A2A-BFT: BYZANTINE FAULT-TOLERANT CONSENSUS FOR AGENT-TO-AGENT PROTOCOLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent-to-Agent (A2A) protocols let autonomous agents collaborate on complex tasks, but reliable consensus in heterogeneous agent networks remains a fundamental challenge once Byzantine failures and soft faults are admitted. We propose **A2A-BFT**, a Byzantine fault-tolerant consensus extension for A2A protocols: a three-phase semantic consensus protocol (Propose–Validate–Commit) with fault-adaptive decision thresholds and a reputation tracker excluded by construction from the vote-counting rule. We prove safety and liveness under $n \geq 3f + s + 1$ (f Byzantine, s soft faults, $s \leq f$) and evaluate entirely on real LLM inference: a 7,750-task fault sweep (n=250 per cell) with four heterogeneous open-source models, spanning mathematical reasoning, code generation, and knowledge QA. On execution-validated code tasks the protocol committed no wrong answer in any within-boundary cell ($0.0 \pm 0.0\%$ under every attack; per-cell 95% Wilson upper bound 1.5%), while sub-boundary settings violate safety up to 72.4% on GSM8K and 90.0% on knowledge QA, consistent with $3f{+}s{+}1$ being a tight operating condition for $s \leq f$; on semantically validated tasks, wrong commits within the boundary are bounded by the validation oracle's error rate (11.6–23.2% on GSM8K, 35.6–60.8% on MMLU, with 8B judges). Under collusion, unvalidated majority-voting and debate baselines commit incorrect answers in 16.7–63.4% of tasks *within* the boundary. Ablations identify view change as the core liveness component—removing it collapses decisions from 74.4% to 4.5% under strategic rejection (three-seed ablation)—and fault-adaptive thresholds as the safeguard against attack-induced deadlock. A measured audit shows the released reputation tracker to be decision-neutral but mis-specified as a deterrent: in the four attacked cells, $1{,}186$ of $1{,}305$ aggressive penalties landed on honest validators whose vote was objectively correct.

Keywords: Byzantine fault tolerance, agent consensus, large language models, distributed systems, reputation mechanism

1 INTRODUCTION

Large language models (LLMs) enable autonomous AI agents to perform complex reasoning, code generation, and collaborative problem-solving. The Agent-to-Agent (A2A) protocol, introduced by Google at I/O 2025 and recently donated to the Linux Foundation, provides a standardized framework for inter-agent communication, with over 150 enterprise supporters including Amazon, Anthropic, and Salesforce (Linux Foundation, 2025). Despite these advances, ensuring reliable consensus in multi-agent systems remains an open challenge.

Motivating Example: five agents must agree on a code solution while one is compromised (Byzantine fault) or intermittently erroneous (soft fault). Traditional BFT protocols assume deterministic, byte-verifiable replicas; LLM-based agents are *probabilistic* replicas with no deterministic verifier, so agreement becomes a semantic problem outside classical BFT theory, requiring new mechanisms.

Recent work by ETH Zurich (Berdoz et al., 2026) found that LLM-based agents reach valid consensus in only 41.6% of trials *even in benign settings without any Byzantine agents*, with failures dominated by liveness loss rather than value corruption. This gap motivates our work: how can a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

consensus protocol achieve reliability in heterogeneous agent networks whose participants may be Byzantine, soft-faulted, or heterogeneous in capability?

Our Approach: We propose A2A-BFT, a Byzantine fault-tolerant consensus extension with three innovations: (i) a three-phase semantic consensus (Propose-Validate-Commit) tailored to LLM-based reasoning tasks; (ii) fault-adaptive decision thresholds that prevent attack-induced deadlock (Section 6); and (iii) formal safety and liveness guarantees under partial synchrony with the fault model $n \geq 3f + s + 1$ (with the standing assumption $s \leq f$). We validate the protocol entirely on real four-model LLM deployments (7,750-task fault sweep, n=250 per cell), showing that (a) execution-validated consensus is mechanically safe within the boundary and violations below it match the theorem's prediction, and (b) on semantically validated tasks, wrong commits under attack are bounded by the validation oracle's error rate (Corollary 5.2). We additionally isolate the reputation tracker by direct measurement: replaying instrumented vote streams shows it decision-neutral, as the design requires, but, as released, it directs $1{,}186$ of $1{,}305$ aggressive penalties in the attacked cells at honest validators whose votes were objectively correct.

2 Related Work

Byzantine fault tolerance in distributed systems. Classical BFT protocols such as PBFT (Castro & Liskov, 1999) assume deterministic, faithfully executing replicas with bounded, known-size faults; Byzantine-robust aggregation in distributed learning (Blanchard et al., 2017) tolerates adversarial gradient contributions but targets numeric aggregation. A2A-BFT instead targets *probabilistic* replicas whose correctness depends on sampled LLM outputs, requiring semantic rather than byte-level validation and a unanimity-style agreement probability $p_h^{n-1-f-s}$ (Section 5). The $+s$ term is not a re-badged omission count—an omission fault goes silent and is discountable, whereas a soft fault keeps voting and may err stochastically; Section 5 encodes this via coefficient 1 on s (vs. 3 on f), worst-case abstention in the safety proof, and the standing assumption $s \leq f$.

Multi-agent consensus with LLMs. Debate- and voting-based aggregation improves answer quality but offers no guarantee against committed wrong answers under Byzantine agents: our experiments show 16.7–63.4% wrong-commit rates for such baselines. Self-aggregation in the A2A-Sim style stops by supermajority vote and has no validation stage. Verifier-guided generation and reranking methods (Cobbe et al., 2021; Lightman et al., 2024) likewise use verification to *rank or filter* candidates; A2A-BFT instead uses it to *gate* commitment. No prior LLM-consensus protocol, to our knowledge, states a wrong-commit bound in terms of the validation oracle's error rate; A2A-BFT makes that dependence explicit (Corollary 5.2) and measures it. Concurrent robust-aggregation designs offer neither property: confidence-probe weighting (Zheng et al., 2026) keeps unvalidated weighted aggregation, and self-anchored filter-and-refine consensus (Lee et al., 2026) commits the majority-filtered answer without a validation oracle. The gap is not hypothetical: even an oracle-assisted debate baseline—keeping any solution that passes all tests, under the same round budget—still commits wrong answers on 46.7–56.7% of MBPP tasks under attack (Appendix A.3). It is validation *inside* the commit rule, not robustness of the aggregation weights, that binds wrong commits to the oracle's error rate. Interoperability standards (MCP, ANP) are orthogonal transport fabrics beneath this consensus layer (extended discussion: Appendix B.3).

3 System Model

3.1 Agent Network Model

We consider a network of n heterogeneous agents $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ operating under the A2A protocol framework. Each agent A_i has a unique decentralized identifier (DID) for authentication, a capability score g_i reflecting its domain expertise, a reputation score $r_i \in [0.1, 1.0]$ updated dynamically, and a participation weight $w_i \in [0.1, 2.0]$ used only when the deployment subsamples validators (not in the vote-counting rule; Section 4.4).

3.2 Fault Model

We extend the classical BFT fault model to account for both Byzantine and soft faults:

Definition 3.1 (Fault Model)**.** *A system with n agents is characterized by parameters (n, f, s) where:*

- *f: maximum number of Byzantine (malicious) agents, which may vote arbitrarily, including equivocation, coordination, or selective rejection.*
- *s: maximum number of soft-fault agents (Definition 3.2); $s \leq f$ is assumed when soft faults vote REJECT rather than abstaining.*
- *Safety condition: $n \geq 3f + s + 1$.*

Definition 3.2 (Soft fault)**.** *A soft fault is an* intermittent, non-malicious *failure: the agent remains live and votes in every round but errs stochastically—it may ABSTAIN or vote REJECT at random, never coordinates with other faulty agents, and* never ACCEPTs an invalid proposal *(Appendix A.1.2 quantifies the relaxation).*

The coefficient 1 on s (vs. 3 on f) reflects the reduced threat: soft faults abstain or err randomly and never coordinate, so they do not actively undermine consensus. This extends the classical $n \geq 3f+1$ bound, recovering it when $s = 0$.

3.3 Task Model

We consider a task T drawn from a distribution $\mathcal{D}$ over a task space $\mathcal{T}$, with a verifiable solution space $\mathcal{S}$ and a ground-truth verification function $V : \mathcal{T} \times \mathcal{S} \to \{0, 1\}$. Tasks include mathematical reasoning, code generation, knowledge-based QA, and logic puzzles.

4 Protocol Design

A2A-BFT operates in rounds, each consisting of three phases as illustrated in Figure 1: **Phase 1: Propose** (the designated primary generates a task solution), **Phase 2: Validate** (other agents independently verify the proposal), and **Phase 3: Commit** (agents vote on the verification result; the count score ϕ is compared against fault-adaptive thresholds).

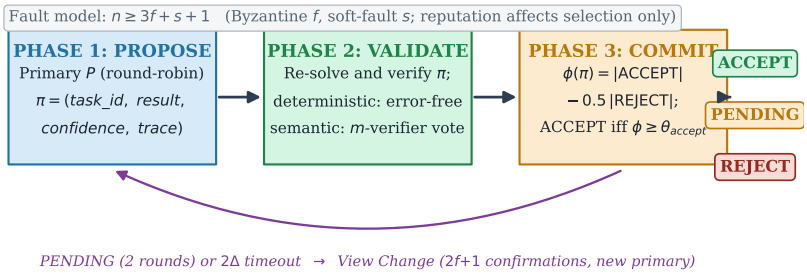

Figure 1: A2A-BFT three-phase consensus protocol. The round-robin primary broadcasts a proposal π (Phase 1); validators re-solve and verify it—error-free checks for deterministic tasks and m-verifier semantic checks otherwise (Phase 2); votes are aggregated by the count score $\phi(\pi)$ under fault-adaptive thresholds (Phase 3). Rejections trigger view change immediately, and PENDING outcomes do so after two consecutive rounds or a 2Δ timeout.

4.1 Phase 1: Propose

The primary agent P receives task T and produces proposal $\pi = (task_id, result, confidence, trace)$, where $task_id$ uniquely identifies the task, $result$ is the proposed solution, $confidence \in [0, 1]$ is a self-assessed confidence, and $trace$ is the execution trace for auditability. The proposal is broadcast to all validators with timestamp τ; the primary is selected via round-robin rotation to ensure fairness.

4.2 PHASE 2: VALIDATE

Each validator V_i independently verifies proposal π:

$$\hat{V}_i(\pi) = \begin{cases} \text{ACCEPT} & \text{if verification succeeds} \\ \text{REJECT} & \text{if verification fails} \\ \text{ABSTAIN} & \text{if verification inconclusive} \end{cases} \tag{1}$$

Verification re-computes the solution and compares results. Formally, writing $\hat{V}_i$ for validator i's verdict on a proposal (the ground-truth oracle V is defined in Section 3.3), let $\mathrm{ver}(T, s) \in \{0, 1\}$ denote whether verification of solution s for task T succeeds: $\mathrm{ver}(T, s) = 1$ iff s matches the ground truth (or an equivalent formulation) in the *execution-validated* domain, where the check is error-free; in the *semantically validated* domain (free-form reasoning and multiple-choice QA), $\mathrm{ver}(T, s) = 1$ iff at least k of m verifiers agree (Zheng et al., 2023), with $k = \lceil 2m/3 \rceil$; the verifier error then falls to the binomial tail $\sum_{j=0}^{k-1} \binom{m}{j} p_h^j (1-p_h)^{m-j}$ rather than $(1-p_h)^m$, where p_h is the probability that an honest validator correctly accepts a valid proposal. Symmetrically, p_{mr} denotes the oracle's *miss* probability—accepting or abstaining on a wrong proposal; safety (Theorem 5.1) is governed by this second error direction. In our experiments code generation (MBPP) is execution-validated while mathematics (GSM8K) and knowledge QA (MMLU) are semantically validated, each validator using a *single* self-model judge (m=1), so redundancy comes *across* validators via Phase 3; a mild self-preference bias is possible when a validator shares the primary's model (Zheng et al., 2023); with four models across five replicas such pairings occur regularly (a residual bias source). The released judge emits a discrete verdict, ABSTAIN marking an explicit UNCERTAIN outcome (the band $[0.4, 0.6]$ on a calibrated probability); an ABSTAIN contributes 0 to $\phi(\pi)$.

This tail decays exponentially only once k/m is held fixed: the ceiling in $k = \lceil 2m/3 \rceil$ makes small-m choices non-monotone (0.028 at m=3 but 0.0815 at m=5 for $p_h = 0.9$), so deployments should pick m as a multiple of three; $m > 1$ trades inference cost for lower verification error.

