

CS395T - Debate

Allen Jue

September 18, 2025

What to learn

- Introduction & Motivation
 - Early multi-agent debate
 - Why divergent thinking matter in LLMs
- Symbolic Mixture of Experts
 - Mixture-of-Experts (MoE)
 - Skill-based routing

Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate

Liang et al. (EMNLP 2024)

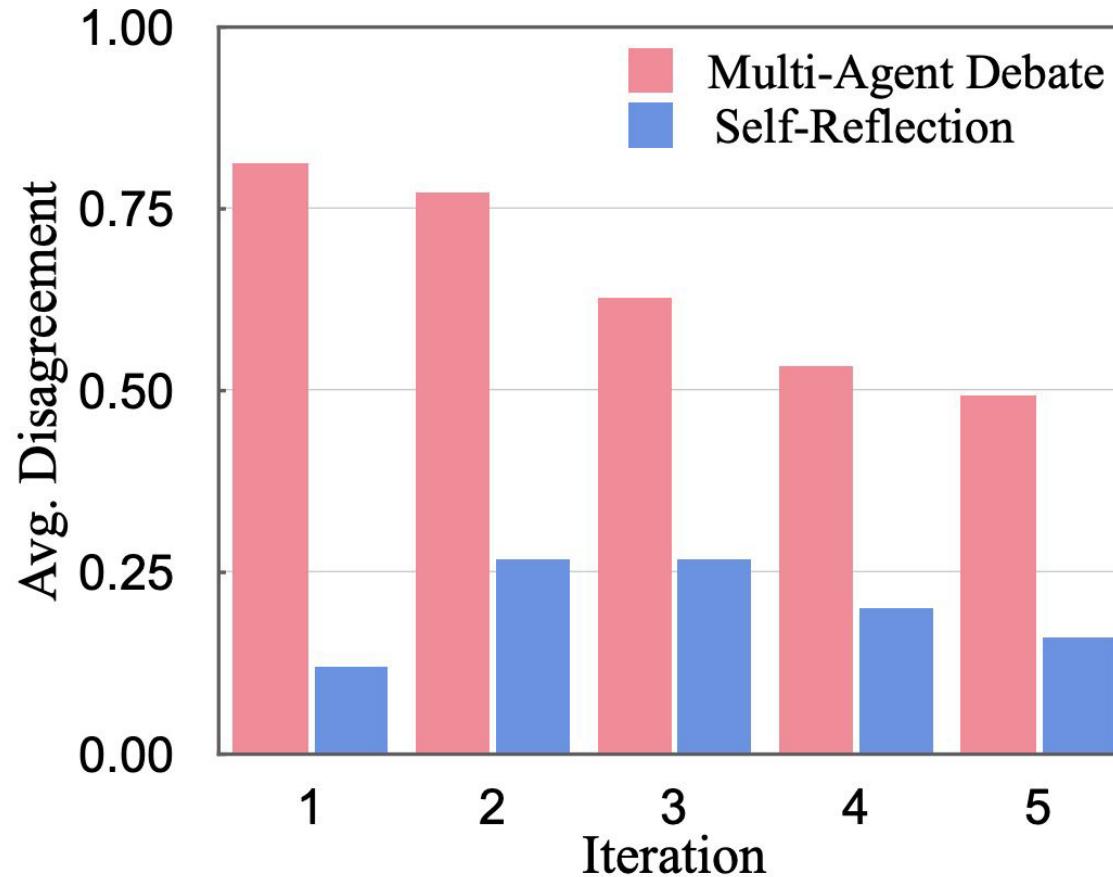
October 2024

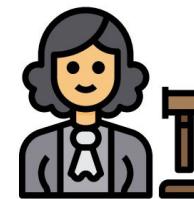
Motivation: Degeneration-of-Thought (DoT)

- LLMs can iteratively refine ideas through self-reflection
 - Chain-of-thought (Wei et al., 2022)
- Degeneration-of-Thought (DoT)
 - Once confident, an LLM can not generate novel ideas
 - Debate maintains higher disagreement → promotes novel reasoning.
- Factors causing DoT:
 1. Bias and distorted perception.
 2. Rigidity and resistance to change.
 3. Limited external feedback.

Solution: Multi-Agent Debate (MAD) framework

1. Corrects distorted thinking through agent interaction.
2. Overcomes resistance by complementing agents.
3. Provides external feedback via inter-agent exchange.





MAD Framework

- Meta prompts
 - *What to debate?*
 - *How to debate (affirm or dissent)?*
 - *How much to debate?*

$$D = \{D_i\}_{i=1}^N$$

$J_d(H)$ = True or False

MAD Framework

Level

Prompt

0

Both sides must reach a full consensus on every point of the debate. Every statement must be agreed upon by both sides.

1

Most of the debate should be characterized by disagreements, but there may still be a small amount of consensus on less significant points.

2 (Default)

It's not necessary to fully agree with each other's perspectives, as our objective is to find the correct answer.

3

Both sides must disagree with each other on every point of the debate. There should be no consensus whatsoever.

Experiment

- Challenging Testbeds
 - Common sense machine translation
 - Counterintuitive arithmetic reasoning
 - Zero-shot
- Comparisons and Evaluation
 - Self-reflect, rerank, CoT, etc.
 - Use accuracy for arithmetic and common translation evaluation techniques

Experiment: Common MT

Source	吃掉敌人一个师。
Correct Ref.	Destroy a division of the enemy.
Incorrect Ref.	Eat up an enemy division.
GPT-4	Eat up an enemy division.
GPT-3.5-Turbo	Eat up an enemy division.
+ Self-Reflect	Eat up an enemy division.
+ MAD	Eliminate an enemy division.

Experiment: Counter-intuitive AR

Components	Content
Question	When Alice walks up the hill, her speed is 1 m/s and when she goes down the hill, her speed is 3 m/s. Then when Alice walks up and down the hill, what is her average speed?
Correct Answer	1.5 m/s
Explanation	If Alice covers a distance of d going up and down the hill, then her total distance is $2d$. Her time going up the hill is $d/1 = d$, and her time going down the hill is $d/3$. So, her total time is $d + d/3 = 4d/3$. Therefore, her average speed is $2d / (4d/3) = 3/2$ m/s.
Incorrect Answer	2 m/s
Explanation	Alice's average speed can be calculated by adding her speed going up the hill and her speed going down the hill, and then dividing by 2. So, $(1 \text{ m/s} + 3 \text{ m/s}) / 2 = 2 \text{ m/s}$. Therefore, Alice's average speed is 2 m/s.

Table 9: An example in Counter-Intuitive AR dataset.

Experiment: Workflow

Counter-Intuitive Question

The two circles are externally tangent and there is no relative sliding. The radius of circle A is 1/3 the radius of circle B. Circle A rolls around circle B one trip back to its starting point. How many times will circle A revolve in total?

Chain-of-Thought

Let the radius of circle A be r and the radius of circle B be $3r$. The distance circle A travels along the circumference of circle B is equal to the circumference of circle B, which is $6\pi r$. The number of revolutions = $(6\pi r) / (2\pi r) = 6\pi r / 2\pi r = 3$. So, circle A will revolve 3 times in total as it rolls around circle B once.



