Consensus-Driven Metacognition in Multi-Agent Systems: A Logic-Based Byzantine Fault-Tolerant Protocol

Saurav Bhattacharya

The New World Foundation
Email: president@thenewworldfoundation.com

***Abstract*—Large language model (LLM) agents fail in a distinctive way: they produce *confident wrong answers*. Recent work has begun adapting Byzantine fault tolerance (BFT) to multi-LLM networks — notably the weighted-BFT protocol of Wang *et al.* [1] and the Aegean consensus engine [2] — typically by treating agent trust as a scalar that flows continuously through the protocol. This paper takes the opposite stance: discretisation is a feature, not a bug. MBFT (Metacognitive BFT) commits a swarm decision via a small, finite ladder of confidence tiers, defeasible counter-proofs, and a reputation-gated veto. The resulting protocol is auditable, deterministically replayable, and its safety / liveness claims are mechanically checkable — properties that continuous Bayesian aggregators struggle to provide. Bayesian and continuous-trust schemes remain the right tool for noisy, well-calibrated, high-volume regimes (recommender systems, sensor fusion, market making); we argue that high-stakes reasoning swarms — legal, medical, safety-critical agentic deployments — belong to the discrete, defeasible regime that MBFT formalises. We accompany the paper with an open-source reference implementation whose property suite encodes each theorem as an executable test.**

***Index Terms*—multi-agent systems, large language models, Byzantine fault tolerance, metacognition, defeasible reasoning, epistemic logic, consensus protocols, AI safety, auditability, hallucination mitigation, weighted voting, reputation systems**

I. Introduction

LLM-based agents fail in a distinctive way: they produce *confident wrong answers*. Classical crash-fault-tolerant (CFT) consensus protocols such as Paxos [3] and Raft [4] assume that any node which responds is truthful, an assumption that does not survive contact with hallucinating agents. We argue that hallucination is a *Byzantine* fault and that the appropriate response is a consensus layer whose semantics are tied to agents' *metacognitive* state rather than to bit-level log agreement.

Design stance: discretisation is the thesis

The recent weighted-BFT-for-LLMs line [1], [2] treats agent trust as a continuous scalar threaded through the protocol. MBFT deliberately departs from that choice. Each agent's metacognitive state is reported on a small, finite ladder of tiers (e.g. $\{\text{Low}, \text{Medium}, \text{High}\}$ or a bucketised $[0, 1]$); rejection requires a constructed counter-proof, not a noisy scalar dip; commit is defined by a discrete predicate over those tiers and over a reputation-slashing rule.

This is not a concession to engineering convenience — it is the central design choice the paper defends. Discretisation buys five properties that high-stakes reasoning deployments need and that continuous Bayesian aggregators struggle to deliver:

1) *Auditability.* A commit can be replayed and inspected as a finite proof tree rather than a posterior trajectory.
2) *Determinism.* Identical agent reports always yield identical commits; no Monte-Carlo variance.
3) *Mechanical verifiability.* Safety and liveness become finite case analyses, encodable as property tests (Sec. IV, V).
4) *Adversarial robustness without calibration assumptions.* Continuous schemes silently amplify miscalibrated overconfidence; a defeasible counter-proof either exists or it does not.
5) *Regulatory legibility.* A discrete commit rule with a named reputation history maps cleanly onto post-hoc audit regimes (medical, legal, financial, safety-critical).

When the Bayesian regime is the right one

We do not claim discretisation dominates universally. Continuous, Bayesian, or stake-weighted aggregators remain the appropriate tool in domains where (i) reports ar-

rive at high volume, (ii) calibration is empirically known and stable, and (iii) decisions are individually low-stakes and amortised over many trials. Recommender ensembles, sensor fusion, online ad allocation, and market-making swarms sit naturally in that regime. We sketch a Bayesian variant of MBFT in Sec. VI for exactly those settings, and ship its reference implementation alongside the discrete protocol. The two are complementary, not competing.

