

Consensus Is All You Need: Gossip-Based Reasoning Among Large Language Models

Saksham Arora
 Sunnyvale, California
 sakshamarora239@gmail.com
<https://orcid.org/0009-0008-4566-7518>

Abstract—Large language models have advanced rapidly, but no single model excels in every area—each has its strengths and weaknesses. Instead of relying on one model alone, we take inspiration from gossip protocols in distributed systems, where information is exchanged with peers until they all come to an agreement. In this setup, models exchange answers and gradually work toward a shared solution. Each LLM acts as a node in a peer-to-peer network, sharing responses and thought processes to reach a collective decision. Our results show that this “gossip-based consensus” leads to robust, resilient, and accurate multi-agent AI reasoning. It helps overcome the weaknesses of individual models and brings out their collective strengths. This approach is similar to how humans build consensus, making AI seem more collaborative and trustworthy instead of just a black-box program.

Index Terms—large language models, consensus algorithms, gossip protocols, distributed artificial intelligence, collaborative reasoning

I. INTRODUCTION

Large language models like GPT-4 [1], Claude [2], Gemini [3], DeepSeek [4], and Grok [5] have demonstrated impressive capabilities across a wide range of tasks. However, they differ in architecture, training data, biases, and response behavior, leading to inconsistent performance. Inspired by consensus methods in distributed systems, we use gossip protocols as a way for different models to interact and reach agreement. Here, every LLM acts like a peer in the network—producing answers, sharing them with others, and updating its view based on feedback. We try to use different pipelines to gather results—by using the gossip protocol, we can enable multiple different architectures for how these LLMs converge to an answer. The gossip protocol enables nodes to communicate with each other, exchanging local information and updating their views. We can also control how many rounds of communication occur before the system reaches a high-confidence consensus.

There are many ways we can make this algorithm work—by refining answers through prompt engineering and then performing the voting process. Though weighted voting can be applied, it often introduces bias, contradicting our goal of decentralizing inference authority. As such, our approach mirrors modern human consensus-building by avoiding centralized dominance.

For instance, with a large number of LLMs, we can form multiple subsets (e.g., three sets of three models each). Each set first reaches an internal consensus, then leaders from each group propose their answers to a final layer of consensus.

This structure helps optimize for limited context windows, particularly for models with smaller memory capacity.

During consensus, we also pass not just final answers but the thought processes of each model—how they arrived at their decisions. Metadata about previous rounds (e.g., why an answer was selected) can also be shared in multi-round iterations. This metadata allows us to explore whether models develop preferences or implicit biases—whether they tend to favor certain peers more frequently. Findings show that this gossip framework can be really simple and also really complex, according to the application needs.

II. MOTIVATION AND BACKGROUND

A. Motivation

- **Decentralization:** Avoids dependence on a single model, utilizing all available models.
- **Redundancy and fault-tolerance:** Local mistakes can be corrected via peer feedback.
- **Scalability:** Easily accommodates new models without retraining or architecture changes.
- **Emergent agreement:** Consensus emerges from iterative interaction, mimicking social and biological systems [6], [7].
- **Human-like collaboration:** Reflects how humans collectively reason and deliberate.

B. Background

Gossip protocols, modeled after how information spreads in social groups, where each node in the system represents a peer, and repeated rounds of information exchange quickly synchronize the network. This repeated, decentralized sharing gradually aligns the global state without requiring strict synchronization. Such techniques have powered some of the largest and most resilient systems—like DynamoDB and Cassandra—for maintaining eventual consistency [8].

Beyond the technical motivations, it is important to acknowledge the human-facing implications of collaborative AI systems. As these artificial intelligence systems become more integrated into everyday life, it is critical that people do not perceive these systems as opaque or threatening. A consensus-based system inspired by human decision-making processes fosters trust and transparency. If models are trained to behave more like collaborative human agents—deliberating, debating, and

arriving at consensus—the resulting system aligns better with human social intuition.