Self-Reflection

Reflection:

- (1) I have a clear understanding of the problem.
- (2) I have used the correct formula for the circumference of a circle and applied it to both circles to find the number of revolutions.
- (3) I have correctly calculated the number of revolutions of circle A.

Based on the results of my self-reflection, I am confident that my answer is correct. Therefore, circle A will revolve 3 times in total as it rolls around circle B once.



Multi-Agent Debate



I disagree with you. To find the total number of revolutions, we need to consider both the rotation around circle B and the rotation of circle A itself. Therefore, circle A will revolve 3 times around its own center and 1 time around circle B, making a total of 4 revolutions.



I see your point, but...



That's a valid point, however...



The negative side correctly considers both the rotation of circle A around its own center and its rotation around circle B, while the affirmative side only considers the rotation around circle B. Therefore, the answer is 4.



Results: Common MT

Method	Lexical			Contextless			Contextual		
	COMET	BLEURT	HUMAN	COMET	BLEURT	HUMAN	COMET	BLEURT	HUMAN
GPT-4	82.0	70.1	3.41	84.7	73.6	3.63	85.0	73.7	3.65
Turbo	80.3	68.2	3.14	84.0	72.9	3.43	84.9	73.4	3.57
+ Rerank	80.9	68.6	3.16	84.5	73.2	3.46	85.3	73.9	3.58
+ MAPS	81.9	70.1	3.43	84.2	73.5	3.45	85.2	74.0	3.56
+ Self-Reflect	81.0	69.1	3.43	83.6	72.2	3.46	84.9	73.5	3.63
+ MAD	82.0	70.9	3.78	84.8	73.7	3.67	85.3	74.0	3.67
Vicuna-7b	74.9	62.0	2.55	78.3	64.6	2.53	80.2	68.2	3.23
+ MAD	75.6	62.6	2.67	78.6	66.0	2.69	81.8	69.9	3.27
Vicuna-13b	76.6	63.7	2.81	77.6	66.8	3.04	82.2	70.0	3.37
+ MAD	77.2	65.1	2.96	80.1	67.3	3.11	82.6	70.9	3.45

Table 1: Translation performance on Common MT. Note that Rerank and MAPS use the external quality estimation tool to select the best translation from multiple translation candidates. HUMAN: direct assessment of translation quality from human evaluators on a scale ranging from 1 to 5.

Method	Math Reasoning		Symbolic Reasoning (BBH)		
	GSM	AddSub	Penguin	Date	Colored Objects
CoT	70.2	87.3	58.9	56.4	57.2
Self-Reflect	70.8	87.6	61.0	58.0	58.0
MAD	73.8	92.1	63.7	65.2	58.8

Results: Counterintuitive AR

Method	ACC (%)
GPT-4	51.0
GPT-3.5-Turbo	26.0
+ CoT	28.0
+ Self-Consistency	29.5
+ Self-Reflect	27.5
+ MAD	37.0

Table 3: Accuracy on Counter-Intuitive AR.

Analysis

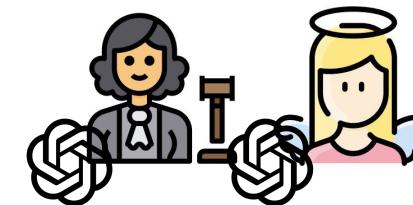
- Remember, we want to mitigate DoT (Bias and rigidity)

Method	Bias↓	Diversity↑
Self-Reflect	29.0	19.3
MAD	24.8	49.7

Table 4: Mitigation of Degeneration-of-Thought.

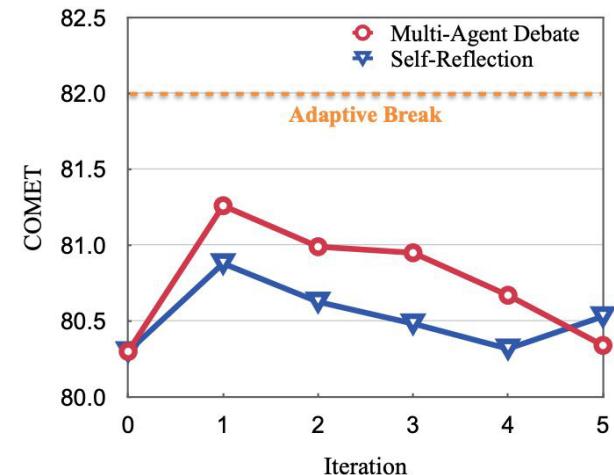
Analysis - Judges

- Strong debaters with weak judges > Weak debaters with strong judges.
- Judges tend to favor the negative side, which is beneficial since the affirmative tends to make mistakes
- Judges are biased towards the same type of LLM



Analysis - Debaters

- Performance is highly-dependent on the context capabilities of the backbone LLM
- Disagreement is good, but too much is harmful.
- For complex problems, multiple rounds are necessary
- Ending the debate early can be good.



Discussion

- MAD is an interesting framework that can improve weaker models by combating DoT.
- I think the improvements would be more interesting if applied to stronger models.
- It is not free – generates up to **2.46x** tokens + need to run multiple models.

Method	Cost (# Generated Tokens)
COT	1.0
Self-Reflect	1.83×
MAD	2.46×

Table 12: Extra generated tokens on the counter-intuitive AR task.

Symbolic Mixture-of-Experts: Adaptive Skill-based Routing for Heterogeneous Reasoning

Justin Chih-Yao Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, Mohit Bansal

March 2025

Motivation: Mixture of Experts (MoE)

- Humans experts work together to solve problems
- Why not machines?
 - Models can be specialized for certain problems... *at the expense of others*
 - Traditionally use sub-models as experts, requires training