Contributions and scope

We position this work as a *refinement and rephrasing* of the recent weighted-BFT-for-LLMs line of research, with discretisation as the organising principle. Our contributions are:

- A discrete, tier-based formulation of metacognitive BFT designed for auditable, high-stakes reasoning swarms (Sec. III).
- A defeasible-logic verification primitive that requires a constructed counter-proof for rejection, in contrast to scalar trust scores [1] or quality signals [2].
- A reputation-gated veto: only un-slashed agents can block commit, so previously-faulty agents retain voice but not unilateral blocking power.
- Safety, liveness, and slashing-monotonicity propositions with sketch proofs (Sec. IV), each encoded as a parameterised property test in the companion repository (Sec. V).
- A complementary Bayesian generalisation (Sec. VI) intended for the high-volume, well-calibrated, low-stakes regime — not as a successor to the discrete protocol but as its sibling.

What we do *not* claim: novelty of weighted BFT for LLM swarms (established by [1], [2]); novelty of multi-agent debate as a hallucination mitigation [5], [6]; or novelty of confidence-weighted aggregation broadly [7].

II. Background and Related Work

A. Multi-agent debate and self-consistency

Multi-agent debate frameworks [5], [6] and self-consistency decoding [7] aggregate multiple LLM samples to reduce hallucination, typically via majority vote or judge-model adjudication. These works do not model Byzantine faults explicitly and treat all agents as honest-but-noisy.

B. Byzantine consensus for LLM swarms

The closest prior work to ours is the Weighted BFT protocol of Wang *et al.* [1], which adapts PBFT-style consensus to multi-LLM networks with *dynamically adjusted voting weights* keyed to historical response quality. Aegean [2] ships a production BFT engine for multi-agent LLM systems with adaptive quorum detection and AutoGen integration. Both treat agent trust as a scalar; both use vote-counting (weighted or unweighted) as the verification primitive.

C. Classical consensus

Paxos [3] and Raft [4] are crash-fault-tolerant and assume any responding node is truthful, an assumption violated by hallucinating LLMs. PBFT [8] and Hot-Stuff [9] provide Byzantine guarantees but operate over bit-level state rather than over reasoning artefacts.

D. Where MBFT sits

MBFT is best understood as a logic-flavoured refinement of [1]. The novel mechanisms are (i) a defeasible-proof verification step in place of a scalar quality signal, (ii) a reputation-gated veto rule, and (iii) a mechanically checked property suite shipped with the paper. We do not benchmark against [1], [2]; a head-to-head empirical comparison is the obvious next step and is left as future work.

III. The MBFT Protocol

A. Ontology

Definition 1 (Agent set)**.** *Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a finite set of agents. A non-empty subset $\mathcal{B} \subset \mathcal{A}$ may be Byzantine; let $f = |\mathcal{B}|$.*

Definition 2 (Metacognitive state)**.** *For task T and proposed solution S_i, agent a_i produces a* metacognitive weight $\tau_i(S_i) \in [0, 1]$ *and a proof trace P_i.*

Definition 3 (Reputation)**.** *Each agent carries a reputation $\rho_i \in (0, 1]$, initialised to 1 and reduced by a multiplicative* slash factor $\sigma \in (0, 1)$ *whenever it leads a failed round.*

B. State transitions

Epistemic Leader Election.:

$$L_r = \arg \max_{i \in \{1,\ldots,n\}} \rho_i \cdot \tau_i(S_i). \tag{1}$$

Semantic Verification.: Each follower a_i executes a verifier $\mathrm{Verify}_i(P_L)$ and emits a vote

$$V_i(S_L) = \begin{cases} \rho_i \cdot \tau_i(S_L) & \text{if } \mathrm{Verify}_i(P_L) = \top \\ -\rho_i \cdot \tau_i(\neg S_L) & \text{if counter-proof produced} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Algorithm 1: MBFT round

each a_i produces (S_i, P_i, τ_i) in parallel
elect L_r via Eq. (1)
each follower computes V_i via Eq. (2)
if Eq. (3) holds **then**
    return S_{L_r}
else
    slash leader; promote counter-proposer; recurse
end if

Confidence-Weighted Finality.:

$$\text{Commit}(S_L) \iff \Big(\sum_{i=1}^{n} V_i(S_L) \geq \theta_{\text{meta}}\Big) \wedge \big(\nexists\, i\colon V_i < 0 \wedge \rho_i = 1\big). \tag{3}$$

View Change.: On commit failure, $\rho_{L_r} := \sigma \cdot \rho_{L_r}$ and the strongest reputation-weighted counter-proposer leads round $r + 1$.