This approach opens the door to building AI systems that feel less like black boxes and more like thoughtful collaborators. In group decision-making contexts—whether in business, governance, or healthcare—humans rarely rely on a single authority. Instead, ideas are debated, rationales are exchanged, and decisions are shaped collectively. Mimicking this process allows models to become relatable and interpretable, giving users the sense that they are engaging with an ensemble of diverse perspectives rather than a singular, monolithic output.

Such systems can also help correct for individual model bias and present disagreements in a manner that promotes user confidence. Rather than being perceived as inflexible or authoritarian, these models can be seen as co-thinkers: part of a collective that values reflection, dialogue, and shared reasoning.

We extend this principle to AI: rather than propagating data states, our gossip system propagates belief states and thought processes, ultimately reaching consensus not on bytes—but on judgment.

III. ALGORITHM DESIGN

We define three primary variants of the gossip consensus mechanism for multi-agent LLM collaboration.

A. Simple Voting Algorithm (With Context)

Input: A set of models $\{M_1, M_2, M_3\}$ and a question Q .

- Each model generates an answer and corresponding thought process.
- During the consensus round, all models receive the responses from their peers (excluding their own).
- A majority vote determines the final answer.

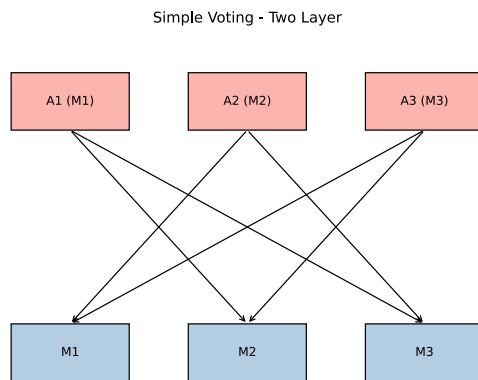


Fig. 1. Simple voting-based gossip consensus flow.

```

1 for model in MODELS:
2     answer[model] =
3         get_answer_and_thought_process(model, Q)

```

```

4     for model in MODELS:
5         final_answer[model] = get_final_answer(model,
6             answers, thoughts)
7
8     counter = {}
9     ANSWER = None
10    for model in MODELS:
11        counter[final_answer[model]] += 1
12        if counter[final_answer[model]] > count:
13            ANSWER = final_answer[model]

```

B. Voting Algorithm with a Judge

- One model is randomly chosen to act as the judge.
- All others submit their answers and thought processes to the judge.
- The judge selects the final answer from peer submissions.
- Judges are rotated across rounds to mitigate long-term bias.

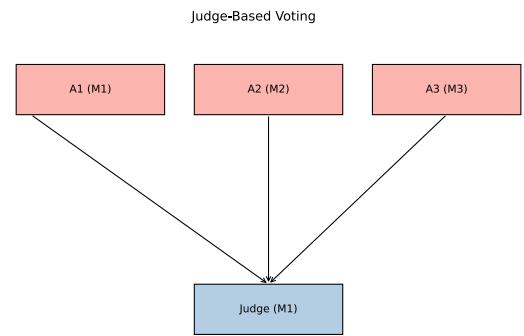


Fig. 2. Consensus with a rotating judge.

```

1 JUDGE = random.choice(MODELS)
2
3 for model in MODELS:
4     if model != JUDGE:
5         answer[model] =
6             get_answer_and_thought_process(model, Q)
7
8 final_answer = get_answer(JUDGE, answers,
9 thoughts)

```

C. Multi-layer Consensus (Hierarchical)

- Models are grouped into sets that independently reach consensus.
- Each group selects a leader whose answer represents the group.
- Leaders then run a second-layer consensus round.
- Suitable for scaling to large LLM clusters and optimizing for context window limitations.

```

1 GROUPS = { [m1, m2, m3], [m4, m5, m6] }
2 for group in GROUPS:
3     for model in group:
4         answer[group][model] =
5             get_answer_and_thought_process(model, Q)
6     leader = elect_leader(group, answer[group])