4.3 PHASE 3: COMMIT

Validators broadcast votes $vote_i = (\hat{V}_i(\pi), signature_i)$ to all agents, with $v_i \in \{-1, 0, 1\}$ encoding REJECT=-1, ABSTAIN=0, ACCEPT=$+1$ over the validator set $\mathcal{V}$ (excluding the primary). The consensus aggregator computes vote count score:

$$\phi(\pi) = |\{i : v_i = \text{ACCEPT}\}| - 0.5 \cdot |\{i : v_i = \text{REJECT}\}| \tag{2}$$

Consensus Decision:

$$Decision(\pi) = \begin{cases} \text{ACCEPT} & \text{if } \phi(\pi) \geq \theta_{accept} \\ \text{REJECT} & \text{if } \phi(\pi) \leq \theta_{reject} \\ \text{PENDING} & \text{otherwise} \end{cases} \tag{3}$$

with dynamic thresholds:

$$\theta_{accept} = n - 1 - 2f - s \tag{4}$$

$$\theta_{reject} = -(n - 1 - f) \cdot 0.5 \tag{5}$$

where the threshold is derived from the safety condition $n \geq 3f + s + 1$. This ensures that if all honest validators accept, the proposal passes even with f Byzantine validators rejecting (and, when soft faults also reject, provided $s \leq f$; Appendix A.1.2).

4.4 REPUTATION MECHANISM

A2A-BFT maintains per-agent reputation scores $r_i \in [0.1, 1.0]$ updated each round. Algorithm 1 tiers penalties by the historical rejection ratio ρ_i over a 10-round window; vote correctness is judged against *proposal quality* (c), not the committee's final decision, so a validator that correctly rejects

Algorithm 1 Reputation Update with Tiered Penalties

Require: Reputation r_i; vote v_i; proposal correct? c; reject ratio ρ_i (10-round window)
Ensure: Updated reputation $\hat{r}_i$

ok $\leftarrow (v_i = \text{ACCEPT} \wedge c) \vee (v_i = \text{REJECT} \wedge \neg c)$ ▷ ABSTAIN is never "correct"
if $\rho_i \geq 0.7$ **then** ▷ persistent rejecter
  $\hat{r}_i \leftarrow \max(0.1,\ r_i - 0.25)$
else if ok $\wedge\ \rho_i < 0.5$ **then** ▷ correct vote
  $\hat{r}_i \leftarrow \min(1.0,\ r_i + 0.1)$
else if $\neg$ok **then** ▷ honest mistake
  $\hat{r}_i \leftarrow \max(0.5,\ r_i - 0.05)$
else ▷ neutral drift
  $\hat{r}_i \leftarrow \max(0.5,\ r_i - 0.02)$
end if
return $\hat{r}_i$

a wrong proposal is *specified* to be rewarded even if the majority accepts it. Section 6.6 measures what the released implementation actually delivers.

The reputation score is an incentive and accountability signal, not an input to the vote-counting rule: Equation 2 aggregates raw vote *counts*, so reputation cannot alter a commit decision—the sweep runs with uniform weights (w_i=1) and contains no reputation logic in its consensus path (Section 6). The reference implementation maintains per-agent scores, a 10-round voting history, and suspect counters, and escalates penalties for persistent misbehavior (Algorithm 1); weighting validator selection and excluding persistent offenders are specified extensions the released code does *not* realize, and no deployment here subsamples validators. The tiered structure (aggressive -0.25 for persistent rejecters, reward $+0.1$ for correct votes, mild -0.05 for honest mistakes, neutral drift -0.02) is *intended* to provide graduated deterrence against strategic rejection; Section 6.6 measures the released trigger against that intent. ABSTAIN is never "correct": the predicate of Algorithm 1 is a disjunction rather than the form $(v_i = \text{ACCEPT}) = c$, which would score abstention on an *incorrect* proposal as correct and could pay the reward for it (Section 6.6).

4.5 View Change Protocol

When the primary fails, when a proposal is rejected, after two consecutive PENDING rounds, or on a 2Δ timeout, a view change is triggered:

1. Any honest agent may initiate view change after 2Δ timeout.
2. Propose VIEW-CHANGE message with new view number.
3. Collect $2f + 1$ confirmations to establish new primary.
4. Switch to new primary and resume consensus.

Safety across views holds because the primary of each view number is fixed by deterministic round-robin order, any $2f{+}1$ confirmation quorum contains at least $f{+}1$ non-Byzantine agents—hence at least $f{+}1{-}s \geq 1$ strictly honest ones (non-Byzantine is weaker than honest when soft faults exist)—and the at most f Byzantine agents cannot establish a view; a committed task terminates its instance. Cross-view state transfer and the assumptions it rests on are detailed in Appendix A.1.3.

4.6 Alternative Proposal Handling

When a proposal is rejected or times out, secondary validators may submit competing proposals, which enter the PENDING state for consideration next round, with priority to higher-confidence submissions.

5 THEORETICAL ANALYSIS

5.1 SAFETY THEOREM

Theorem 5.1 (Safety)**.** *Under $n \geq 3f + s + 1$, DID-authenticated channels, single-proposal broadcast with consistent dissemination of honest votes, and worst-case soft-fault abstention without false-positive ACCEPTs on invalid proposals (Definition 3.1), A2A-BFT satisfies: (i)* decision consistency*: no two honest validators reach* opposite *terminal decisions on the same proposal, and honest validators may differ only between ACCEPT and PENDING, which the commit broadcast or view change resolves; and (ii)* commit validity*: within the boundary a wrong commit requires an honest validation error (or a wrong honest-primary proposal), whereas below the boundary Byzantine validators alone can commit a tampered proposal.*

Proof sketch. (i) Honest replicas share the same honest votes, so scores differ only through Byzantine vote equivocation, which the threshold gap provably absorbs ($\theta_{accept} - \theta_{reject} \geq 2f + 0.5s > 1.5f$; Appendix A.2); opposite terminal decisions are therefore impossible, and ACCEPT-vs-PENDING divergence resolves via the commit broadcast or view change. Proposals carry the primary's DID signature, and conflicting signed proposals from one primary trigger a view change that voids the round, so equivocation cannot yield two concurrent acceptances. (ii) Byzantine validators number at most $f - 1$ when the primary is Byzantine (f otherwise) and contribute $+1$ each, so commitment requires at least $\theta_{accept} - (f-1) = n-3f-s \geq 1$ honest ACCEPT votes when the primary is Byzantine, and $\theta_{accept} - f = n-1-3f-s \geq 0$ otherwise. Within the boundary, then, a wrong proposal cannot be committed through Byzantine votes alone—it needs an honest validation error or a wrong honest-primary proposal, both governed by the oracle's error rates (p_h, p_{mr}; Section 4.2). Below the boundary the implication fails instead through the honest validators' own REJECTs: a non-tampering primary's wrong proposal, rejected by $n-1-f-s$ honest validators (soft faults abstaining, the worst case) against f Byzantine ACCEPTs, clears θ_{accept} exactly when $7f + 3s + 3 \geq 3n$, matching Table 1. Full proof in Appendix A.2. □

Corollary 5.2 (Oracle composition)**.** *Within the boundary, every committed wrong answer implies a validation-oracle error (claim (ii) of Theorem 5.1): a false accept by an honest validator, or—at the exact boundary $\theta_{accept} = f$—a collective honest abstention carried by the f Byzantine ACCEPTs. The wrong-commit rate is governed by the oracle's miss probability p_{mr} (Section 4.2) and is zero under a perfect oracle; measured on the released GSM8K vote streams, the honest validators' false-accept rate is* 18.1% *(Wilson* 95% *CI* $[15.5, 21.2]$*; Appendix A.6), the same order as the* 11.6–23.2% *wrong-commit rates observed under attack (Table 1).*

5.2 LIVENESS THEOREM

Theorem 5.3 (Liveness)**.** *Under partial synchrony, $n \geq 3f + s + 1$, and the standing assumption $s \leq f$, A2A-BFT reaches a decision (ACCEPT or REJECT) for any valid task proposal with probability* 1—including *under sustained coordinated rejection of every honest proposal by the Byzantine validators. Coordinated rejection blocks termination only if soft faults vote REJECT (rather than abstain) and $f < s$.*

Proof sketch. The primary rotates round-robin, so an honest agent becomes primary within every $f + 1$ rounds. With an honest primary, a sufficient event for acceptance is that all $(n - 1 - f - s)$ honest validators accept; Lemma 5.4 lower-bounds the single-round acceptance probability by $p_h^{(n-1-f-s)} = 0.81$ for n=5, f=1, s=1 ($p_h = 0.9$). This event survives coordinated rejection: all-honest-accept with soft faults abstaining (the worst-case convention of Theorem 5.1) gives $\phi - \theta_{accept} = 0.5f \geq 0$ *regardless of the Byzantine votes*. If the decision is PENDING, view change (requiring $2f + 1$ confirmations) rotates the primary within $f + 1$ further rounds. After GST, the probability of no acceptance within k honest-primary rounds is bounded by $(1 - p_h^{(n-1-f-s)})^k$, so termination occurs with probability 1 as rounds grow; the bound is informative only for large p_h, and realized round-bounded rates are measured in Section 6. The assumption $f \geq s$ enters only when soft faults vote REJECT; if $f < s$, sustained coordinated rejection can block acceptance—the sole case outside the guarantee. Full proof in Appendix A.2. □

Lemma 5.4 (Acceptance Probability Bound). *For $n \geq 3f + s + 1$ and honest primary, the probability of acceptance in a single round is at least $p_h^{(n-1-f-s)}$, under conditional independence of validator errors given the proposal (an idealization: measured judge errors are model-correlated, Appendix A.6).*

Proof. When the primary is honest, it generates a valid proposal. Each honest validator independently accepts with probability p_h. Since there are $(n-1-f-s)$ honest validators, the probability that all accept is $p_h^{(n-1-f-s)}$. All-accept ensures $\phi \geq \theta_{accept}$ under the stated abstention behavior of soft faults; if the f Byzantine *and* s soft-fault validators all vote REJECT the implication requires $f \geq s$, as assumed throughout (Appendix A.1.2). □

5.3 Complexity Analysis

Table 2 (Appendix A.4) compares message complexity with that of representative BFT protocols. A2A-BFT keeps $O(n^2)$ messages per round, like PBFT; measured rounds remain small (2.6–4.5, Table 11) because fault-adaptive thresholds and view change restore progress under attack.

6 Experimental Evaluation

6.1 Experimental Setup

Reproducibility: All protocol-level fault-tolerance experiments share one heterogeneous four-model deployment with 5 seeds and 50 tasks per (configuration, seed, dataset), i.e., n=250 per cell (7,750 tasks over 31 cells); task sets are per-seed shuffles over GSM8K (Cobbe et al., 2021), MBPP (Austin et al., 2021), and MMLU (Hendrycks et al., 2020), reused across configurations for paired comparison. **No simulation:** every reported number comes from real LLM inference (7,750 fault sweep + 4,320 ablation/baseline + 770 API tasks); seeds, models, hardware, and injection are specified in the Reproducibility Statement and Appendix A.3.

Statistical Methodology: results are reported as mean ± standard deviation over seeds with 95% Wilson score intervals on pooled task counts; rounds are compared by two-sided Welch t-tests and decision rates by the exact McNemar test, both over identical per-task sampling; tests are confirmatory only for the pre-specified round-count and run-agreement comparisons of Section 6.4, all other between-cell comparisons are descriptive.

6.2 Baseline Results

Table 6 (Appendix A.7) shows baseline performance without Byzantine faults.

Note on decision semantics: a task is *decided* when a proposal is committed within the 6-round bound (rejections and PENDING rounds trigger view change and retry); *wrong-commit rate* counts tasks committed with an incorrect answer. Under semantic validation the single-round acceptance bound $p_h^{n-1-f-s}$ is consistent with the 55–59% fault-free decision rate on GSM8K—heterogeneous replicas disagree often, and the protocol refuses to commit when validation is inconclusive; on MBPP, where validation executes all test assertions, it is 70–74% with *zero* wrong commits.

6.3 Byzantine Fault Tolerance

Table 1 reports consensus performance under the three Byzantine attack types across fault configurations.