Background

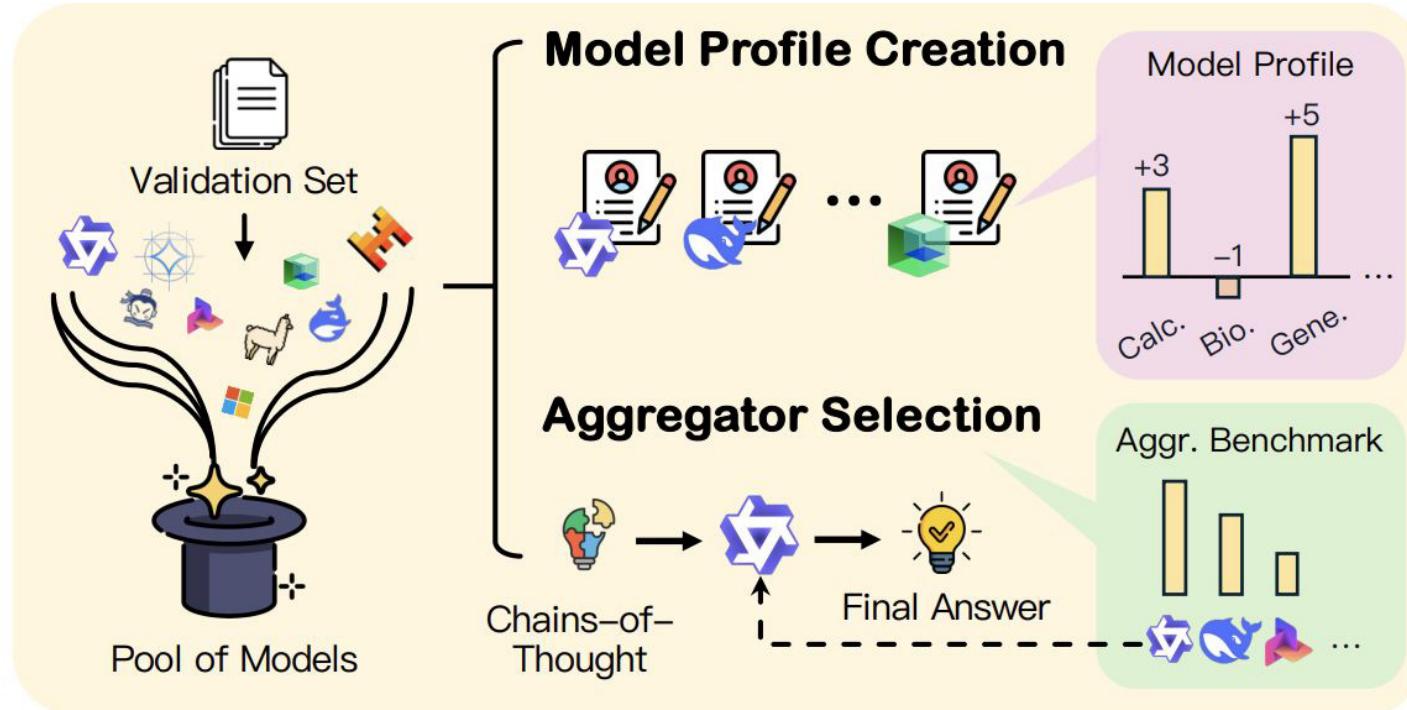
- Mixture of Experts (MoE) and Mixture of Agents (MoA)
 - Submodels trained jointly (Jacobs et al., 1991; Jordan & Jacobs, 1994)
 - Sparse MoE (Shazeer et al., 2017a) - activate what's needed
 - Ensemble of fixed LLM agents (Wang et al., 2024a)
 - **Costly, MoE needs training,**
- Multi-agent Reasoning
 - Working together can be good!

Solution: *Symbolic* MoE

- Cut out the training
- Use *symbolic* structures in the model's state:
 1. Measure the skills of existing models and routes appropriate experts.
 2. Use language as a channel for communication
- So this is good, but to do this:
 1. Efficient expert selection
 2. Scalable Expert Mixing

Preprocessing

$$M = \{M_i\}_{i=1}^n$$



Inference: Routing

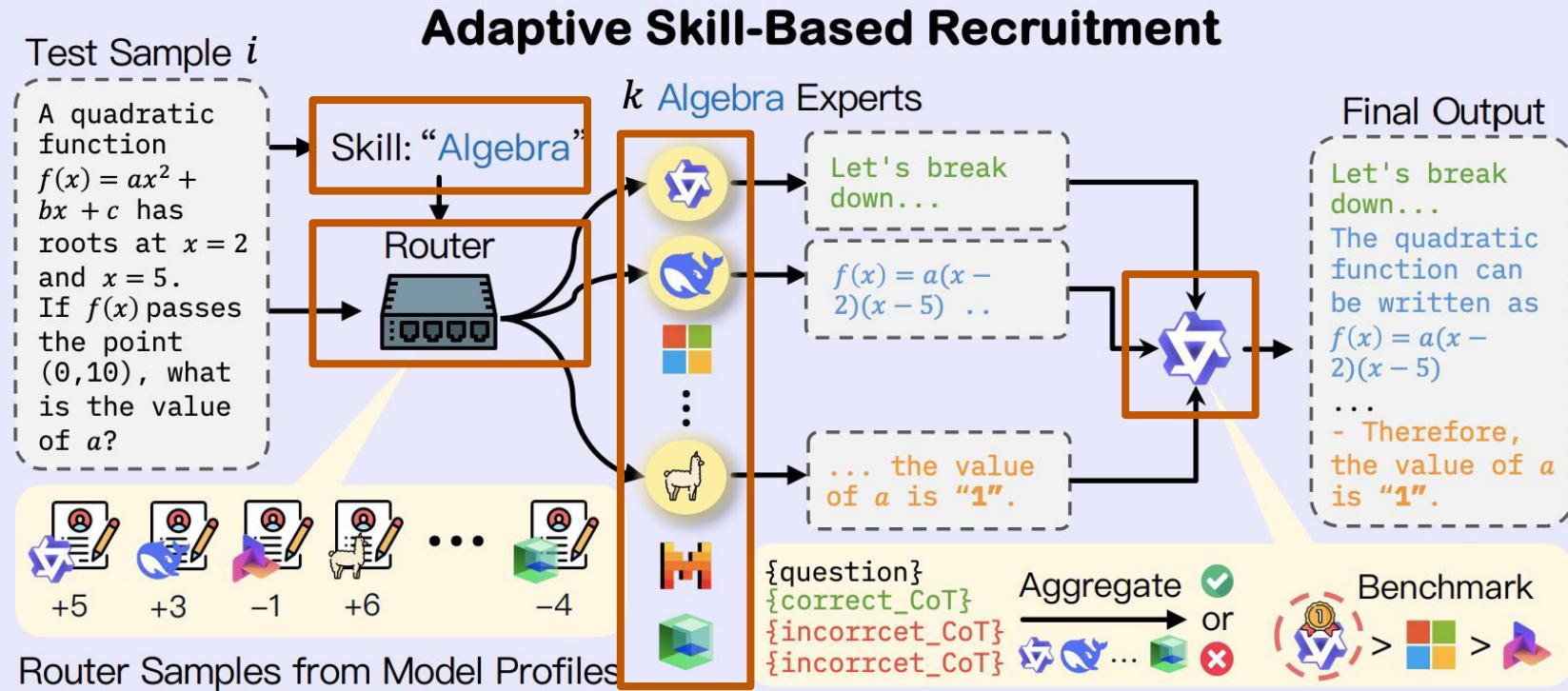
- Each model has a weighted competency

$$w_q^{(i)} = \gamma_i \mathcal{S}(M_i, q)$$

- Sample experts from this distribution

$$E_q^{(i)} \sim \text{Categorical}(w_q^{(1)}, w_q^{(2)}, \dots, w_q^{(n)}), \quad i = \{1, 2, \dots, k\}$$

