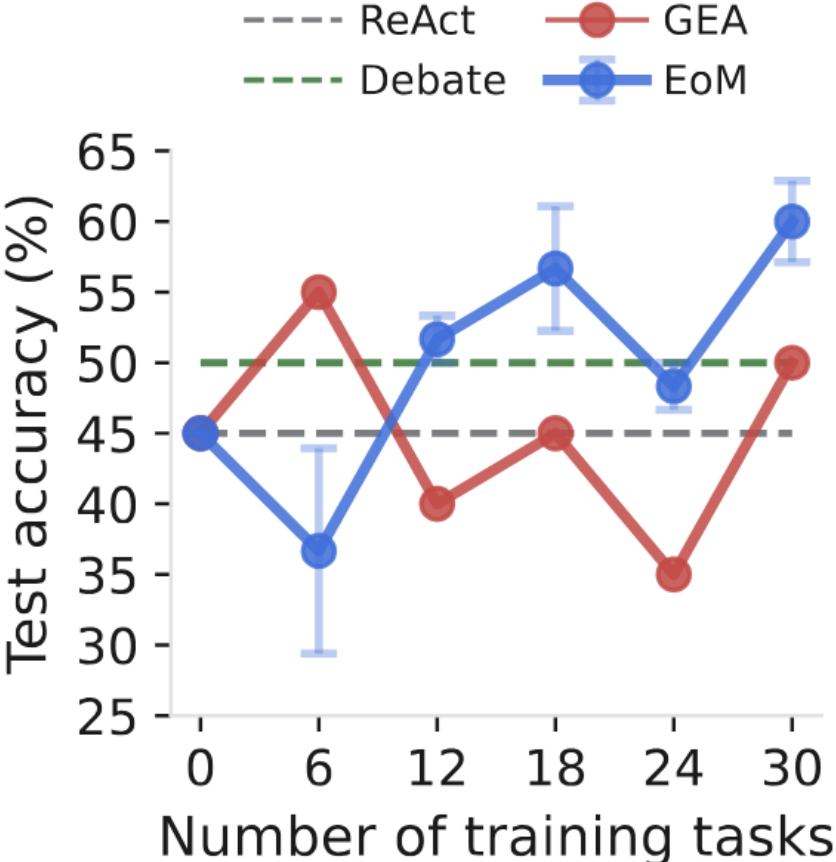
[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preparation

# FrontierScience Research Agent



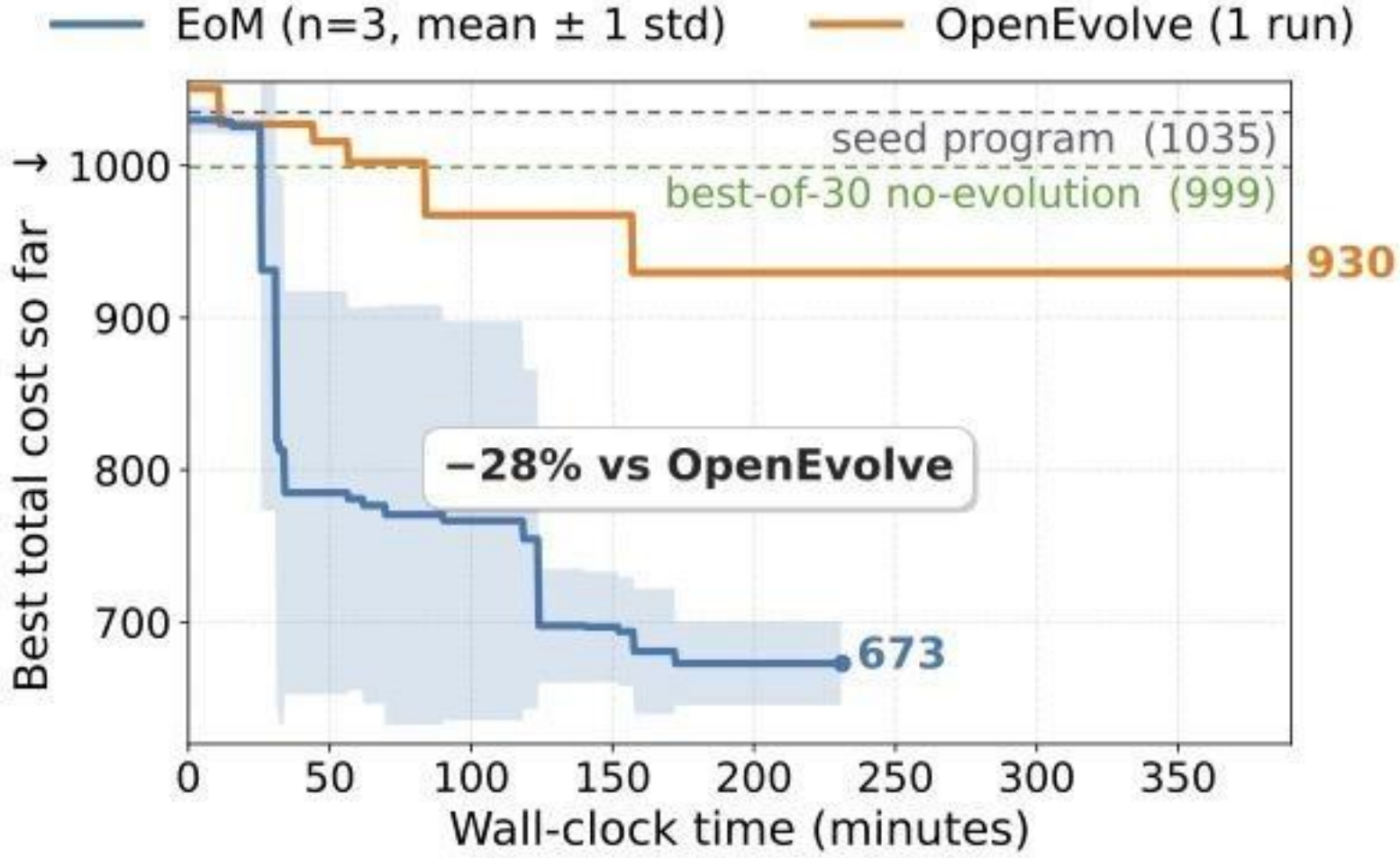
[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preperation