Three findings emerge. **(i) Safety is exactly as strong as the validation oracle; the boundary is tight.** On MBPP, every configuration within the safety boundary commits a wrong answer in $\mathbf{0.0 \pm 0.0}\%$ of tasks (n=250/cell; zero events; per-cell 95% Wilson upper bound 1.5%) regardless of attack, whereas all n=5, f=2 configurations below the boundary violate safety (5.6–9.6%): $\theta_{accept} = 0$ lets two Byzantine ACCEPTs offset the honest REJECTs of a non-tampering primary's wrong proposal (the n=6 MBPP row stays at 0.0%: execution mechanically rejects tampered code,

Table 1: BFT Performance Under Attack (real heterogeneous 4-model deployment; cells: decision% / wrong-commit%, mean±std over sampling seeds; n=250 per cell except the n=5, f=1, s=1 strategic-reject row, n=90, from Table 7, Appendix A.7). Bold marks the *best* (lowest) wrong-commit rate per column. †: configuration below the safety boundary $3f+s+1$ ($n \geq 7$ required for f=2). ‡: smaller three-seed ablation deployment; not comparable to the n=250 rows. MMLU results in Table 10.

n	f	s	Attack	GSM8K (semantic)		MBPP (execution)	
				decision	wrong	decision	wrong
4	1	0	Random	90.8 ± 3.0	14.0 ± 4.0	73.6 ± 6.5	$\mathbf{0.0 \pm 0.0}$
4	1	0	Strat. Reject	80.8 ± 3.6	23.2 ± 7.0	71.2 ± 6.4	$\mathbf{0.0 \pm 0.0}$
4	1	0	Sybil	90.4 ± 3.3	12.8 ± 4.4	73.6 ± 6.5	$\mathbf{0.0 \pm 0.0}$
5	1	1	Random	89.2 ± 3.0	11.6 ± 3.8	73.2 ± 7.2	$\mathbf{0.0 \pm 0.0}$
5	1	1	Strat. Reject‡	74.4 ± 8.4	$\mathbf{4.5 \pm 3.9}$	70.0 ± 11.5	$\mathbf{0.0 \pm 0.0}$
5	2	0	Random†	96.8 ± 3.6	22.0 ± 5.5	78.8 ± 6.1	5.6 ± 1.7
5	2	0	Strat. Reject†	95.2 ± 2.3	67.6 ± 7.4	74.4 ± 6.2	9.6 ± 1.7
5	2	0	Collusion†	98.4 ± 1.7	72.4 ± 5.0	75.6 ± 4.8	9.2 ± 5.0
6	2	0	Collusion†	82.4 ± 3.3	25.2 ± 1.8	66.4 ± 4.8	$\mathbf{0.0 \pm 0.0}$

so its violation surfaces through text proposals; Appendix A.2). **(ii) On semantic domains, eventual acceptance trades safety for liveness under sustained attack.** GSM8K wrong commits rise to 11.6–23.2% within the boundary: a Byzantine primary submits a plausible-but-wrong answer, weak 8B judges accept it at non-negligible rates, and retries sample until one passes validation—the system *does* decide (74–91%) but inherits the judge's error rate (held-out, two of the four MMLU judges accept wrong answers at 65–79%, Appendix A.6, consistent with Table 10's 35.6–60.8%). **(iii) Decision rates rise under attack** (at matched n=5 within the boundary, 89.2% vs. 56.7% fault-free; sub-boundary rows reach 98%): Byzantine primaries inject extra proposals that eventually clear validation; we therefore report wrong commits as the primary safety metric.

6.4 Real LLM Validation (Homogeneous Upper Bound)

As a complementary check we validated the protocol on 320 real DeepSeek-V2-Lite tasks (GSM8K, MMLU, MBPP). These homogeneous single-model runs let honest validators agree far more often; the resulting 100% acceptance across all eight configurations (291 of 320 completed; 29 API/harness failures excluded, Appendix A.5), under strategic rejection and collusion alike, is therefore an *upper bound* on decision rates, not a contradiction of the 55–74% heterogeneous results. At the boundary configuration n=8, f=2, s=1 (150 GSM8K tasks per configuration, real DeepSeek API), the protocol maintains 98–100% consensus under both attacks across two independent runs, with answer correctness within 2 points of the fault-free baseline (92.0–92.7% vs. 94.0%), and the round-count gap vs. fault-free (1.03 vs. 2.89–2.92) confirms genuine attack resistance (Welch $t = -13.1$ and -13.9, $p < 10^{-26}$; reproduced at 1.04 vs. 2.38–2.43, with per-task decision agreement across runs, McNemar exact $p = 0.25$). Details, scaling, and MMLU sweeps: Appendices A.5 and A.7.

6.5 Ablation Study

To isolate each protocol component we ablate in the *heterogeneous four-model* setting (Section 6.1 models, one per replica) on two domains with different validation semantics: *GSM8K*, where Validate uses an independent LLM judge, and *MBPP*, where Validate executes the submitted code against test assertions. Each scenario uses 30 tasks per seed over three seeds (90 each); Byzantine validators employ strategic rejection (n=5, f=1, s=1) or collusion (n=8, f=2, s=1). Soft faults exercise the abstention arm (ABSTAIN w.p. 0.3; Appendix A.1.2)—the stochastic-REJECT arm of Definition 3.1 is analyzed, not injected (Table 7, Appendix A.7, reports decision and wrong-commit rates).

The ablation yields four findings. **(i) View change is the core liveness component:** removing it collapses decisions from 74.4 ± 8.4% to 4.5 ± 3.9% (GSM8K, strategic rejection) and to 0% (MBPP, both attacks)—blocked rounds never rotate the primary, confirming Theorem 5.3. **(ii) Semantic validation is the prerequisite for heterogeneous consensus:** under naive exact matching heteroge-

neous replicas almost never agree byte-for-byte and the code domain deadlocks (0–1.1% decisions). On GSM8K, dropping semantic validation *lowers* wrong commits ($2.2 \pm 1.9\%$ vs. the full protocol's $4.5 \pm 3.9\%$)—as Corollary 5.2 predicts: the exact-match oracle rarely commits and so rarely errs; the LLM-judge oracle commits more and inherits its error rate—a 20-point decision-rate trade. **(iii) Fault-adaptive thresholds prevent attack-induced deadlock:** a fixed majority threshold drives every strategic-rejection round into PENDING ($0.0 \pm 0.0\%$), whereas fault-adaptive thresholds restore $74.4 \pm 8.4\%$. **(iv) Safety is oracle-bounded:** at most $10.0 \pm 3.3\%$ wrong commits on GSM8K and none on MBPP (per-cell 95% Wilson interval $[0.0, 4.1]\%$ at n=90).

Baselines under attack (Table 8, Appendix A.7; definitions: Appendix A.3). Three observations (descriptive; three seeds per cell). **First, high decision rates are not safety.** All voting- and debate-based baselines decide on every task, yet under collusion commit incorrect answers in 16.7–63.4%—unvalidated aggregation converts Byzantine injections and honest errors into committed consensus; the full protocol forgoes them (60–74%). **Second, unverified debate amplifies attacks:** LLM-Debate (Du et al., 2024) is the least safe on code ($63.4 \pm 5.8\%$ wrong commits under collusion), letting Byzantine agents drag honest replicas toward corrupted solutions. **Third, the correct-when-decided gap matters:** conditioned on committing, A2A-BFT reaches 84–100% answer correctness in all eight attacked cells (Tables 7, 8) vs. 37–88% for baselines; vs. A2A-Sim: GSM8K wrong commits within noise, but it decides fewer tasks and collapses on code (34.4–35.5% vs. our 0.0 ± 0.0).

6.6 Reputation Tracker: Measured Behavior

The tracker cannot change a commit decision (Section 4.4); we measure *what it reports*: logging every validation vote from the four-model deployment (120 runs, 457 scored rounds), we replay them offline through Algorithm 1 under four settings (two c sources $\times$ two reset policies; Appendix A.8). Decision-neutrality is *checked* rather than assumed: recomputing ϕ from the raw votes reproduces all 457 recorded rounds; replaying the vote-and-threshold state machine reproduces all 120 decisions; injecting reputation weights flips up to 5 decisions, so the check can fail.

The measured behavior is negative. **(i) The aggressive tier fires on honest, correct votes:** across the four attacked cells, 1,186 of 1,305 aggressive penalties ($\rho_i \geq 0.7$) landed on honest validators whose votes were *objectively correct* (the reward Section 4.4 specifies)—and *all 20 tasks* per cell had a mis-penalty. **(ii) As a Byzantine detector the tier fails:** precision 0.232/0.195 (recall 1.000) under strategic rejection but precision *and* recall 0.000 under collusion—colluders never reach $\rho = 0.7$; honest validators rejecting the same proposals do. **(iii) The score ordering inverts:** honest validators finish *below* Byzantine ones in both collusion cells (0.337 vs. 0.427 GSM8K, 0.300 vs. 0.412 MBPP), and per-task reset does not repair this (0.391 vs. 0.761), so the failure is structural, not a cold-start artifact. **(iv) The released c carries negative information:** `_estimate_correctness` is a per-domain constant (always `False` on GSM8K, `True` on MBPP) agreeing with ground truth on 46.2% of rounds (trivial always-`False` predictor: 74.6%).

This failure mode is known (Jøsang & Ismail, 2002; Jøsang et al., 2007; Kamvar et al., 2003); we add its scale in a real LLM deployment. No reported safety or liveness *number* depends on the tracker; the Appendix A.9 deterrent is specified, not realized. A repaired trigger needs a proposal-quality oracle and a key untied to the validator's own rejection ratio (Appendix A.8).

Limitations. A2A-BFT assumes DID-authenticated channels (Ed25519); fault-free decision rates (55–59% GSM8K, 11–15% MMLU) show weaker semantic-validation reliability, and Theorem 5.3's asymptotic guarantee is unreached within six rounds. Ablation cells (n=90) limit statistical power; the tracker is decision-neutral yet mis-specified (single seed).

7 Conclusion

We presented A2A-BFT, a Byzantine fault-tolerant consensus protocol for agent collaboration with provable safety and liveness under $n \geq 3f+s+1$ ($s \leq f$). Real-LLM evaluation (7,750 tasks) shows liveness under every attack and theory-predicted safety: zero wrong commits on execution-validated tasks within the boundary, violations below it, oracle-bounded errors on semantic tasks. Two limits remain: the measured reputation tracker is mis-specified; realized liveness on weak oracles falls short (implications, privacy: Appendix B).

AI USE STATEMENT

We used generative AI tools (large language model assistants) in the three roles described below. (i) *Software development*: they assisted in writing and debugging the implementation of A2A-BFT and of the baselines, the experiment harness that drives the real-LLM runs, the verification suite (ten independent auditors and five negative controls), and the scripts that regenerate the figures and the aggregate statistics. (ii) *Documentation*: they assisted in drafting and editing the documentation released with the artifact (repository README, reproduction guide, and audit notes). (iii) *Manuscript preparation*: they assisted in editing the language of this paper, and served as review-style critics whose suggested revisions to wording, comparisons, and the statement of assumptions were each evaluated by the authors and adopted only where verified correct. No experimental data or result was generated by an AI tool: every number reported here is an output of the released scripts running real LLM inference, and the released audit suite recomputes it independently of this paper. Every AI-assisted artifact was executed and reviewed by the authors, who take responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

All source code, experimental data, and model configurations are available at `https://github.com/A2A-hub-ai/a2a-bft` (anonymized for review). Section 6.1 summarizes the setup; in full, all protocol-level fault-tolerance runs use the same heterogeneous four-model deployment with 5 seeds ($\{42, 43, 44, 45, 46\}$) and 50 tasks per (configuration, seed, dataset) (n=250 per cell; 7,750 tasks over 31 cells), with task sets drawn per seed by a seeded shuffle over GSM8K, MBPP, and MMLU and reused across configurations for paired comparison, while ablation and baseline comparisons use 3 seeds ($\{42, 43, 44\}$) with n=90 per cell. The four models (Llama-3.1-8B, DeepSeek-V2-Lite, InternLM3-8B, Qwen2.5-7B; Instruct/Chat variants) are served by vLLM (Kwon et al., 2023) 0.11 (temperature 0, max_tokens 512) on two NVIDIA A800-SXM4-80GB GPUs (CUDA 12.8, PyTorch 2.8.0). No simulated workers are used anywhere: every reported number comes from real LLM inference—7,750-task fault sweep; 4,320 ablation/baseline tasks (48 cells $\times$ 90); and 770 API tasks (320 multi-domain validation, Table 3, and 450 adversarial-boundary scaling, Table 9). Of the 48 ablation/baseline cells, 36 are tabulated (Tables 7 and 8); the remaining 12 are fault-free baseline-condition pairings retained in the released aggregates. The reputation-tracker measurement of Section 6.6 additionally releases the fully instrumented vote streams (`reputation_vote_stream.json`) for 120 consensus tasks (6 cells $\times$ 20), the two-phase measurement-and-replay script, the decision-neutrality audit (which recomputes the tally from raw votes, reproduces the recorded decisions, and mutation-tests its own resolution), and the fidelity audit against the released penalty code, so that every reported tracker quantity can be recomputed offline without any LLM calls; those quantities are event counts, reported without a variance.

REFERENCES

Anthropic. Model Context Protocol (MCP) Specification. `https://github.com/modelcontextprotocol/specification`, 2024. Open standard for connecting LLM applications to data sources and tools.

Jacob Austin, Augustus Odena, Maxwell Nye, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Frédéric Berdoz, Leonardo Rugli, and Roger Wattenhofer. Can AI agents agree? *arXiv preprint arXiv:2603.01213*, 2026.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ethan Buchman, Jae Kwon, and Zarko Milosevic. The latest gossip on BFT consensus. *arXiv preprint arXiv:1807.04938*, 2018.

Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *OSDI*, pp. 173–186, 1999.

M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, B. Chopra, R. Tiwari, et al. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. arXiv:2503.13657.

B. Chen, G. Li, X. Lin, Z. Wang, and J. Li. BlockAgents: Towards byzantine-robust LLM-based multi-agent coordination via blockchain. In *ACM Turing Award Celebration Conference (TURC)*, pp. 187–192, 2024. doi: 10.1145/3674399.3674445.

H. Chen, W. Ji, L. Xu, and S. Zhao. Multi-agent consensus seeking via large language models. *arXiv preprint arXiv:2310.20151*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI. DeepSeek-V2: A strong, economic, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *ICML*, 2024. arXiv:2305.14325.

Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, 1988.