Inference: All Together Now



Inference: Batching

(I) Parallel Inference w/ Multiple GPUs

Question	Expert 1	Expert 2	Expert k
How ...	Qwen	Llama	Mistral



Question	Expert 1	Expert 2	Expert k
A large ...	Gemma	Qwen	Phi



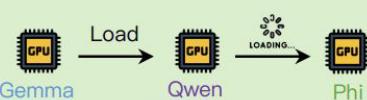
 Requires n GPUs at the same time

(II) Sequential Inference w/ Single GPU

Question	Expert 1	Expert 2	Expert k
How ...	Qwen	Llama	Mistral



Question	Expert 1	Expert 2	Expert k
A large ...	Gemma	Qwen	Phi



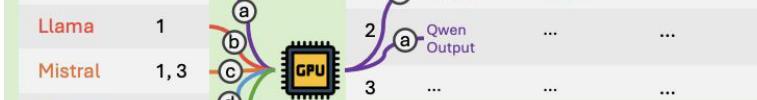
 Takes n times longer than using 1 model

(III) Batch Inference: Flexible to the Number of GPUs

ID	Question	Expert 1	Expert 2	Expert k
1	How many of the following ...	Qwen	Llama	Mistral
2	A large gene has dozens of ...	Qwen	Gemma	Phi
3	Consider the following ...	Mistral	Gemma	Phi

 Group by Experts

Expert	ID	Sequential Generation	ID	Expert 1	Expert 2	Expert k
Qwen	1, 2		1	Qwen Output	Llama Output	...
Llama	1		2	Qwen Output
Mistral	1, 3		3
Gemma	2, 3					
Phi	2, 3					



 Fits in a single GPU

 Can be accelerated w/ more GPUs

Experiments

- 16 LLMs ranging from 3.5B to 12B
- Tested on:
 - MMLU-Pro (Wang et al., 2024c)
 - AIME 2024
 - GPQA Diamond (Rein et al., 2023)
 - MedMCQA
 - **These are hard problems**

Results - Accuracy

Category	Method	Model	MMLU-Pro	AIME	GPQA	MedMCQA	Avg.
Close-Source Single Model	Zero-Shot CoT	GPT4o-mini	63.95	10.00	42.93	68.18	46.27
	Zero-Shot CoT	Gemini 1.5 Pro	76.38	36.67	61.62	72.68	61.84
	Zero-Shot CoT	DeepSeekV3	76.29	26.00	60.10	74.09	59.12
Open-Source 70B Model	Zero-Shot CoT	Qwen2.5 72B	71.54 ± 0.88	25.55 ± 3.85	51.02 ± 0.27	69.02 ± 0.32	54.28
	Zero-Shot CoT	Llama3.3 70B	69.26 ± 0.47	32.22 ± 3.85	51.44 ± 0.62	59.78 ± 0.74	53.18
Open-Source 7B Model	Zero-Shot CoT	QwenR1	52.57 ± 0.45	55.93 ± 5.16	44.95 ± 1.49	38.72 ± 0.44	48.04
	Zero-Shot CoT	Task-Best	54.89 ± 0.53	55.93 ± 5.16	48.43 ± 3.10	55.44 ± 0.50	53.62
Advanced Single Model	Self-Refine (SR)	Task-Best	53.74 ± 0.20	53.33 ± 3.34	50.84 ± 3.65	49.57 ± 0.59	51.87
	Self-Consistency (SC)	Task-Best x5	56.71 ± 0.14	67.78 ± 1.57	53.54 ± 0.36	56.85 ± 0.11	58.72
Single-Model Multi-Agent	Debate	Task-Best x3	56.21 ± 0.55	56.67 ± 6.67	50.51 ± 0.51	51.63 ± 0.80	53.76
	Self-MoA	Task-Best x3	55.43 ± 0.72	55.56 ± 5.09	52.86 ± 1.46	53.27 ± 0.60	54.28
Multi-Model Multi-Agent	MoA	Top-3	61.78 ± 0.25	41.11 ± 5.09	52.86 ± 3.37	59.29 ± 0.32	53.76
	ReConcile	Top-3	56.46 ± 0.10	50.00 ± 7.20	47.98 ± 2.32	60.74 ± 0.43	53.80
	SYMBOLIC-MOE	Adaptive	63.71 ± 0.43	68.88 ± 5.08	57.78 ± 2.09	59.35 ± 0.14	62.43

Results - Accuracy

- **Consistently outperforms all baselines**
 - Single model (using strongest), multimodal, and multi-agent.
- **Generalizes well across tasks.**
- **Matches strong proprietary models and larger 70B models.**
 - Use only 4 models (3 experts and 1 aggregator)

Results - Efficiency

Method	# GPUs	Run Time (s)
Sequential Inference	1	196.92
MoA	1	45.98
MoA	4	21.66
SYMBOLIC-MoE	1	25.76
SYMBOLIC-MoE	4	10.85

Results - Discussion-based

Discuss	Aggr.	MMLU-Pro	GPQA
✓	Adaptive	59.07	57.01
✗	Adaptive	57.12	58.01
✓	Task-best	57.81	57.78
✗	Task-best	56.67	57.01
✓	Task-specific	63.83	57.72
✗	Task-specific	<u>63.71</u>	<u>57.78</u>

Results - Additional Analysis

- **Sampling k relevant experts is the best.**
 - Random k experts is really bad
 - Top-5 is worse than Top-3
 - Top- n is worse than sampling
- **Generalizes to unseen tasks.**
 - OmniMATH (Gao et al., 2025)
 - MATH Hard (Hendrycks et al., 2021b)
- **Role and selection of the aggregator**
 - Good reasoner \neq Good aggregator

Results - Additional Analysis

- **Sampling k relevant experts is the best.**
 - Random k experts is really bad
 - Top-5 is worse than Top-3
 - Top- n is worse than sampling
- **Generalizes to unseen tasks.**
 - OmniMATH (Gao et al., 2025)
 - MATH Hard (Hendrycks et al., 2021b)
- **Role and selection of the aggregator**
 - Good reasoner \neq Good aggregator

Results - Additional Analysis

- **Synergy between expert and aggregator selection**
 - Strong experts and task-level aggregator 
 - Random aggregator or voting (no aggregator) 
 - Weak experts 
- **Sensitivity to the keyword LLM**
 - Not very important

Discussion

- Symbolic MoE is:
 - Consistent
 - Modular
 - Inference-time scaling
- Limitations:
 - Distillation - needs multiple models, can incorporate discussions into a single model
 - Relies on keyword LLM

Improving LLM Reasoning and Factuality: Debate and RECONCILE

Aditya S Ellendula

September 18, 2025

The Promise of LLMs

- Large Language Models (LLMs) → impressive at text generation, reasoning, problem-solving.
- Unlock new capabilities in math, coding, commonsense reasoning.
- But... still prone to serious flaws.
- **LLMs are strong but systematically flawed:** they *look* like good reasoners, but often fail at multi-step reasoning and hallucinate with high confidence.

What Was Tried Before

“Large Language Models are Zero-Shot Reasoners”

- Showed that LLMs have **latent zero-shot reasoning abilities** that can be unlocked with a single generic prompt: “Let’s think step by step”
- Establishes Zero-shot CoT as the “**minimal strongest zero-shot baseline**”.