# Self-Improvement in Finance Bench



[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preperation

# Iteratively Improving Data Transfer Cost



[1] Qi et al. Economy of Minds: Emerging Multi-Agent Intelligence with Economic Interactions. In Preperation

# Talk Outline

## Multiagent Debate

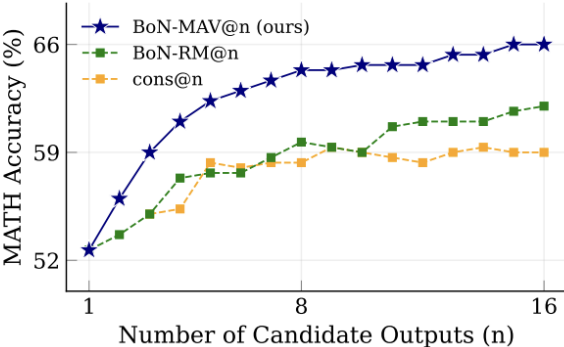
*Using the solutions from other agents, give an updated response* **Prompt**  
<copy Agent 1 and Agent 2 responses>

**Agent 1** Given the information provided in other agents' solutions ... + 245 = 560. Our final answer is **560**. ❌

**Agent 2** After reviewing the solutions provided by other agents ... my answer to the math problem is **595**. ✅

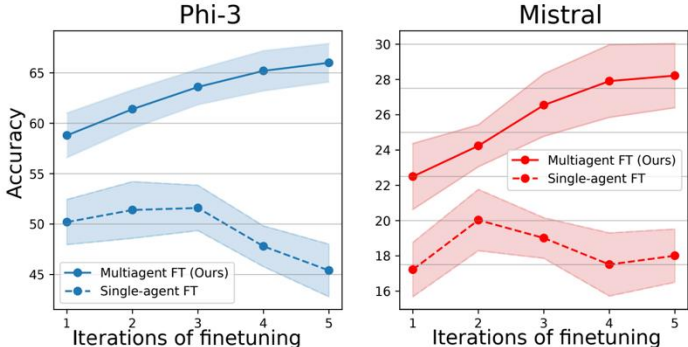
Iterative Communication  
Between Language  
Models

## Multiagent Verification



Combining Language Models  
through Verification

## Multiagent Self-Improvement



Finetuning Compositions of  
Language Models

# Multiagent Scientific Reasoning Systems

Yilun Du  
May 2026



**HARVARD**  
UNIVERSITY



**Kempner**  
INSTITUTE