A. Ehtesham, A. Singh, G. K. Gupta, and S. Kumar. A survey of agent interoperability protocols: MCP, ACP, A2A, and ANP. *arXiv preprint arXiv:2505.02279*, 2025.

F. Grötschla, L. Müller, J. Tönshoff, M. Galkin, and B. Perozzi. AgentsNet: Coordination and collaborative reasoning in multi-agent LLMs. *arXiv preprint arXiv:2507.08616*, 2025.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *IJCAI*, 2024. arXiv:2402.01680.

Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2308.00352.

InternLM Team. InternLM3-8B-Instruct model card. `https://huggingface.co/internlm/internlm3-8b-instruct`, 2025. Released 2025-01-15; predecessor series documented in the InternLM2 technical report (arXiv:2403.17297).

Y. Jo and C. Park. Byzantine-robust decentralized coordination of LLM agents. *arXiv preprint arXiv:2507.14928*, 2025.

Audun Jøsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, pp. 324–337, 2002.

Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on World Wide Web (WWW)*, pp. 640–651, 2003.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

Haejoon Lee, Vincent-Daniel Yun, Hyeonho Oh, Dimitra Panagou, and Sai Praneeth Karimireddy. Robust multi-agent LLMs under byzantine faults. In *EMNLP*, 2026. arXiv:2605.09076.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2303.17760.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, 2023. arXiv:2305.19118.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.20050.

Linux Foundation. Agent2Agent (A2A) Protocol Specification. `https://github.com/a2aproject/A2A`, 2025. Donated by Google and hosted by the Linux Foundation; announced at Google I/O 2025.

Meta AI. Llama 3 technical report. *arXiv preprint arXiv:2407.21783*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2303.11366.

W3C. Decentralized identifiers (DIDs) v1.0. `https://www.w3.org/TR/did-core/`, 2022. W3C Recommendation.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2203.11171.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *First Conference on Language Modeling (COLM)*, 2024. arXiv:2308.08155.

Michael Yin, Dahlia Malkhi, Michael K. Reiter, Ittai Golan-Gueta, and Ittai Abraham. HotStuff: BFT consensus with linearity and responsiveness. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 347–356, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Lifan Zheng, Jiawei Chen, Qinghong Yin, Jingyuan Zhang, Xinyi Zeng, and Yu Tian. Rethinking the reliability of multi-agent system: A perspective from byzantine fault tolerance. In *AAAI*, pp. 35012–35020, 2026. doi: 10.1609/aaai.v40i41.40806. arXiv:2511.10400.

A Appendix

A.1 Proof Details

Additional proof details and extended theorems are provided in this appendix.

A.1.1 Reputation Weights and Safety

The Safety Theorem (Theorem 5.1) is stated for the unweighted vote count $\phi(\pi)$, and this is the rule the reference implementation uses: reputation scores r_i never enter Equation 2. Reputation affects only (i) the score update and penalty schedule of Algorithm 1 and (ii) the suspect counters used for anomaly detection (Section 4.4); it does not appear in Equation 2. Selection weights $w_i = w_i^{\text{base}} \cdot \max(r_i, 0.3) \in [0.1, 2.0]$ and exclusion of persistent offenders are specified for deployments that subsample validators; *the released implementation contains neither, and no experiment in this paper subsamples validators*. Since the consensus decision aggregates raw vote counts, the proof of Theorem 5.1 applies unchanged. We deliberately do not analyze a weighted-vote variant here: weighting votes by w_i would couple the threshold analysis to reputation dynamics and require re-deriving θ_{accept}; we leave that variant, which trades a stronger incentive channel for a more complex safety argument, as future work.

A.1.2 Soft Fault Adversarial Behavior

Theorem 5.1 assumes soft-fault agents abstain in the worst case (Definition 3.1 additionally excludes false-positive ACCEPT votes on invalid proposals, which the analysis below relaxes). If soft-fault agents vote adversarially (REJECT on valid proposals), the minimum vote count becomes:

$$\phi_{min}^{\text{adversarial}} = (n - 1 - f - s) - 0.5f - 0.5s = n - 1 - 1.5f - 1.5s \tag{6}$$

Substituting $n = 3f + s + 1$:

$$\phi_{min}^{\text{adversarial}} = 1.5f - 0.5s \tag{7}$$

For $s = 0$, this equals $1.5f > f = \theta_{accept}$. For $s > 0$, the bound weakens but $\phi_{min}^{\text{adversarial}} = 1.5f - 0.5s \geq f = \theta_{accept}$ whenever $s \leq f$, with strict inequality for $s < f$. Thus, the safety condition $n \geq 3f + s + 1$ is robust even against adversarial soft faults that *reject* valid proposals, when $s \leq f$.

False positives on invalid proposals. The more dangerous direction is a soft-fault agent that ACCEPTs an invalid proposal. Let each soft-fault agent do so with probability $p \in [0, 1]$ and abstain otherwise, and let honest validators reject. With a Byzantine primary the wrong-side score is $\phi = (f-1) + sp - 0.5(n-f-s)$, and safety requires $\phi < \theta_{accept} = n - 1 - 2f - s$, i.e.

$$7f + 2sp + 3s < 3n. \tag{8}$$

Substituting the boundary condition $n = 3f + s + 1$ reduces this to the single requirement $sp < f + 1.5$: with $p = 1$ it is $s \leq f + 1$, and for $f = 0$ it becomes $sp < 1.5$, so a deployment with many unreliable agents relative to its Byzantine count must either bound their false-positive rate or enlarge n to $n > (7f + 2sp + 3s)/3$. All experiments in this paper use $s \in \{0, 1\}$ and $f \geq s$, comfortably inside the condition; we state it because it is the one place where the boundary $n \geq 3f + s + 1$ alone is not sufficient.

Injection used in the evaluation. The released soft-fault agents ABSTAIN with probability 0.3 (and, as primary, return an empty proposal with probability 0.3), so the experiments exercise the abstention arm—which for $s \leq f$ is the worst case for the accept threshold—together with the stochastic-REJECT arm covered analytically above. We note explicitly that the stochastic-REJECT behavior of Definition 3.1 is analyzed rather than injected.

A.1.3 View Change Protocol Formalization

The view change protocol ensures that at most one primary is active per view. When a VIEW-CHANGE is initiated:

1. Honest agents broadcast VIEW-CHANGE with the highest view number they have observed.
2. The new primary is selected as the agent with view number $v' = v+1$ in round-robin order.
3. $2f + 1$ VIEW-CHANGE confirmations are required to establish the new view.
4. The new primary includes the highest-confidence proposal from the previous view (if any) in its new proposal.

Safety across views rests on three mechanisms rather than on quorum intersection alone: (i) the primary for view v' is a deterministic function of the view number (round-robin), so two conflicting views with the same number cannot be established; (ii) any $2f + 1$ confirmation quorum contains at least $f + 1$ non-Byzantine agents (and hence at least $f{+}1{-}s$ strictly honest ones when soft-fault agents join it), so the at most f Byzantine agents cannot establish a view unilaterally; and (iii) a committed task terminates its instance, so a later view cannot reverse a prior commitment. Note that quorum intersection of two $2f{+}1$ quorums ($\geq 4f{+}2{-}n$ agents) is not itself the load-bearing argument here, since mechanism (i) already pins the primary.

Cross-view state transfer. The protocol deliberately does not carry quorum certificates across views. A view change only rotates the primary deterministically; a committed task terminates its instance, and the new primary re-proposes (or promotes the highest-confidence pending alternative). Under partial synchrony a replica that observes $\phi \geq \theta_{accept}$ broadcasts the commit, so a slower replica either adopts that commitment or, if it times out first, enters a view in which the instance has already been terminated by the commit; the two cannot diverge because the primary of a given view number is unique, so there is never a second decision for the same instance. The load-bearing assumption is therefore that a committed task is never re-instantiated, which the reference implementation enforces by binding each task instance to its committed round. The residual exposure is the converse of this choice: a proposal that has *not* yet committed is not transferred, so progress after a view change costs a repeat generation. A two-phase commit with signed commit certificates in the VIEW-CHANGE message (as in PBFT (Castro & Liskov, 1999) or HotStuff (Yin et al., 2019)) would remove the assumption and preserve partially committed state, at the price of an extra message round; we regard this as the natural next step for a distributed deployment (Appendix A.12).

Self-assessed confidence. Alternative proposals are prioritized by the proposer's own confidence (Section 4.6), which a Byzantine agent can set to 1.0. Priority only selects *which* pending alternative is re-proposed; it never enters $\phi(\pi)$ or the thresholds, so a self-promoted proposal still has to clear the same vote count, and on execution-validated domains it still has to pass the tests. Hardening the heuristic (e.g. ranking by execution outcome, or dropping priority entirely) is orthogonal to the safety theorem, which depends only on vote counts.

A.1.4 Boundary Configuration Analysis

The configuration $n = 4, f = 1, s = 0$ sits exactly on the safety boundary ($n = 3f + s + 1 = 4$). In this case, $\theta_{accept} = 4 - 1 - 2(1) - 0 = 1$ and $\phi_{min} = 1.5(1) = 1.5$, so $\phi_{min} > \theta_{accept}$ holds. However, any deviation from the assumed fault counts (e.g., an additional soft fault) would violate the safety condition. We report this configuration for completeness but recommend $n \geq 5$ for practical deployments where fault counts may be uncertain.

Note on sub-boundary configurations ($f{=}2$, $n \leq 6$): The configurations $n = 5, f = 2, s = 0$ and $n = 6, f = 2, s = 0$ violate the theoretical safety boundary ($3f + s + 1 = 7 > n$). The real-LLM results (Table 1) show exactly what the theorem predicts: with $\theta_{accept} = n - 1 - 2f - s \leq 1$ the honest majority no longer outvotes the Byzantine bloc (f ACCEPT votes against $n{-}1{-}f{-}s$ honest REJECTs, soft faults abstaining, clear θ_{accept} whenever $7f{+}3s{+}3 \geq 3n$), and wrong-commit rates rise to 22.0–72.4% (GSM8K) and 5.6–9.6% (MBPP). Notably, on MBPP the $n{=}6, f{=}2$ collusion row still commits $0.0 \pm 0.0\%$ wrong answers: tampered code is rejected mechanically by execution, so the violation there surfaces through degraded thresholds interacting with plausible wrong *text* proposals, not through code tampering. These rows empirically demonstrate that the boundary is tight, not a safety claim.

Author Contributions

Omitted for double-blind review.

Acknowledgments

Omitted for double-blind review.

A.2 Full Proofs

A.2.1 Full Proofs of Theorem 5.1, Corollary 5.2, Theorem 5.3, and Lemma 5.4

Proof. We prove safety by establishing six properties: (1) threshold consistency, (2) vote aggregation correctness, (3) single proposal per round, (4) commit validity, (5) PENDING-state preservation, and (6) view-change safety.

Property 1: Threshold Consistency. The accept threshold is defined as:

$$\theta_{accept} = n - 1 - 2f - s \tag{9}$$

where $n-1$ is the number of validators (excluding primary), f is the Byzantine fault count, and s is the soft fault count. This threshold is derived from the safety condition and is deterministic and publicly known.

Property 2: Vote Aggregation Correctness. The consensus score $\phi(\pi)$ aggregates votes as:

$$\phi(\pi) = |\{\text{ACCEPT}\}| - 0.5 \cdot |\{\text{REJECT}\}| \tag{10}$$

where ABSTAIN votes contribute 0. All honest validators compute $\phi(\pi)$ using the same formula (Equation 2) over the same *honest* vote set—authenticated channels plus consistent dissemination of honest votes prevent forgery and divergent honest views. Byzantine validators may still equivocate by sending different signed verdicts to different replicas, which shifts a replica's score by at most $1.5f$; since $\theta_{accept} - \theta_{reject} = 1.5(n-1) - 2.5f - s \geq 2f + 0.5s > 1.5f$ for $f + s > 0$, no two honest validators can reach opposite terminal decisions, establishing claim (i) in its decision-consistency form.

Property 3: Single Proposal per Round. Proposals carry the primary's DID signature and are broadcast to all validators. If a Byzantine primary sends conflicting proposals $\pi_1 \neq \pi_2$ to different validators in the same round, any honest replica observing both signatures immediately initiates view change and no proposal is accepted in that round. Thus equivocation cannot produce two concurrent acceptances, and at most one proposal is committed per task instance (a commitment terminates the instance).