C. Algorithm

IV. Theoretical Guarantees

Proposition 1 (Safety under $f < n/3$)**.** *Let $f < n/3$ and let θ_{meta} satisfy $\theta_{meta} > f \cdot \tau_{\max}$. Then* MBFT *never commits a proposal originating from $\mathcal{B}$.*

Proof sketch. A Byzantine leader cannot accumulate θ_{meta} without recruiting at least one honest vote, but honest verifiers reject any unsound proof and emit a counter-proof, triggering the veto clause of Eq. (3). □

Proposition 2 (Liveness under unanimity)**.** *If all n agents are honest and propose the same S with mean confidence $\bar{\tau}$, and $\theta_{meta} \leq n\bar{\tau}$, then* MBFT *commits in round* 0.

Proposition 3 (Slashing monotonicity)**.** *For every agent a_i and every round r, $\rho_i^{(r+1)} \leq \rho_i^{(r)}$.*

Proposition 4 (Wisdom-of-the-swarm)**.** *Holding $\bar{\tau}$ fixed, the committed aggregate weight is non-decreasing in n.*

Each proposition is encoded as a parameterised property test in `tests/test_propositions.py`; see Sec. V.

V. Reference Implementation

The companion repository[1] provides an `asyncio` simulator that implements Algorithm III-C verbatim. State is modelled with strict `pydantic` schemas mirroring the formalism. The `MockAgent` class supports Byzantine injection via a single flag, enabling the property tests of Sec. IV to be re-run mechanically:

```
$ pytest -q
........................... 23 passed
```

[1]https://github.com/sauravbhattacharya001/metacognition

VI. Discussion and Future Work

The Bayesian sibling, and where it belongs.: The discretisation defended in Sec. I is the right default for high-stakes reasoning swarms, but it is not the only useful regime. For domains with high report volume, empirically stable calibration, and low individual stakes — recommender ensembles, sensor fusion, online bidding, market making — a continuous Bayesian aggregator is more natural and more sample efficient. We sketch such a variant here. Each report (S_i, τ_i) becomes a likelihood signal:

$$P(\text{report}_i \mid h) = \begin{cases} \tau_i & \text{if } S_i = h \\ (1 - \tau_i)/(k - 1) & \text{otherwise,} \end{cases}$$

with reputation acting as a tempering exponent. Commit requires both a posterior-mass threshold and a Bayes-factor margin. A reference implementation lives in `src/core/protocol_bayesian.py`. We emphasise that this variant is a *sibling*, not a successor: the two protocols target different regimes and the discrete MBFT commit rule is what we defend for reasoning-critical deployments.

Open problems.: (i) Calibration of τ from raw token log-probabilities — acute for the Bayesian sibling, less so for the discrete tiered form; (ii) modelling correlated agent errors that violate the independence assumption of the Bayesian variant; (iii) pipelined view changes, in the spirit of HotStuff, to amortise the cost of leader rotation; (iv) a head-to-head empirical comparison against WBFT [1] and Aegean [2] on a reasoning-heavy benchmark.

VII. Conclusion

MBFT stakes out a deliberately discrete, tier-based corner of the weighted-BFT-for-LLMs design space [1], [2]. The wager is that auditability, determinism, and mechanical verifiability matter more than continuous calibration in the domains where consensus on machine reasoning will actually be deployed under scrutiny — legal, medical, financial, safety-critical. Where those constraints relax, the Bayesian sibling sketched in Sec. VI is the right tool. The two protocols partition the space; neither subsumes the other.

REFERENCES

[1] e. a. Wang, "A weighted Byzantine fault tolerance consensus driven trusted multiple large language model network," 2025, arXiv:2505.05103. [Online]. Available: https://arxiv.org/abs/2505.05103

[2] Advaita Labs, "Aegean consensus: A Byzantine fault-tolerant consensus protocol for multi-agent LLM systems," 2025. [Online]. Available: https://github.com/AdvaitaLabs/aegean-consensus

[3] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[4] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *USENIX Annual Technical Conference*, 2014.

[5] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *ICML*, 2024, arXiv:2305.14325.

[6] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi, "Encouraging divergent thinking in large language models through multi-agent debate," in *EMNLP*, 2024, arXiv:2305.19118.

[7] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *ICLR*, 2023.

[8] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI*, 1999.

[9] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus with linearity and responsiveness," in *PODC*, 2019.