Limitations:

- Unreliable Prompt Sensitivity
- Lack of True Reasoning called “**cognitive mirage**” or “**illusion of reasoning**”
- Overconfidence & Hallucination
- Inconsistency Across Runs
- No mechanism to handle variability

Therefore → **Self-Consistency**: aggregate over many reasoning paths to stabilize and improve results.

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. **X**

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. **✓**

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 **X**

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

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

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. **✓**

What Was Tried Before

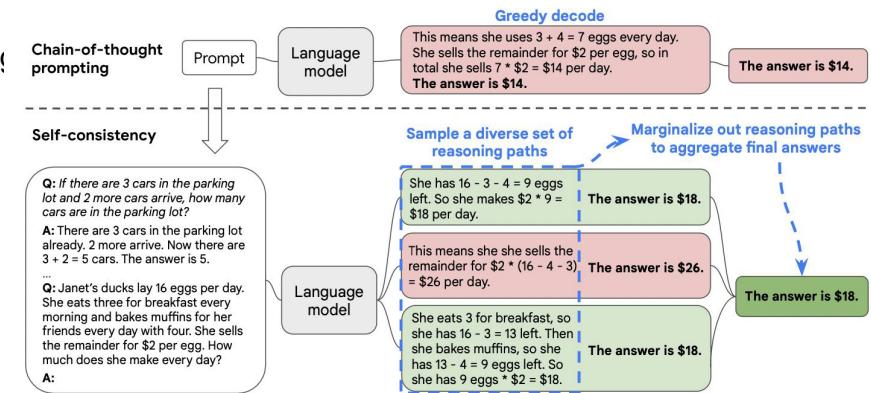
“Self-Consistency”

- Instead of relying on a single chain-of-thought (greedy decoding), sample *multiple diverse reasoning paths*.
- Aggregate answers via **majority voting** (self-ensemble)
- Task-agnostic, Unsupervised “self-ensemble” → works off-the-shelf, more robust to prompt errors and imperfect rationales.

Limitations:

- Computational cost
- Still passive
- Answer-level focus
- No learning from mistakes

Next step → **Self-Refine** = active, iterative refinement where the model learns from its own critiques.



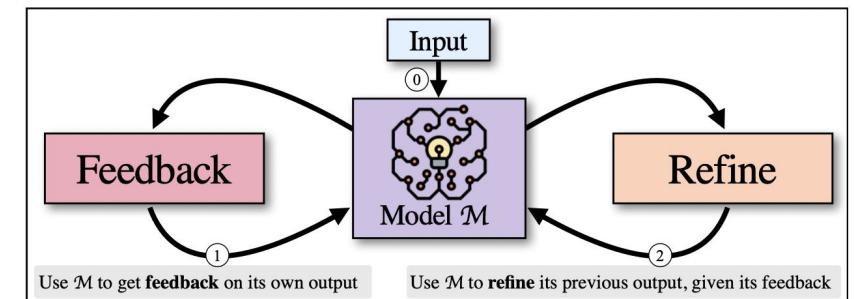
What Was Tried Before

“SELF-REFINE:Iterative Refinement with Self-Feedback”

- **Active correction:** Unlike Self-Consistency, which only aggregates answers, Self-Refine *fixes* errors in reasoning or generation.
- **Efficiency:** Needs fewer iterations than brute-force sampling of many outputs.
- **Learning-like behavior:** Through iterative feedback-refine loops, the model effectively *learns from its own mistakes* within a single query.

Limitations:

- Intrinsic Blindness
- No Long-Term Learning
- High Variance & Prompt Sensitivity
- Single Agent Perspective



Self-Refine → RL / Multi-Agent: To overcome blind self-correction and lack of diversity

What Was Tried Before

“Reflexion: Language Agents with Verbal Reinforcement Learning”

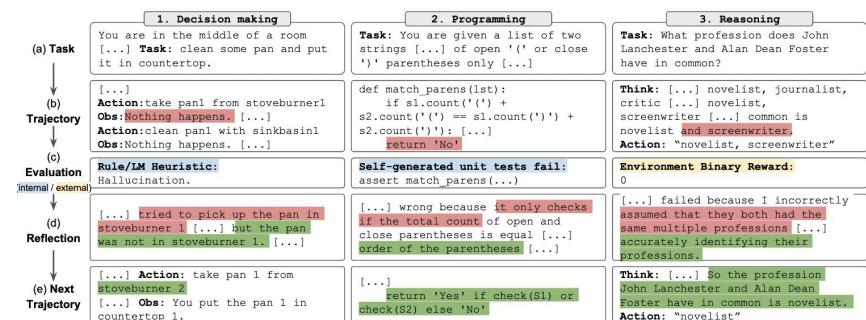
- After acting/answering, the agent critiques its past output and stores experiences in memory.
- Introduces **verbal reinforcement learning**: instead of gradient updates, the model improves via *language feedback loops*.
- **More sample efficient** and **learns from mistakes**

Limitations:

- Overconfidence traps
- Shallow reflections
- Low novelty
- Error propagation

Reflexion showed that reflection can describe mistakes, *but without external signals and memory, it often just repeats them*.

Reflection ≠ true correction.



What Was Tried Before

“Large Language Models Cannot Self-Correct Reasoning Yet”

Core question: *Can LLMs, after producing an answer, reliably self-correct by reflecting and revising?*

Findings:

- Experiments across reasoning tasks show that self-correction **rarely improves accuracy**.
- LLMs often **repeat the same flawed logic** rather than producing new reasoning chains

Shows that **self-correction ≠ true reasoning improvement**.

Demonstrates “**degeneration of thought**”: once a model is overconfident in a wrong path, reflection only reinforces it

Why Multi-Agent Approaches?

- Humans reason better in groups: critique, defend, refine.
- **Key insight:** What one model misses, another may catch.
- Multiple agents debating → diversity of thought, error checking, consensus building.
- Leads to frameworks like **Debate** (Du et al., 2023) and **RECONCILE** (Chen et al., 2024).

Improving Factuality and Reasoning in Language Models through Multiagent Debate Yilun Du et al.

It was the **first step**: showed that agents can argue and reach better reasoning than solo LLMs

Advantages:

- Reduces simple hallucinations
- Forces models to articulate reasoning, improving factuality.

Steps in Multi-Agent Debate

1. Initial Answer Generation
2. Sharing & Critique
3. Debate Rounds
4. Consensus Formation
5. Final Answer Extraction

Illustration of Debate. Illustration of the debate procedure.

User Input: A treasure hunter found a buried treasure chest filled with gems. There were 175 diamonds, 35 fewer rubies than diamonds, and twice the number of emeralds than the rubies. How many of the gems were there in the chest?

Round 1

Agent 1: Let x be the number of rubies. Then the number of diamonds is 175. And the number of emeralds is $2(x-35)$, since there are twice as many emeralds as rubies and there are 35 fewer rubies than diamonds.

To find the total number of gems, we add the number of diamonds, rubies, and emeralds: Total gems = $175 + x + 2(x-35) = 3x + 105$.