Property 4: Commit Validity. Byzantine validators number at most $f-1$ when the primary is Byzantine and f otherwise; each contributes at most $+1$ to ϕ. Committing π with $\phi(\pi) \geq \theta_{accept}$ therefore requires at least

$$\theta_{accept} - (f-1) \;=\; n - 3f - s \;\geq\; 1 \tag{11}$$

honest ACCEPT votes when the primary is Byzantine (using $n \geq 3f + s + 1$), and at least $\theta_{accept} - f = n - 1 - 3f - s \geq 0$ when the primary is honest. So within the boundary a wrong proposal cannot be committed through Byzantine votes alone: it requires an honest validation error, or a wrong proposal from the honest primary itself—both governed by the oracle's error rates (p_h, p_{mr}; Section 4.2). At the exact boundary ($\theta_{accept} = f$), all f Byzantine ACCEPTs plus universal honest abstention also commit a wrong proposal ($\phi = f$)—a collective miss, charged to p_{mr}. Below the boundary this implication fails, but not through the Byzantine primary alone: non-Byzantine validators *actively* reject (-0.5 each), so with $f-1$ Byzantine and $n-f$ non-Byzantine validators a tampering Byzantine primary reaches only $\phi = (f-1) - 0.5(n-f) < \theta_{accept}$. What the collapsed threshold does permit is a wrong proposal from a primary that does *not* tamper, rejected by the $n-1-f-s$ honest validators (the worst case, with soft faults abstaining) while all f Byzantine validators vote ACCEPT: then $\phi = f - 0.5(n-1-f-s) \geq \theta_{accept}$ exactly when $7f+3s+3 \geq 3n$; if soft faults REJECT alongside the honest validators the violation window shrinks to $7f+2s+3 \geq 3n$. For n=5, f=2 this reads $17 \geq 15$ (violation possible) and for n=6, f=2 it reads $17 \geq 18$ (violation impossible), matching the measured execution-validated wrong-commit rates of 5.6–9.6% versus 0.0% in Table 1.

Property 5: PENDING State Preservation. If $\theta_{reject} < \phi(\pi) < \theta_{accept}$, the decision is PENDING (matching Equation 3, where $\phi \leq \theta_{reject}$ is a REJECT). In this case:

- No proposal is committed
- Validators retain their votes for potential retry

- The view change protocol is triggered after 2Δ timeout

This ensures that PENDING states never lead to inconsistent decisions: by Property 2 honest replicas cannot reach opposite terminal decisions, and a replica that remains PENDING either observes the commit broadcast from the committing quorum (becoming bound by the same commitment) or triggers view change once the timeout expires.

Property 6: View Change Safety. The primary for each view number is fixed by deterministic round-robin ordering, so two conflicting views with the same view number cannot arise. Any view-change quorum of $2f + 1$ confirmations contains at least $f + 1$ non-Byzantine agents, so the at most f Byzantine agents cannot establish a view unilaterally. Since a committed task terminates its instance, later views cannot reverse a commitment. □

Proof of claim (ii), commit validity, by contradiction. Suppose a wrong proposal π (i.e., $V(\pi) = 0$ under ground truth) is committed within the boundary with an honest primary. Then the honest primary itself produced a wrong proposal, contributing an oracle error; or, with a Byzantine primary, Property 4 shows at least $n - 3f - s \geq 1$ honest validators must have voted ACCEPT on π, i.e., at least one honest validation error occurred; or, at the exact boundary, all honest validators abstained on π (a collective miss). In all three cases the wrong commit is attributable to the validation oracle rather than to the consensus layer, and its rate is bounded by the oracle's two-sided error probability (p_h, p_{mr}). Below the boundary the implication fails exactly as Property 4 predicts, which the sub-boundary rows of Table 1 confirm empirically. □

Proof. We prove liveness by showing that the protocol terminates under the partial synchrony model with explicit bounds on rounds and probability analysis.

Step 1: Honest primary rotation. The primary rotates round-robin among all n agents. Since at most f agents are Byzantine, at least one honest agent becomes primary within every $f + 1$ rounds. Formally, in any window of $f + 1$ consecutive rounds, there exists at least one round where the primary is honest.

Step 2: Honest proposal acceptance probability. When an honest agent is primary, it generates a valid proposal π. Honest validators verify π independently. Let $p_h \geq 0.9$ be the probability that an honest validator correctly accepts a valid proposal (based on LLM accuracy; the measured value on the released GSM8K vote streams is ≈ 0.80, Appendix A.6, which slows the geometric rate but does not affect the asymptotic statements). The probability that all $(n - 1 - f - s)$ honest validators accept is:

$$P(\text{all honest accept}) = p_h^{(n-1-f-s)} \tag{12}$$

For $n = 5, f = 1, s = 1$, there are $(n - 1 - f - s) = 2$ honest validators and this gives $P = 0.9^2 = 0.81$; in that event $\phi \geq 2 - 0.5f = 1.5 > \theta_{accept} = 1$. As intuition only—$E[\phi] > \theta_{accept}$ motivates acceptance but is not itself a per-round guarantee, since ϕ is a random variable; the sufficient event below makes the argument precise—the expected score already clears the threshold:

$$E[\phi] \geq p_h \cdot (n - 1 - f - s) - 0.5f = 0.9 \cdot 2 - 0.5 = 1.3 > 1 = \theta_{accept}. \tag{13}$$

Step 3: Threshold satisfaction in expectation. By the worst-case convention of Theorem 5.1, when all honest validators accept and soft faults abstain, $\phi(\pi) - \theta_{accept} = \big[(n-1-f-s) - 0.5f\big] - (n-1-2f-s) = 0.5f \geq 0$, regardless of how the Byzantine validators vote. Acceptance can also occur with fewer honest accepts, so the bound below is a lower bound:

$$P(\text{accept}) \geq p_h^{(n-1-f-s)} \tag{14}$$

Step 4: PENDING and view change. If $\phi(\pi)$ falls strictly between θ_{reject} and θ_{accept}, the decision is PENDING. After 2 consecutive PENDING rounds, a view change is triggered:

1. Honest agents detect the persistent PENDING state
2. VIEW-CHANGE messages are broadcast with new view number
3. $2f + 1$ confirmations establish the new primary

4. Consensus resumes with the new primary

Since at least one honest agent becomes primary within $f + 1$ rounds, and the view change requires only $2f + 1$ confirmations (more than the f Byzantine agents), progress is guaranteed.

Step 5: Bounded rounds in partial synchrony. In the partial synchrony model:

- Before GST (Global Stabilization Time): messages may be delayed arbitrarily
- After GST: messages are delivered within Δ

After GST, honest agents deliver messages within Δ. The primary rotation ensures each honest agent becomes primary within n rounds. The probability of not accepting within k rounds is bounded by:

$$P(\text{no acceptance in } k \text{ rounds}) \leq (1 - p_h^{(n-1-f-s)})^k \tag{15}$$

For $p_h = 0.9$ and $n - 1 - f - s = 2$ (n=5, f=1, s=1), after $k = 10$ rounds:

$$P(\text{no acceptance}) \leq (1 - 0.81)^{10} \approx 6 \times 10^{-8} \tag{16}$$

This shows that acceptance probability approaches 1 exponentially fast.

Step 6: Coordinated rejection. Sustained coordinated rejection of every honest proposal by the f Byzantine validators does not escape the bound of Step 5: in an honest-primary round, the sufficient event of Step 3 (all honest validators accept) yields $\phi - \theta_{accept} = 0.5f \geq 0$ even when every Byzantine validator votes REJECT, provided soft faults abstain (the worst-case convention of Theorem 5.1). The standing assumption $f \geq s$ extends this to soft faults voting REJECT ($\phi - \theta_{accept} = 0.5(f - s)$, Appendix A.1.2); if both $f < s$ and soft faults vote REJECT, the attack can block acceptance—the sole case outside Theorem 5.3. In deployments, view change rotates the primary within every $f + 1$ rounds regardless, and where the reputation tracker is enabled (Algorithm 1) persistent rejecters are penalized (-0.25 per round once the reject ratio reaches 0.7)—though Section 6.6 measures that the released tracker's aggressive tier in fact fires predominantly on honest validators, so this deterrent should be read as specified rather than realized. The ablation (Table 7) confirms the practical consequence, since removing view change collapses decision rates to 0–4.5% under exactly this attack.

Conclusion: Under the standing assumptions of Theorem 5.3 ($f \geq s$), A2A-BFT eventually reaches a decision with probability 1: the probability of non-termination after k honest-primary rounds decays exponentially in k, and view change delivers an honest primary within every $f + 1$ rounds. □

Proof. When the primary is honest, it generates a valid proposal. Each honest validator independently accepts with probability p_h. Since there are $(n - 1 - f - s)$ honest validators, the probability that all accept is $p_h^{(n-1-f-s)}$. All-accept ensures $\phi \geq \theta_{accept}$ outright when soft faults abstain ($\phi - \theta_{accept} = 0.5f$, regardless of the Byzantine votes); if instead all f Byzantine *and* s soft-fault validators vote REJECT, $\phi = (n-1-f-s) - 0.5(f+s)$ and $\phi - \theta_{accept} = 0.5(f - s)$, so the implication requires $f \geq s$ as assumed throughout (Appendix A.1.2). □

A.3 Attack Injection, Sampling, and Baseline Details

A.3.1 Evaluation-Budget Asymmetry

The baselines commit in a single round, whereas A2A-BFT may retry for up to six rounds. The retries are view-change-driven responses to rejected proposals rather than budget-free resampling, and the retry budget is part of the protocol design rather than an evaluation courtesy. The oracle-assisted debate baseline—evaluated under the same six-round budget, so pure budget effects are controlled—still commits wrong answers on 46.7–56.7% of MBPP tasks under attack, indicating that validation inside the commit rule, not additional sampling, drives the safety gap.

A.3.2 BYZANTINE AND SOFT-FAULT INJECTION

Byzantine replicas occupy fixed indices (f of the n agents are flagged Byzantine at deployment). Their operational behavior is:

- **As primary:** a Byzantine primary first solves the task with its own model (masquerading as honest) and then tampers with the answer before broadcasting: on GSM8K the extracted number is perturbed by $+U(1,5)$; on MBPP the `return` statement is deleted from the generated code (deterministically breaking correctness); on MMLU a random incorrect choice is submitted. Tampered proposals carry self-assessed confidence 0.6.
- **As validator:** *strategic reject* votes REJECT unconditionally; *collusion* votes ACCEPT if and only if the primary is Byzantine and REJECT otherwise; *Sybil* solves honestly but votes REJECT with probability 0.75 (REJECT-heavy masquerade; we follow the A2A-simulation literature's usage of "Sybil" for this behavior—classical identity duplication is precluded by DID authentication); *random fault* votes ACCEPT or REJECT with equal probability.
- **Soft faults (s):** as primary, return an empty proposal with probability 0.3; as validator, ABSTAIN with probability 0.3.
- **Honest validators:** a single self-model LLM judge (semantic domains), or execution of the proposal against all MBPP test assertions (code domain); judges output ABSTAIN when the verdict is uncertain.

View change triggers immediately on an empty proposal, on rejection, after two consecutive PENDING rounds, or on a 2Δ timeout. The heterogeneous sweep and ablation bound each task at 6 rounds; the homogeneous API runs (Tables 3 and 9) use a 10-round bound, so their mean round counts are not directly comparable to the sweep's.

A.3.3 TASK SAMPLING

Per sampling seed $\sigma \in \{42, \ldots, 46\}$, the task set is `random.sample`(dataset, 50) with `random.seed`(σ), drawn once per dataset and reused across all configurations of that seed, so every configuration sees identical tasks under a given seed (paired comparison). Datasets: GSM8K test split (1,319 problems), MBPP (500 tasks), MMLU (312 questions from abstract algebra, college mathematics, machine learning). The ablation/baseline deployment uses 30 tasks per seed over seeds $\{42, 43, 44\}$ (n=90 per cell), and the adversarial-boundary scaling study (Table 9) uses 150 GSM8K tasks per configuration via the DeepSeek API.

A.3.4 BASELINE IMPLEMENTATIONS

All four baselines consume the same heterogeneous four-model proposals as A2A-BFT:

- **Simple Majority:** commit the plurality answer across replicas; no validation stage, no thresholds, no view change.
- **Weighted Majority:** plurality vote weighted by each replica's capability score; otherwise identical to Simple Majority.
- **A2A-Sim-style:** adapts the aggregation of Berdoz et al. (2026) to task answering—iterative answer proposals terminated by a supermajority-style agreement check—augmented with a 0.7 confidence filter on countable proposals; no validation stage, no view change, no Byzantine-aware thresholds.
- **LLM-Debate** (Du et al., 2024): two debate rounds in which honest agents revise their answer after seeing the others', followed by a majority commit; the tabulated runs use *no* execution oracle. The oracle-assisted variant—which retains any solution already passing all tests—is reported separately below.

On self-refinement and multi-agent framework baselines. Reflexion (Shinn et al., 2023) is a single-agent self-refinement loop and MetaGPT (Hong et al., 2024) a role-based software pipeline; neither defines a quorum, a fault model, or a commit rule, so neither can be placed on the wrong-commit axis without first supplying the consensus layer that is the object of study. We therefore

compare against the closest aggregation-based alternatives and, to avoid under-testing the debate baseline, also run an *oracle-assisted* variant that rewrites each agent's solution but keeps any solution that already passes all tests: on MBPP it still commits incorrect answers in 46.7% (strategic rejection) and 56.7% (collusion) of tasks, versus 26.7% fault-free (`mbpp_debate_fix.json`). Giving the debate baseline a verification oracle therefore does not rescue an unvalidated aggregation rule, which is precisely the gap A2A-BFT closes.

A.4 Implementation Details

Table 2: Message Complexity Comparison. *Measured mean rounds per task (Table 11); PBFT's 3 counts its prepare/commit phases.