```

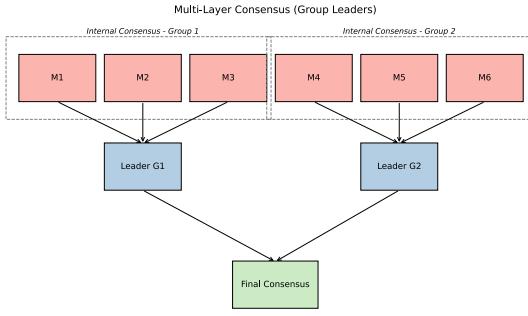


Fig. 3. Multi-layer hierarchical consensus.

```

6     leaders.append(leader)
7
8     ans = []
9     for leader in leaders:
10        ans[leader] = get_answer_and_thought_process(
11            leader, Q)
12
ANSWER = max(ans)

```

IV. FINDINGS AND OBSERVATIONS

We tested our gossip-based consensus idea on two sets of models: one with stronger, newer models and one with low-end, lighter models. The goal was to see if letting them “talk” and then vote would beat their individual performance. All evaluations were run on a sample of MMLU benchmark [9] dataset.

A. High-End Models

The first group included o4-mini, gemini-2.5-pro, grok-4-0709, and deepseek-reasoner. On their own, these models scored between 82.6% and 89.4%. The best was gemini-2.5-pro at 89.4%. When combined through majority voting, the accuracy jumped to 93.3%.

That is a relative improvement of about **+4.3 percentage points** over the best single model and **+10.7% fewer errors**. Even when the models are already strong, consensus consistently squeezes out a little more performance.

B. Low-End Models

The second group had o3-mini, gemini-1.5-flash, grok-3, and deepseek-chat. These were weaker overall, scoring between 62.2% and 77.3%. The best single model was grok-3 at 77.3%. But when we ran consensus, accuracy went up to 84.2%.

That means a relative improvement of about **+6.9 percentage points** over the best single model and **+30.4% fewer errors**. The effect here is much bigger compared to the stronger models, showing that consensus provides the largest benefit when the models are less reliable.

Importantly, the total cost of running this entire low-end experiment was roughly **half the price of a single run**.

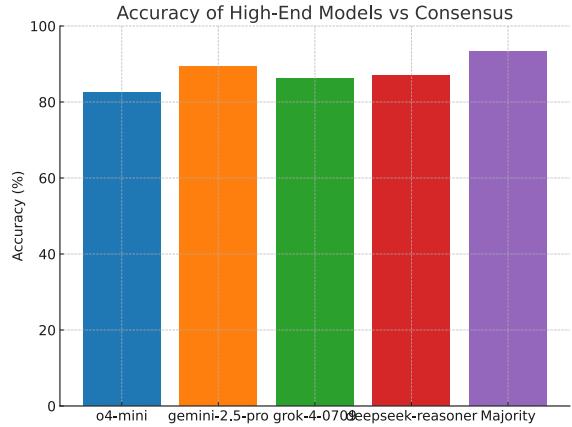


Fig. 4. High-end models compared to their consensus.

with only gemini-2.5-pro. This highlights a key point: consensus not only improves accuracy, it can do so while being *cost-efficient*. With the right mix of models, we can achieve near frontier-level performance without always paying frontier-level prices.

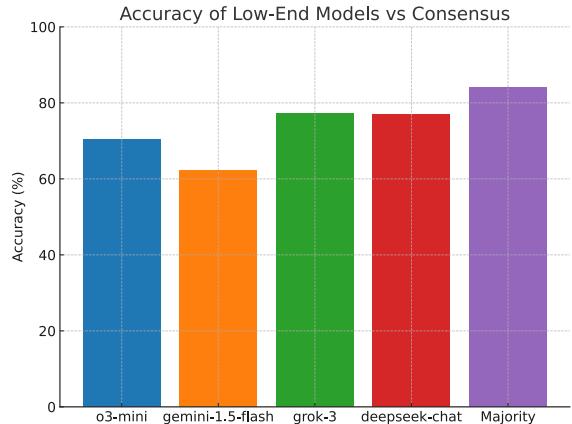


Fig. 5. Low-end models compared to their consensus.