We don't know the exact value of x , but we do know that it represents the number of rubies, which is less than 175 (since there are 35 fewer rubies than diamonds). Let's try some values of x : If $x = 50$, then the total gems = $3(50) + 105 = 255$. If $x = 100$, then the total gems = $3(100) + 105 = 405$.

Since we know there were fewer rubies than diamonds, the answer must be closer to 255 than to 405. After some trial and error, we can find that $x = 40$ gives us: Total gems = $3(40) + 105 = 225$.



User Input: Using the solutions from other agents as additional information, can you give an updated response....
(Copy Agent 2's reply) (Copy Agent 1's reply)

Round 2

Agent 1: Given the information provided in other agents' solutions, we have two answers: One agent found that the number of rubies is 40 and the total number of gems is 225. Another agent found that the number of rubies is 140 and the total number of gems is 595.

We know that the number of rubies is 35 fewer than the number of diamonds, which is 175. So the number of rubies should be $175 - 35 = 140$, as the second agent found. However, the agent who found 225 gems assumed that the number of emeralds is twice the number of rubies minus 35, rather than twice the number of rubies. If we use this assumption, we get: Number of rubies = 140. Number of emeralds = $2 * 140 - 35 = 245$. Total number of gems = $175 + 140 + 245 = 560$. Our final answer is 560.



User Input: Using the solutions from other agents as additional information, can you give an updated response....
(Copy Agent 2's reply) (Copy Agent 1's reply)

Round 3

Agent 1: After reviewing the solutions provided by other agents, I agree with the second agent who found that the number of rubies is 140 and the total number of gems in the chest is 595. Therefore, my answer to the math problem is 595.



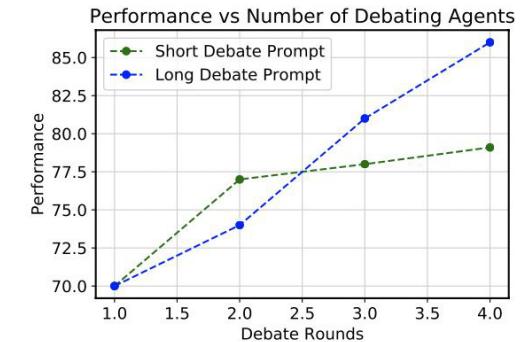
Agent 2: After reviewing the solutions provided by other agents, I also agree that the number of rubies is 140 and the total number of gems in the chest is 595. Therefore, my answer to the math problem is also 595.



Consensus in Debates

- Debate is a **multi-agent game**; convergence is not guaranteed.
- **Empirically**: agents usually reach a single shared answer after several rounds.
- Debate **duration** depends on prompts:
 - **Agreeable prompts** → fast consensus, lower accuracy.
 - **Stubborn prompts** → longer debates, better solutions.
- LLMs tend to be **overly agreeable**, likely due to instruction-tuning/RLHF.

Debate Length	Prompt
Short	<p><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></p>
Long	<p><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></p>



Experiments:

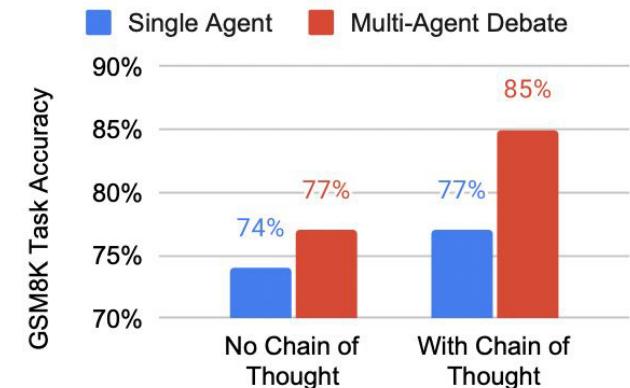
Q1. To what extent does multi-agent debate improve reasoning?

Benchmarks: **Arithmetic**, **GSM8K** (grade school math), **Chess move prediction**.

Result: Debate > single model, reflection, or majority vote.

- Debate can **recover from all-wrong starts** → agents critique each other until they reach the correct answer.
- Debate is **not just amplifying correct answers** — it can *fix mistakes collaboratively*.

Works **orthogonally with other methods**: combining CoT + Debate gives the best GSM8K performance.



Experiments:

Q2. To what extent does debate improve factual validity?

- Reflection-based methods performed poorly on factual QA.
- **Debate achieved the best factuality**, outperforming all baselines.
- Debate responses were more **consistent across agents**.

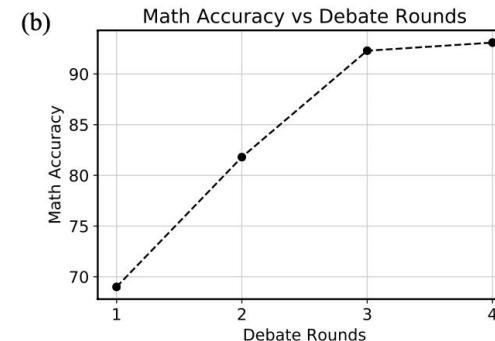
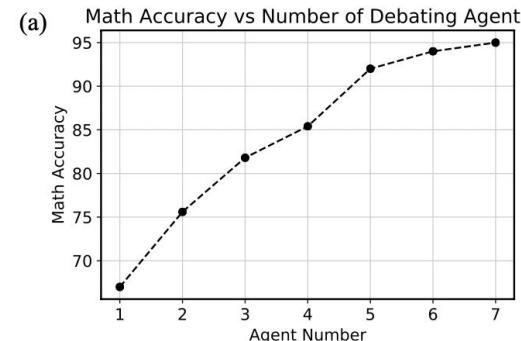
Model	Biographies	MMLU	Chess Move Validity
Single Agent	66.0 ± 2.2	63.9 ± 4.8	29.3 ± 2.6
Single Agent (Reflection)	68.3 ± 2.9	57.7 ± 5.0	38.8 ± 2.9
Multi-Agent (Debate)	73.8 ± 2.3	71.1 ± 4.6	45.2 ± 2.9

Interesting Insight

- Each agent's *self-reported confidence* was still **high**, even when wrong.
- After communication, agents converged to a **more accurate consensus answer**.
- On questions where the model was already confident, agents were **hard to persuade to change**.
- Suggests that "**ease of persuasion**" may reflect factual confidence.

Experiments:

Q3. What design choices enable debate to help?



Takeaway: Debate isn't just about quality of samples; structure and interaction design matter.

Limitations

- **Echo chambers:** Multiple copies of the same model → same biases reinforced.
- **No external validation:** If agents all converge on a wrong answer, there's no safeguard.
- **Confidence blind spot:** All votes treated equally; can't weigh by certainty or quality.
- **Costly:** Multi-round, multi-agent debates scale poorly.
- **Consensus ≠ correctness:** Debate often *sounds* persuasive but isn't always factually right.

These gaps directly motivated adding a *referee/judge* role.

Judge and MAD-Judge

- Introduced an external **judge model** to evaluate the debate and decide on the “winning” reasoning path.
- MAD-Judge: adds **multi-dimensional debates** and judges; for robustness.