Protocol	Message Complexity	Rounds	Fault Tolerance
PBFT	$O(n^2)$	3	$f < n/3$
A2A-BFT	$O(n^2)$	2.6–4.5*	$3f + s \leq n - 1$
A2A-Sim (Baseline)	$O(n^2)$	Variable	No guarantee

A.4.1 System Architecture

Figure 2 shows the A2A-BFT system architecture with four key components:

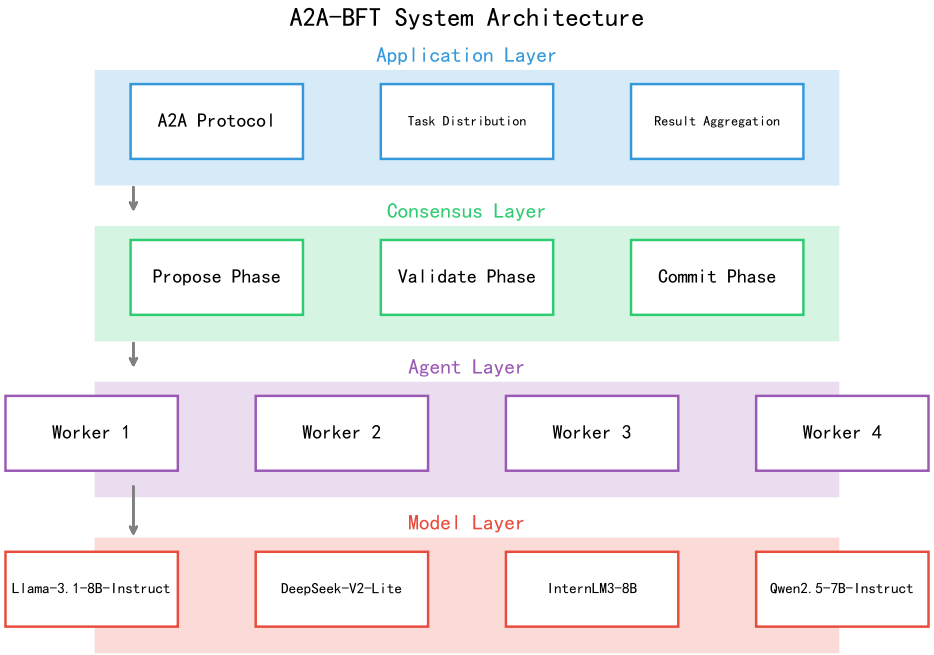

Figure 2: A2A-BFT System Architecture

Agent Layer: Each agent implements the DistributedWorker interface with:

- Task solving capability (LLM inference)
- Proposal generation
- Independent verification
- Vote signing and broadcasting

Consensus Layer: Manages the three-phase protocol, threshold computation, and reputation tracking.

Model Validator: Automatically tests model quality across mathematical reasoning, logic, translation, and knowledge tasks before deployment.

Network Layer: Handles A2A protocol communication with message authentication and ordering.

A.4.2 Model Integration

A2A-BFT supports multiple LLM backends deployed on GPU infrastructure:

- **Llama-3.1-8B-Instruct** (Meta AI, 2024): General-purpose reasoning
- **DeepSeek-V2-Lite** (DeepSeek-AI, 2024): Cost-effective alternative
- **InternLM3-8B** (InternLM Team, 2025): Strong coding capability
- **Qwen2.5-7B-Instruct** (Qwen Team, 2024): General-purpose reasoning, strong code capability

Each model undergoes validation testing before deployment, ensuring minimum quality thresholds.

A.4.3 Reputation Implementation

The reputation tracker maintains:

- Per-agent reputation scores (initialized to 1.0)
- Voting history (last 10 rounds) for the reject ratio ρ_i
- Suspect counters for anomaly detection. The released code *does* escalate: once a counter reaches its threshold the score is *set* to $\max(0.3,\ r_i - 0.2)$, i.e. a further clamped decrement of 0.2 applied to the post-tier value (an assignment, not an additive term). A step-by-step fidelity audit against the released penalty code ($12{,}000$ update steps over all three vote types, zero divergence) confirms that the escalation fires. There is no exclusion or reweighting.
- A correctness predicate c for vote scoring. Algorithm 1 requires "is the proposal correct"; the released `_estimate_correctness` approximates this with a format check (a final-answer marker on GSM8K, a Python code block on MBPP), evaluated on the *full* model response rather than a prefix—truncating the response would flip the predicate on answers whose marker appears late. Section 6.6 measures that this predicate is a per-domain constant in practice across all 457 logged proposal rounds.

A.5 Real LLM Validation Details

We conducted comprehensive experiments using the DeepSeek-V2-Lite API to validate A2A-BFT with real LLM outputs. This evaluation uses a *homogeneous* single-model deployment, which upper-bounds validation agreement relative to the heterogeneous sweep; it covers 320 tasks across three domains: mathematical reasoning (GSM8K), knowledge QA (MMLU), and code generation (MBPP).

Table 3: Real LLM Experiment Results (DeepSeek-V2-Lite; GSM8K, MMLU, MBPP; 320 tasks, of which 291 completed—acceptance computed over completed tasks; see the note below)

Configuration	Tasks	Accept%	Avg Time (s)
Math baseline ($n = 4, f = 0, s = 0$)	40	100	6.30
Math BFT ($n = 5, f = 1, s = 1$)	40	100	9.22
Math + Strategic Reject ($n = 5, f = 1, s = 1$)	40	100	8.90
Math + Collusion ($n = 6, f = 2, s = 0$)	40	100	8.49
Knowledge baseline ($n = 4, f = 0, s = 0$)	40	100	27.00
Knowledge BFT ($n = 5, f = 1, s = 1$)	40	100	28.13
Code baseline ($n = 4, f = 0, s = 0$)	40 (12)*	100	38.16
Code BFT ($n = 5, f = 1, s = 1$)	40 (39)*	100	9.01
Total	**320**	**100**	–

Key Findings:

1. A2A-BFT achieves high acceptance rate across all task types with real LLM outputs, confirming cross-domain portability of the consensus protocol.
2. Byzantine attacks (strategic rejection, collusion) are resisted without impacting honest agent consensus.
3. Inference latency is dominated by LLM generation and varies by task type: math ($\sim$6.3–9.2s), knowledge ($\sim$27–28s), code ($\sim$9–38s).
4. No proposal was rejected by the protocol in any configuration; every non-acceptance is an API/harness failure (see the note below).

Note on Table 3: ⋆The "Tasks" column shows submitted (and, in parentheses, completed) tasks: 28 code-baseline tasks and 1 code-BFT task terminated in API/harness errors, and 100% is computed over the completed ones. All 320 tasks were submitted to the live DeepSeek API, and acceptance is computed over the 291 that completed execution—every one was accepted. The other 29 terminated in API/harness errors (28 in the code-baseline row, whose wall-clock is inflated to 38.16s; 1 in the code-BFT row) and are excluded; all 29 are generation-layer transport failures, uncorrelated with the attack configuration (they occur in fault-free and attacked rows alike) and cannot affect the consensus verdicts, which are computed over completed tasks only. No proposal was *rejected* in any configuration (`reject_count` $= 0$), so the 100% column reflects consensus, not filtering, and every latency is a genuine end-to-end measurement—the sole caveat being the code-baseline average noted above, which those 28 terminations inflate. This table is a homogeneous-deployment upper bound (Section 6.4); the authoritative adversarial-boundary results—real Byzantine tampering, numerical verification, measured rounds—are in Table 9.

Measuring Answer Correctness. A consensus protocol can reach agreement on an answer that is nonetheless wrong. To validate the *validity* (not merely the *agreement*) guarantee of A2A-BFT, we additionally measure answer correctness: for each accepted proposal we extract the final numerical answer and compare it against the GSM8K ground truth. This distinguishes a protocol that merely terminates from one that terminates on the *correct* answer. The results (Table 9) show that under Byzantine attacks the consensus rate remains at or near 100% (98–100% across two independent runs), while answer correctness stays within 2 percentage points of the fault-free baseline (92.0–92.7% vs. 94.0%). The decisive evidence of genuine resistance is the round-count gap (1.03 vs. 2.89–2.92 rounds; Welch $t \leq -13.1$, $p < 10^{-26}$): Byzantine primaries tamper with answers, honest validators reject them, and view change rotates to an honest primary before a correct answer is accepted. This gap is reproduced in a repeat run (1.04 vs. 2.38–2.43 rounds), whereas the consensus rate there is 100%; the paired per-task difference between the two runs is not significant (McNemar exact $p = 0.25$ on each attacked configuration, computed over the identical task sets), so we report the more conservative run and treat the residual decision-rate variation as API run-to-run variance.

The full performance trade-off analysis and statistical significance tests, the complete attack-resistance breakdown, and the direct comparison with A2A-Sim under identical conditions are provided in Appendix A.7.

A.6 Judge Error-Rate Calibration from Released Vote Streams

The safety guarantee of Corollary 5.2 is conditional on the validation oracle's miss probability p_{mr}. We measure it directly from the released per-round vote streams (Section 6.6): every round records each validator's verdict together with the ground-truth correctness of the proposal under vote (241 of 254 GSM8K rounds carry ground truth; 13 empty-proposal rounds are excluded). Over all honest-validator votes on GSM8K—Byzantine and soft-fault voters excluded—the semantic judge accepts a wrong proposal with rate 18.1% ($127/700$; 95% Wilson CI $[15.5, 21.2]$) and rejects a correct proposal with rate 19.7% ($53/269$; $[15.4, 24.9]$); the measured judge accuracy on correct proposals is thus ≈ 0.80 rather than the idealized $p_h \geq 0.9$. Table 4 breaks this down by scenario.

Three observations. **First**, the measured false-accept rates are of the same order as the observed within-boundary wrong-commit rates under attack (11.6–23.2%, Table 1), closing the loop between the oracle-error account of Corollary 5.2 and the end-to-end measurements. **Second**, at the measured $p_h = 0.80$ the single-round acceptance bound is ≥ 0.64 for $n-1-f-s = 2$, so the asymptotic liveness guarantee (Theorem 5.3) is unaffected—only the geometric rate is slower than the $p_h = 0.9$ illustration. **Third**, these are *on-protocol* measurements: proposals arise inside the consensus loop

Table 4: Measured honest-validator semantic-judge error rates on GSM8K, from the released per-round vote streams (honest voters only; empty-proposal rounds excluded). CI: Wilson 95%.

	False accept (on wrong)		False reject (on correct)	
Scenario	rate (%)	CI	rate (%)	CI
Fault-free (n=5)	23.0	[15.8, 32.2]	27.6	[21.2, 35.1]
Strategic reject (n=5, f=1, s=1)	25.2	[18.9, 32.6]	10.2	[4.4, 21.8]
Collusion (n=8, f=2, s=1)	14.7	[11.7, 18.3]	7.8	[3.4, 17.0]
Overall	**18.1**	[15.5, 21.2]	**19.7**	[15.4, 24.9]

rather than from an independent held-out set. The held-out calibration below removes this caveat; MBPP is excluded from both because its execution validation is mechanically correct, and per-round MMLU vote streams were not logged (its higher wrong-commit rates, Table 10, are consistent with a weaker semantic judge on that domain).

Held-out calibration. We additionally calibrate every judge on an independent held-out sample: 150 GSM8K and 124 MMLU tasks whose indices are excluded from every task set used in this paper (the full-sweep, ablation, reputation, and multiseed samplings), each presented to each judge model under two conditions—the ground-truth answer (measuring FNR) and a deterministic plausible perturbation, a numeric delta on GSM8K and a wrong choice letter on MMLU (measuring FPR)—using the protocol's judge prompts and verdict parsing verbatim, on the same four vLLM endpoints (2,192 calls; per-call records released as `heldout_judge_calibration.json`). Table 5 reports per-judge rates. Three findings. *First*, FPR is dominated by a single judge: DeepSeek-V2-Lite accepts wrong answers at 73.3% (95% CI [65.3, 80.1]) on GSM8K and 65.3% on MMLU, whereas the other three judges reject almost all wrong GSM8K answers ($\leq$ 1.4%); decomposing the on-protocol vote streams by judge reproduces the same pattern (DeepSeek FPR 78.1% on wrong proposals, the other judges $\leq$ 1.9%), so the 18.1% aggregate of Table 4 is a mixture of three strict judges and one permissive judge, not a uniform oracle error. *Second*, held-out FNR (5.1–15.2% on GSM8K, 3.1–32.7% on MMLU) is consistent with the on-protocol aggregate (19.7%). *Third*, InternLM3 often fails to emit a parseable verdict (32% of GSM8K and 73% of MMLU calls; other judges $\leq$ 25%), which the protocol maps to ABSTAIN—an abstention, not an accept—so judge weakness degrades liveness, not safety. The implication is that deployed safety rests on the commit rule aggregating across heterogeneous judges: a validator pool dominated by permissive judges would inherit their FPR, so validator selection belongs in deployment checklists (Appendix A.12).

Table 5: Held-out per-judge calibration (rate in %, Wilson 95% CI in brackets). Unparsed verdicts map to ABSTAIN and are excluded from the rates; InternLM3 abstains on 32% (GSM8K) and 73% (MMLU) of calls, other judges $\leq$ 25%.