C. Key Takeaways

- Consensus always beats the single best model in the group.
- The boost is modest for high-end models (+4.3 points, 10.7% fewer errors), but large for low-end ones (+6.9 points, 30.4% fewer errors).
- Consensus can be more cost-effective: the low-end set cost **50% less** than a single gemini-2.5-pro run.
- Using a diversity of models is often better than relying on just one; each model brings different strengths.
- An automatic “model chooser” system could make this practical, routing questions to the most cost-efficient mix of models for consensus.

- Having multiple models balance each other also reduces mistakes and biases.
- The way the models agree feels closer to how humans reason together in groups.

In short, when models “gossip” and agree, they perform better than when they work alone. For weaker models, this makes a significant difference, and for stronger ones, it still adds reliability but also considerable cost and latency. Therefore, a smart way would be to combine diverse small models, which would increase accuracy and also decrease the overall cost, and hence, an affordable solution.

D. Future Work

While majority voting already shows strong improvements, there are several promising directions to extend this framework:

- **Confidence-based consensus:** Weighted voting, where each model has given more influence, may seem attractive. However, it risks introducing long-term bias if a single model repeatedly dominates. A better approach is to make weights *dynamic*, rewarding models when their answers align with ground truth and penalizing them when they are wrong. Over time, this would act like a training-like signal, producing a confidence-based consensus that balances fairness with accuracy.
- **Automatic model chooser:** Different models excel in different domains. For example, one model may be stronger in reasoning while another is better at factual recall. An automatic router could classify the incoming query and direct it to the most suitable set of models for consensus. This would maximize both accuracy and cost-efficiency by avoiding unnecessary use of frontier models.
- **Scaling to larger ensembles:** Our current experiments used four-model groups, but the gossip protocol naturally supports larger clusters. Future work could explore hierarchical or multi-layer gossip across dozens of heterogeneous models, allowing consensus to form at scale without exceeding context limits.
- **Human-AI hybrid consensus:** Since our framework mimics group deliberation, an intriguing direction is to include human reasoning as another “node” in the gossip. If ever the consensus is not found, a human can intervene and takeover the consensus. This could become critical for decision in fields such as healthcare, law, or governance, where human oversight is critical.

In short, consensus is not just a mechanism for combining models, but a foundation for building **adaptive, cost-aware, and trustworthy** multi-agent reasoning systems.

V. REFERENCES

REFERENCES

- [1] OpenAI, “GPT-4 Technical Report,” Mar. 2023. [Online]. Available: <https://openai.com/research/gpt-4>
- [2] Anthropic, “Claude 2 Release,” Jul. 2023. [Online]. Available: <https://www.anthropic.com/index/clause>
- [3] Google DeepMind, “Gemini 1.5 Technical Report,” Feb. 2024. [Online]. Available: <https://deepmind.google/technologies/gemini/>
- [4] DeepSeek, “DeepSeek LLM Release,” 2024. [Online]. Available: <https://deeplearn.com/>
- [5] xAI, “Grok Model Overview,” 2024. [Online]. Available: <https://x.ai/>
- [6] D. Kempe, J. Kleinberg, and É. Tardos, “Gossip-based computation of aggregate information,” in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, 2003, pp. 482–491.
- [7] R. van Renesse, Y. Minsky, and M. Hayden, “A gossip-style failure detection service,” in *Proc. IFIP Int. Conf. on Distributed Systems Platforms*, 1998, pp. 55–70.
- [8] A. Lakshman and P. Malik, “Cassandra: A decentralized structured storage system,” *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 2, pp. 35–40, 2010.
- [9] D. Hendrycks, C. Burns, S. Basart, et al., “Measuring Massive Multitask Language Understanding,” in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021. [Online]. Available: <https://huggingface.co/datasets/cais/mmlu>