Still limited by:

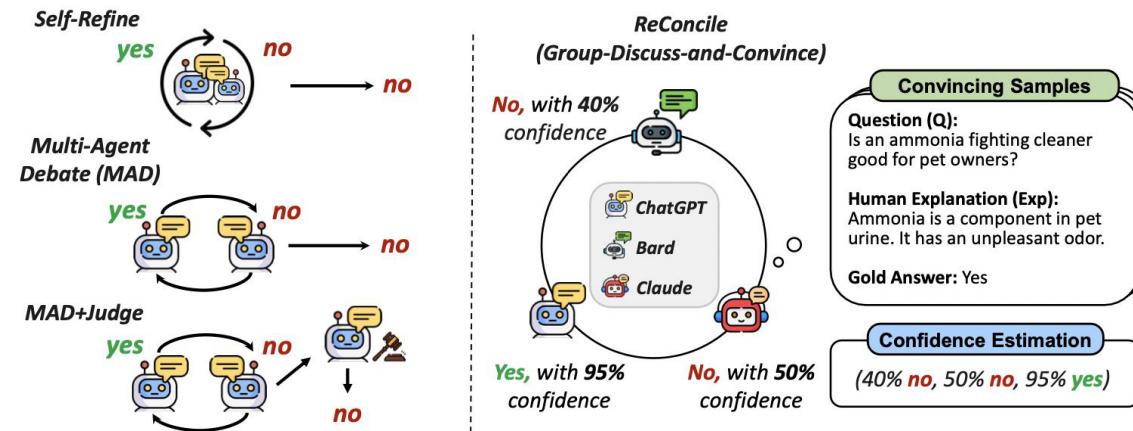
- Agents still copies of the same model (bias not solved).
- Judges themselves inherit model biases.

This pointed to the need for **diversity across models**, not just roles.

RECONCILE: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs Justin et al

Core idea: Round-table conference among diverse *models* (e.g., ChatGPT, Bard, Claude2) with:

1. Initial answers + confidence + explanations.
2. Multi-round discussion where agents use *convincing samples* (human-style explanations) to refine.
3. **Confidence-weighted voting** to form consensus



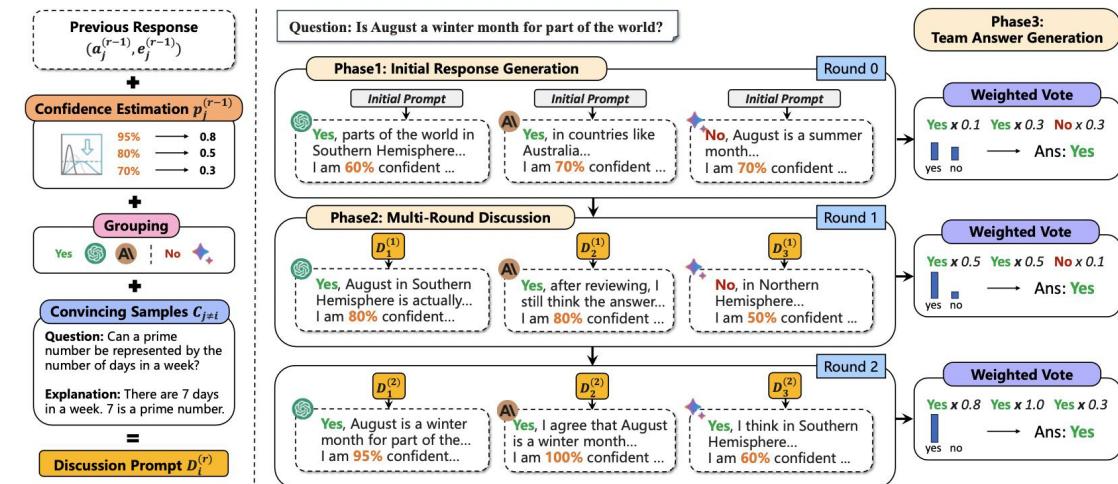
Q. If multiple diverse LLMs collaboratively solve a task, are they capable of discussing their solutions with each other to reach a better consensus?

RECONCILE: A Collaborative Discussion Framework

Phase 1: Initial Response Generation

Each agent (ChatGPT, Bard, Claude2, etc.) generates:

- An **answer**
- A **Chain-of-Thought explanation**
- A **confidence score**

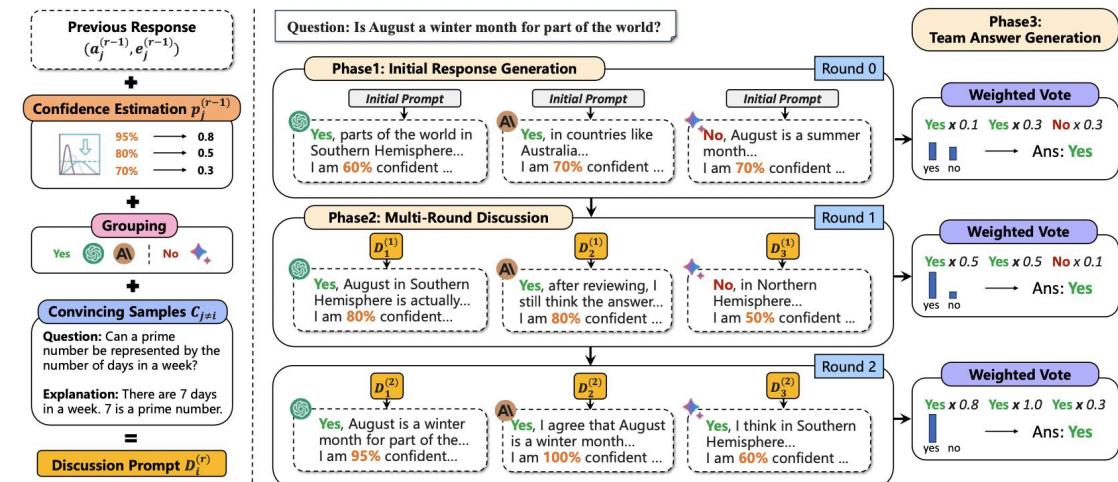


Q. If multiple diverse LLMs collaboratively solve a task, are they capable of discussing their solutions with each other to reach a better consensus?

RECONCILE: A Collaborative Discussion Framework

Phase 2: Multi-Round Discussion

- Agents receive a **discussion prompt** containing:
 - Grouped answers & explanations from all agents (previous round)
 - Confidence scores of peers
 - Convincing samples (human explanations that rectify errors)
- Each agent updates its answer, explanation, and confidence.
- This repeats for several rounds until consensus stabilizes.