	GSM8K		MMLU	
Judge	FPR	FNR	FPR	FNR
Llama-3.1-8B	1.4 [0.4, 5.0]	15.2 [10.2, 22.1]	37.6 [28.8, 47.4]	32.7 [24.3, 42.3]
InternLM3-8B	0.0 [0.0, 3.8]	7.5 [3.9, 14.2]	79.4 [63.2, 89.7]	3.1 [0.6, 15.7]
DeepSeek-V2-Lite	**73.3** [65.3, 80.1]	8.6 [5.0, 14.5]	**65.3** [55.5, 74.0]	31.3 [22.9, 41.1]
Qwen2.5-7B	1.4 [0.4, 5.0]	5.1 [2.5, 10.2]	24.5 [16.9, 34.0]	22.3 [15.1, 31.8]

A.7 Additional Experimental Analysis

Tables 6, 7, and 8 report the fault-free baseline grid and the full ablation and baseline-comparison grids referenced in Section 6.

A.7.1 Adversarial-Boundary Scaling (n=8, Real DeepSeek API)

These homogeneous-API runs are an upper bound on decision rates (Section 6.4); Table 9 details answer correctness at the boundary configuration.

Table 6: Baseline Performance Without Faults (real heterogeneous 4-model deployment; cells: decision% / wrong-commit%, mean±std over sampling seeds). n=250 tasks per cell (5 seeds × 50 tasks) except n=5 rows (n=90; 3 seeds × 30 tasks, from the ablation study deployment).

Configuration	GSM8K (semantic)		MBPP (execution)	
	decision%	wrong-commit%	decision%	wrong-commit%
$n = 4, f = 0, s = 0$	59.2 ± 9.4	1.2 ± 1.1	73.2 ± 6.1	$\mathbf{0.0 \pm 0.0}$
$n = 5, f = 0, s = 0$	56.7 ± 5.8	1.1 ± 1.9	70.0 ± 11.5	$\mathbf{0.0 \pm 0.0}$
$n = 6, f = 0, s = 0$	55.2 ± 7.6	1.2 ± 1.1	73.6 ± 6.5	$\mathbf{0.0 \pm 0.0}$

Table 7: Ablation Study under Byzantine Attacks (Heterogeneous 4-Model vLLM; mean±std over three sampling seeds, 30 tasks each, n=90 per cell; cells: decision% / wrong-commit%). Bold marks the *highest* wrong-commit rate per column; columns where all variants tie at 0.0 are left unmarked.

Variant	GSM8K (semantic)		MBPP (execution)	
	n=5, f=1, s=1	n=8, f=2, s=1	n=5, f=1, s=1	n=8, f=2, s=1
Full protocol	74.4±8.4/**4.5±3.9**	61.1±10.7/10.0±3.3	70.0±11.5/0.0±0.0	60.0±8.8/0.0±0.0
No View Change	4.5±3.9/**4.5±3.9**	12.2±5.1/**12.2±5.1**	0.0±0.0/0.0±0.0	0.0±0.0/0.0±0.0
No Sem. Validation	54.4±8.4/2.2±1.9	44.5±10.7/2.2±1.9	1.1±1.9/0.0±0.0	0.0±0.0/0.0±0.0
Fixed Threshold	0.0±0.0/0.0±0.0	38.9±7.7/0.0±0.0	0.0±0.0/0.0±0.0	51.1±13.4/0.0±0.0

Table 8: Baseline Comparison under Byzantine Attacks (Heterogeneous 4-Model vLLM; mean over three sampling seeds, n=90 per cell). Cells report decision% / answer-correctness% / wrong-commit%; bold marks the *highest* wrong-commit rate per column (the boldface method name identifies our row); seed-level std is at most 16.8 points (0–13.5 for wrong-commit), full mean±std in released data. LLM-Debate uses two debate rounds with a majority commit and no execution oracle; an oracle-assisted variant is reported in Appendix A.3.

Method	GSM8K		MBPP	
	Strat. Reject	Collusion	Strat. Reject	Collusion
A2A-BFT (ours)	74.4/70.0/4.5	61.1/51.1/10.0	70.0/70.0/0.0	60.0/60.0/0.0
Simple Majority	100/71.1/**28.9**	100/66.7/**33.3**	100/43.3/**56.7**	100/38.9/61.1
Weighted Majority	100/71.1/**28.9**	100/68.9/31.1	100/51.1/48.9	100/45.6/54.4
A2A-Sim	62.2/54.4/7.8	66.7/58.9/7.8	82.2/46.7/35.5	83.3/48.9/34.4
LLM-Debate	100/82.2/17.8	100/83.3/16.7	100/43.3/**56.7**	100/36.6/**63.4**

Table 9: Answer Correctness at the Adversarial Boundary n=8 (Real DeepSeek API, GSM8K, 150 tasks per configuration). Raw per-task logs: `experiments/results/correctness_50x3_log.txt`. A repeat run gave 100.0/93.3/2.43, 100.0/94.0/2.38, 100.0/94.0/1.04 for the three rows; the round-count gap is reproduced, the decision-rate delta is within API run-to-run variance.

Configuration	Consensus%	Correctness%	Rounds
n=8, f=2, s=1 + Collusion	98.0	92.7	2.89
n=8, f=2, s=1 + Strategic Reject	98.0	92.0	2.92
n=8, f=0, s=0 (baseline)	100.0	94.0	1.03

A.7.2 MMLU (KNOWLEDGE QA) FAULT SWEEP

Table 10 completes the fault sweep of Table 1 with the knowledge-QA domain (MMLU, three core subjects; n=250 per cell).

Table 10: MMLU Fault Sweep (real heterogeneous 4-model deployment; cells: decision% / wrong-commit%, mean±std over 5 seeds, 50 tasks each). †: below the safety boundary.

n	f	s	Attack	decision%	wrong-commit%
4	0	0	baseline	14.8 ± 5.6	3.6 ± 2.6
5	0	0	baseline	14.8 ± 5.9	4.4 ± 2.6
6	0	0	baseline	11.2 ± 5.6	2.8 ± 1.8
4	1	0	Random	65.2 ± 2.7	35.6 ± 3.6
4	1	0	Strat. Reject	73.2 ± 7.2	60.8 ± 8.6
4	1	0	Sybil	68.4 ± 6.2	39.6 ± 6.2
5	1	1	Random	75.2 ± 1.8	40.8 ± 5.0
5	2	0	Random†	88.8 ± 4.8	55.6 ± 6.1
5	2	0	Strat. Reject†	94.8 ± 2.7	85.2 ± 7.9
5	2	0	Collusion†	96.0 ± 2.0	90.0 ± 3.7
6	2	0	Collusion†	85.2 ± 4.6	74.4 ± 7.0

MMLU is the most demanding domain and the results are correspondingly stark. *Fault-free*, only 11–15% of tasks are decided: the subjects are hard enough that heterogeneous 8B replicas rarely agree, and the protocol correctly withholds commitment rather than gamble on an unvalidated majority. *Under attack*, decision rates rise to 65–96% (Byzantine primaries inject a stream of proposals; eventual acceptance commits whichever clears validation) and wrong commits reach 56–90% below the boundary. This domain illustrates the oracle-composition principle most sharply: consensus safety cannot exceed validation reliability, and on hard knowledge questions 8B semantic judges are unreliable validators. The practical implication is architectural rather than negative: for knowledge-QA deployment, the Validate stage should be backed by retrieval or stronger verifier models, and the protocol's contribution is to make this requirement *measurable* (wrong-commit rate) rather than invisible.

A.7.3 PERFORMANCE ANALYSIS

Table 11 reports the real end-to-end cost of the protocol on the four-model deployment (measured wall-clock per task, including all LLM inference, validation, and view changes).

Table 11: Real End-to-End Cost per Task (heterogeneous 4-model vLLM, n=250 tasks per cell). "Calls" counts LLM requests per task; MBPP validation executes code locally instead of invoking the judge, hence its far lower call count. GSM8K CoT generation dominates the latency.

Configuration	Domain	Time/Task	Calls/Task	Rounds
$n = 4, f = 0, s = 0$	GSM8K	61.3s	12.7	3.20
	MBPP	11.9s	2.6	2.60
$n = 4, f = 1, s = 0$ Strat. Reject	GSM8K	88.0s	13.4	3.90
	MBPP	14.8s	3.4	3.40
$n = 5, f = 2, s = 0$ Collusion†	GSM8K	72.1s	11.0	2.90
	MBPP	19.3s	4.5	4.50
$n = 6, f = 0, s = 0$	GSM8K	88.4s	20.0	3.30
	MBPP	15.4s	2.6	2.60

Cost structure: latency is dominated by LLM generation, and it tracks the number of generations: the n=4 → n=6 increase (61.3 → 88.4s on GSM8K) accompanies 12.7 → 20.0 LLM calls per task, a linear growth in the validator count rather than the $O(n^2)$ message complexity. Attacks raise latency only moderately within the boundary (61.3 → 88.0s under strategic rejection); the sub-boundary collusion row is *faster* (2.90 vs. 3.20 rounds) because colluding validators accept immediately—precisely why it is unsafe.

Figure 3 shows the performance trade-offs across configurations.

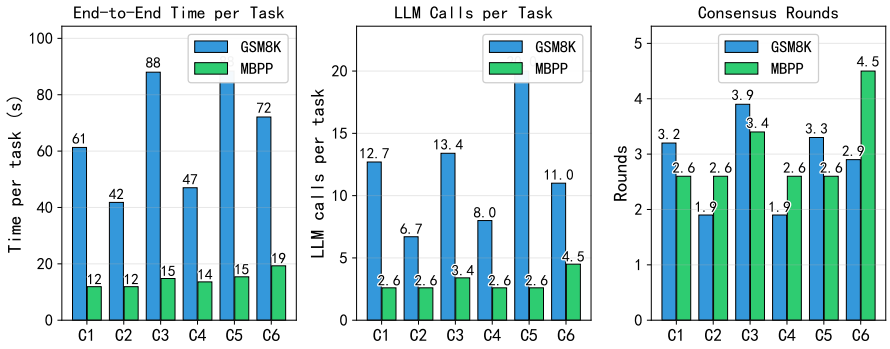

Figure 3: Real End-to-End Cost per Task (heterogeneous 4-model deployment): wall-clock time, LLM calls, and consensus rounds across six configurations—C1: n=4 baseline; C2: n=4, f=1 random; C3: n=4, f=1 strategic reject; C4: n=5, f=1, s=1 random; C5: n=6 baseline; C6: n=5, f=2 collusion (below the boundary). GSM8K CoT generation dominates latency; MBPP execution validation requires far fewer LLM calls.

A.8 Reputation Tracker: Full Measurement Details

This appendix expands Section 6.6 with the full measured table, the per-finding numbers, and the complete repair analysis.

The reputation tracker is excluded from the vote-counting rule (Equation 2; Section 6.6), so a reputation-on/off comparison of decision metrics would return an identity by Theorem 5.1; we instead measure *what the tracker reports*. We instrumented the protocol to log every validation vote from the same four-model deployment (single sampling seed; 20 tasks per cell over the six full-protocol scenarios underlying Table 12; 120 consensus runs, 457 scored proposal rounds, and 2,461 scored validation events, where one *event* is one reputation update triggered by one validation vote) and replayed the recorded vote streams offline through Algorithm 1 under four settings: c taken from ground truth (*oracle*; the definition Algorithm 1 requires) or from the released `_estimate_correctness` heuristic (*released*); and reputation persisting across tasks, as in the reference implementation where one consensus layer lives for the whole run, or reset per task. Replay issues no LLM calls and is therefore exactly reproducible; in *all four* settings the replayed decision sequence is identical to the online record, which is both a correctness check and direct evidence that no reported decision depends on the tracker. Table 12 reports the oracle/persistent setting.

Table 12: Measured reputation-tracker behavior (oracle c, persistent reputation; single seed, 20 tasks per cell). *Aggr.* = aggressive-tier events ($\rho_i \geq 0.7$, -0.25); one event is one reputation update triggered by one validation vote, so events are *not* independent observations. *Mis-pen.* = aggressive penalties landing on an honest validator whose vote was *objectively correct*—the case Section 4.4 specifies should be rewarded. Pre./Rec. treat an aggressive penalty as a Byzantine detector. *Escal.* $\rightarrow$ *hon.* = fraction of suspect-counter escalations (-0.2) landing on honest agents. r = mean end-of-task reputation, averaged over all (task, validator) snapshots in the cell; "—" = the cell contains no Byzantine validator.