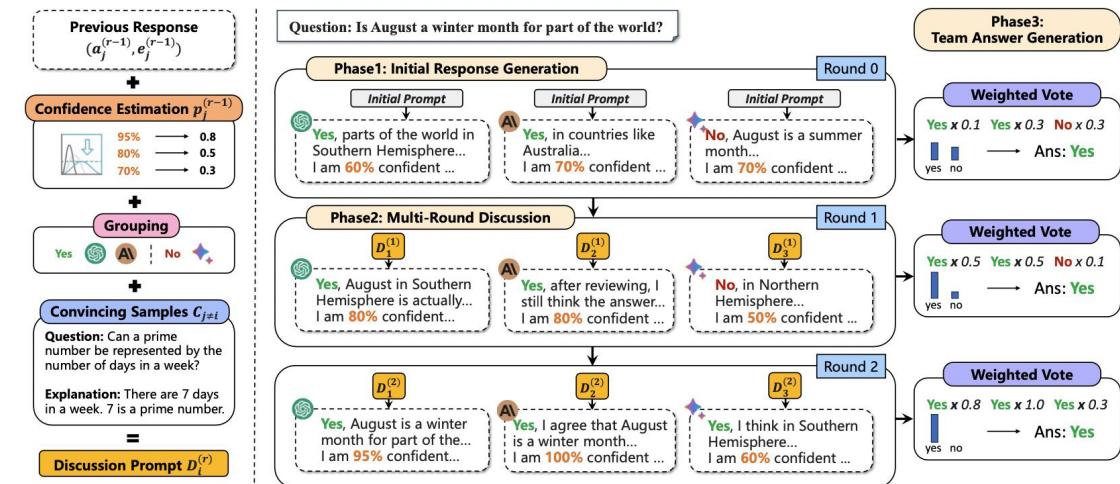


Q. If multiple diverse LLMs collaboratively solve a task, are they capable of discussing their solutions with each other to reach a better consensus?

RECONCILE: A Collaborative Discussion Framework

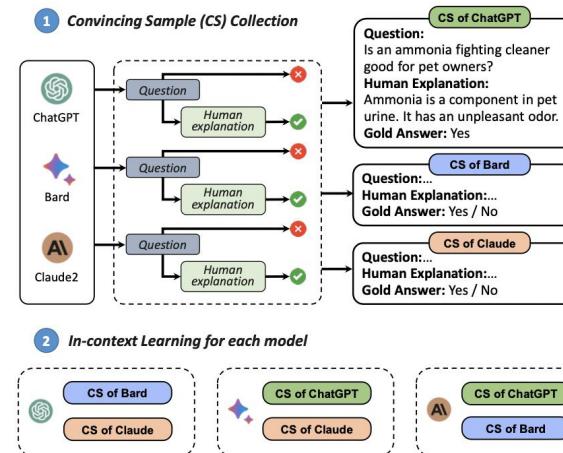
Phase 3: Team Answer Generation

- At the end of each round, a **confidence-weighted vote** aggregates answers.
- Final **team answer** is selected once all agents converge or max rounds reached.



Convincing Samples in RECONCILE

- Convincing samples are *human-provided corrective explanations*.
- They show how to turn an **initially wrong answer** into a **correct one**.
- During discussion, these samples are used in-context to help one agent persuade another.



- Example: ChatGPT gives a wrong answer → Bard and Claude2 use a convincing sample to nudge ChatGPT toward the correct reasoning.
- This mechanism increases **cross-agent learning** and improves **consensus quality**.

Main Results of RECONCILE

1. Outperforms GPT-4 and single-model debate in accuracy on **reasoning** (GSM8K, StrategyQA, ANLI), **factuality** (MMLU, CSQA), **robustness**
2. **Importance of Diversity:** Using *different model families* → boosts accuracy.
3. **Confidence-Weighted Consensus Works:** Voting weighted by self-reported confidence yields more reliable outcomes than simple majority.
4. **Convincingness Matters:** Human-like “convincing samples” help agents correct each other’s mistakes and leads to faster convergence and fewer hallucinations.

Method Category	Method	Agent	StrategyQA	CSQA	GSM8K	AQuA	Date
Vanilla Single-agent	Zero-shot CoT	🕒 GPT-4	75.6±4.7	73.3±0.4	90.7±1.7	65.7±4.6	89.0±2.2
	Zero-shot CoT	🕒 ChatGPT	67.3±3.6	66.0±1.8	73.7±3.1	44.7±0.5	67.7±1.2
	Zero-shot CoT	⭐ Bard	69.3±4.4	56.8±2.7	58.7±2.6	33.7±1.2	50.2±2.2
	Zero-shot CoT	🕒 Claude2	73.7±3.1	66.7±2.1	79.3±3.6	60.3±1.2	78.7±2.1
	Eight-shot CoT	🕒 Claude2	74.3±0.8	68.3±1.7	84.7±0.9	64.7±1.2	78.7±1.7
Advanced Single-agent	Self-Refine (SR)	🕒 ChatGPT	66.7±2.7	68.1±1.8	74.3±2.5	45.3±2.2	66.3±2.1
	Self-Consistency (SC)	🕒 ChatGPT	73.3±0.5	73.0±0.8	82.7±0.5	60.3±1.2	69.3±0.4
	SR + SC	🕒 ChatGPT	72.2±1.9	71.9±2.1	81.3±1.7	58.3±3.7	68.7±1.2
Single-model Multi-agent	Debate	🕒 × 3	66.7±3.1	62.7±1.2	83.0±2.2	65.3±3.1	68.0±1.6
	Debate	⭐ × 3	65.3±2.5	66.3±2.1	56.3±1.2	29.3±4.2	46.0±2.2
	Debate	🕒 × 3	71.3±2.2	68.3±1.7	70.7±4.8	62.7±2.6	75.3±3.3
	Debate+Judge	🕒 × 3	69.7±2.1	63.7±2.5	74.3±2.9	57.3±2.1	67.7±0.5
Multi-model Multi-agent	RECONCILE	🕒, ⭐, 🕒	79.0±1.6	74.7±0.4	85.3±2.2	66.0±0.8	86.7±1.2

Method	Accuracy
RECONCILE	79.0±1.6
w/o Multiple Models	72.2±2.1
w/o Grouping	76.7±2.5
w/o Convincingness	74.5±1.7
w/o Conf Estimation	77.7±1.3

Table 7: Ablations of RECONCILE on StrategyQA.

Metric	Method	Accuracy	D (A1, A2)	D (A1, A3)	D (A2, A3)	D (A1, A2, A3)
BERTScore	RECONCILE (🕒 Paraphrased)	72.2	0.9364	0.9376	0.9453	0.9398
	RECONCILE (🕒 × 3)	72.2	0.9077	0.9181	0.9049	0.9102
	RECONCILE (🕒, ⭐, 🕒)	79.0	0.8891	0.8833	0.8493	0.8739

Limitations of RECONCILE

- High computational cost
- Dependence on model diversity
- Confidence calibration issues
- Convincing samples require human-crafted explanations
- Noisy discussions

What came after RECONCILE?

- **LLM-Blender** → learns to fuse outputs from multiple LLMs instead of debating.
- **Adaptive Multi-Agent Methods** → selectively involve only helpful agents, reducing cost.
- **Mixture-of-Agents + Tools** → combine general LLMs with domain experts, retrieval, or verifiers for stronger reasoning.