Scenario	Aggr.	Mis-pen.	Rate	Pre.	Rec.	Escal. $\rightarrow$ hon.	r hon./Byz.
GSM8K f=0 (no attack)	66	42	0.636	—	—	1.000	0.587 / —
GSM8K f=1 strat. rej.	164	116	0.707	0.232	1.000	0.832	0.436 / 0.300
GSM8K f=2 collusion	373	355	**0.952**	**0.000**	**0.000**	0.856	**0.337 / 0.427**
MBPP f=0 (no attack)	88	88	1.000	—	—	1.000	0.573 / —
MBPP f=1 strat. rej.	200	155	0.775	0.195	1.000	0.843	0.338 / 0.332
MBPP f=2 collusion	568	560	**0.986**	**0.000**	**0.000**	0.888	**0.300 / 0.412**

Four findings, from descriptive single-seed measurements (20 tasks per cell; see Limitations). **(i) The aggressive tier fires almost exclusively on honest, correct votes.** Across the four cells containing Byzantine validators, 1,186 of 1,305 aggressive penalties landed on an honest validator whose vote was objectively correct—exactly the behavior that Section 4.4 specifies should be *re-*

warded. Reputation updates are clustered within validators and tasks, so we do not attach a binomial interval to that ratio; the task-level statistic is unambiguous: *all 20 tasks* in each of the four attacked cells produced at least one mis-penalty. The mechanism is structural: honest validators correctly reject the wrong proposals that Byzantine primaries inject, so their own rejection ratio ρ_i climbs past 0.7 and the ρ-keyed tier overrides the reward branch. In the two cells with *no* Byzantine validator at all the tier still fires (66 and 88 events, in 10 and 9 of 20 tasks respectively, of which 8 and 9 produced a mis-penalty—a penalty on an honest, correct vote in a cell where no adversary exists), because disagreeing with a wrong proposal is itself what raises ρ_i. **(ii) As a Byzantine detector the tier fails.** Treating an aggressive penalty as a Byzantine flag yields recall 1.000 but precision $0.232/0.195$ under strategic rejection, and *precision and recall both* 0.000 under collusion. The logged ρ_i makes the mechanism explicit: a colluding validator rejects only in the round where an honest primary's proposal is finally committed, so its mean rejection ratio stays at 0.358 (GSM8K) and 0.392 (MBPP) and never reaches the 0.7 trigger in a single validator-task pair, whereas honest validators—who must reject the wrong proposals those same primaries inject—average 0.600 and 0.851, crossing 0.7 in 61.7% and 90.0% of pairs. Under strategic rejection, by contrast, Byzantine validators reject unconditionally ($\bar{\rho} = 1.000$, 1.9 aggressive hits per task) and are caught, which is why recall is 1.000 there. **(iii) The score ordering inverts.** In both collusion cells honest validators finish *below* Byzantine ones (0.337 vs. 0.427 on GSM8K; 0.300 vs. 0.412 on MBPP; Figure 4, right), the opposite of the accountability ordering the tracker is meant to produce. Resetting reputation per task does not repair this (honest 0.391 vs. Byzantine 0.761 on GSM8K collusion), so the failure is structural rather than a cold-start artifact. **(iv) The released c carries negative information.** Algorithm 1 requires c = "is the proposal correct". The released `_estimate_correctness` is a format check that returns `False` for all 241 GSM8K rounds and `True` for all 216 MBPP rounds—a per-domain constant. It is not merely uninformative but counter-productive: the base rate of correct proposals across all 457 rounds is 25.4%, so the trivial always-`False` predictor would agree with ground truth 74.6% of the time, whereas the released predicate—constant within each domain—agrees on only 46.2%. On GSM8K it happens to pick the better of the two constants (69.7% vs. 30.3% for always-`True`); on MBPP it picks the worse one (19.9% vs. 80.1%). Overall it is worse than the cheapest possible baseline. In the released configuration the reward and mild-penalty tiers are therefore driven by vote *direction* alone, and the promised reward for correctly rejecting a wrong proposal does not occur.

Two conclusions follow. First, the tracker is genuinely decision-neutral: recomputing ϕ from the raw votes reproduces all 457 recorded rounds and replaying the vote-and-threshold state machine reproduces all 120 online decisions together with the primary rotation, whereas a mutation that injects reputation weights shifts ϕ in up to 421 rounds and flips up to 5 decisions—a check that can fail rather than a tautology. Second, as an incentive mechanism the released tracker is misspecified: it taxes precisely the behavior it is meant to encourage. Because no reported safety or liveness *number* depends on it, the quantitative results stand; but the deterrent that Appendix A.9 lists against strategic rejection is not delivered by the released code, so that entry should be read as specified rather than realized. This failure mode is itself known to the trust literature—reputation rules keyed on a rater's own behavior punish honest negative feedback (Jøsang & Ismail, 2002; Jøsang et al., 2007), and P2P designs such as EigenTrust (Kamvar et al., 2003) inherit the same exposure. What we add is the scale of the effect in a real LLM consensus deployment ($1{,}186/1{,}305$ mis-attributed penalties; $20/20$ affected tasks in every attacked cell) and evidence that it is structural rather than a tuning artifact.

Repairing it requires (a) a correctness oracle for c that measures the *proposal* rather than its surface form, and (b) a trigger that is a function of the proposal's validated quality rather than of the validator's own rejection ratio, which honest disagreement inflates. We deliberately do *not* suggest keying the trigger on agreement with the committee's majority: that would reintroduce exactly the coupling to the committee's decision that Algorithm 1 is designed to avoid, and under attack it would penalise precisely the lone correct dissenters the mechanism exists to protect. Triggers compatible with this constraint include execution-validated outcomes on verifiable domains and DID-signed cross-round consistency on semantic ones. We leave the repaired design to future work; the instrumented vote streams and the four-setting replay are released for independent verification.

Figure 4 depicts the reputation update rule (Algorithm 1) alongside *measured* per-agent reputation trajectories from the real four-model deployment (Section 6.6); the trajectories are replay-derived

from the instrumented vote streams (Algorithm 1 re-executed offline on the logged votes), so the right panel is measured rather than schematic.

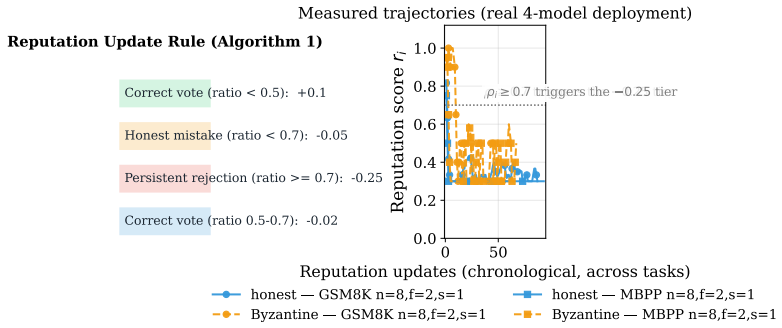

Figure 4: Reputation update rule (Algorithm 1; left) and *measured* per-agent reputation trajectories under collusion in the real four-model deployment (right; replay-derived by re-executing Algorithm 1 offline on the logged vote streams; 20 tasks per cell, oracle c, reputation persisting across tasks; Section 6.6). In both collusion cells the honest curves finish *below* the Byzantine ones.

A.9 Attack Resistance Analysis

A2A-BFT's behavior under all tested attack vectors separates cleanly along the safety boundary:

Table 13: Attack Resistance Summary (real 4-model deployment; ranges over configurations of Table 1, n=250 per cell, plus two n=90 cells: the within-boundary collusion configuration n=8, f=2, s=1 from Table 8 and the strategic-reject row n=5, f=1, s=1 marked ‡ in Table 1). "Boundary" = configurations satisfying $n \geq 3f+s+1$; "below" = n=5, f=2 and n=6, f=2 for f=2.

		Within boundary		Below boundary†	
Attack	**Type**	**decision%**	**wrong%**	**decision%**	**wrong%**
Strategic Reject	Malicious	70.0–80.8	0.0–23.2	74.4–95.2	9.6–67.6
Collusion	Malicious	60.0–61.1	0.0–10.0	66.4–98.4	0.0–72.4
Sybil Attack	Malicious	73.6–90.4	0.0–12.8	–	–
Random Fault	Soft	73.2–90.8	0.0–14.0	78.8–96.8	5.6–22.0

Note: Wrong-commit ranges span the two validation semantics: the lower end is always MBPP (execution validation, mechanically 0% within the boundary) and the upper end is GSM8K (8B semantic judges). Within the boundary the protocol committed no wrong answer on execution-validated tasks; on semantic tasks the wrong-commit rate is bounded by the judge's susceptibility to plausible-but-wrong proposals. Below the boundary (f=2, n=5, 6, where $3f+s+1 = 7 > n$) safety is violated on both domains (up to 9.6% on execution-validated code and 72.4% on GSM8K)—consistent with $3f+s+1$ being a tight operating condition in the evaluated regime $s \leq f$.

- **Strategic Rejection:** Specified to be penalized by the reputation tracker (Algorithm 1; Figure 4); the four-model sweep executes with uniform weights. Section 6.6 measures the released tracker as a Byzantine detector at precision 0.195–0.232 under strategic rejection (recall 1.000) but 0.000 under collusion, with the bulk of aggressive penalties landing on honest validators whose vote was objectively correct (Section 6.6), so the deterrent is best read as unrealized rather than merely unmeasured.
- **Collusion:** Within the boundary, collusion inflates wrong commits on semantic tasks exactly as the threshold analysis permits (Property 4; e.g., $17 \geq 15$ at n=5, f=2); on execution-validated tasks wrong commits remain 0% because the commit rule requires passing the test suite. Below the boundary, collusion produces the largest violations in the sweep (72.4% on GSM8K). The thresholds bound *when* safety can break, not *whether* the validation oracle catches the injected proposals.

- **Sybil Attack:** DID-based identity authentication prevents identity replication, so an attacker cannot inflate its voting power by forging replicas. Because Equation 2 counts raw votes with uniform weights and reputation is excluded from the commit rule, REJECT-heavy or abstention-heavy behavior yields no additional weight; a Sybil set is therefore limited to the modeled f Byzantine seats.

Figure 5 visualizes attack resistance across configurations.

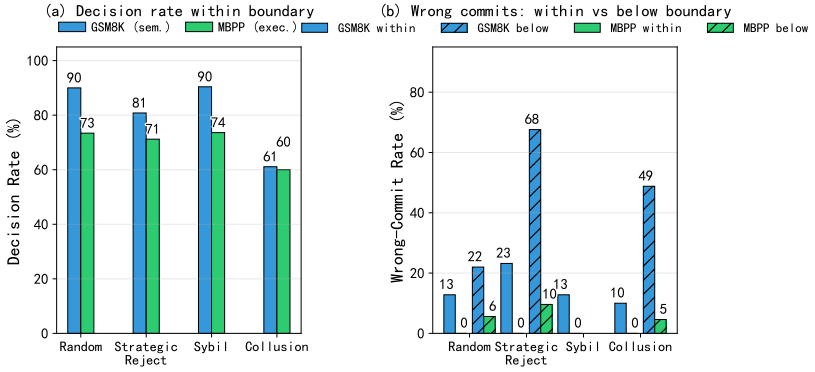

Figure 5: Attack Resistance (real 4-model deployment). Bars are from the n=250 four-model sweep, except the within-boundary collusion bars (n=8, f=2, s=1), which come from the n=90 ablation/comparison deployment (cf. Table 13); where an attack spans multiple configurations of a table, the bar shows the mean over them. (a) Decision rates within the safety boundary. (b) Wrong-commit rates: hatched bars are sub-boundary configurations, where safety violations emerge exactly as the theorem permits.

A.10 Comparison with A2A-Sim

A2A-Sim (Berdoz et al., 2026) provides a comprehensive evaluation of LLM agent consensus without BFT guarantees. Table 14 presents a direct comparison. **Note:** Only the benign consensus rate (41.6%) is quoted directly from the original publication (Berdoz et al., 2026); the remaining rows are computed from our heterogeneous four-model evaluation (Table 8; 3 sampling seeds, n=90 per cell), in which an A2A-Sim-style unvalidated scalar-consensus baseline is compared under identical conditions. These rows are *not* directly comparable to the original study's "invalid consensus" rate: the original measures agreement on a value outside the valid set in a scalar no-stake game and finds most failures to be liveness losses, whereas we measure committed answers against GSM8K/MBPP ground truth under an answer-tampering Byzantine primary.

Table 14: A2A-BFT vs A2A-Sim. The benign rate is quoted from the original publication; all remaining cells (incl. the fault-free A2A-BFT row) use the heterogeneous four-model evaluation (3 seeds, n=90/cell; strategic rejection, GSM8K/MBPP), where $\pm$ is the seed-level std of the replication run. †Different metric and source—agreement rate in a scalar no-stake game, quoted from Berdoz et al. (2026)—so no Δ is computed for this row.

Metric	A2A-Sim	A2A-BFT	Δ
Acceptance decision (no faults)	41.6%†	56.7±5.8 / 70.0±11.5%	—
Decision rate (f=1, strat. reject)	62.2% ± 5.1%	74.4% ± 8.4%	+12.2 pp
Wrong-commit (f=1 attacks)	7.8–35.5%	0.0–4.5%	−3.3 to −35.5 pp

Key Differences:

- A2A-Sim is a measurement framework rather than a protocol, so its 41.6% benign rate is not a protocol deficiency but evidence that LLM agents struggle to converge on a shared value even with no faults present—the same phenomenon that caps our own fault-free GSM8K decision rate at 55–59%. What it does not provide is any guarantee once faults *are* present, which is the gap A2A-BFT addresses.

- With one Byzantine replica under strategic rejection, A2A-Sim decides fewer tasks (62.2% vs. 74.4%) and commits wrong answers in up to 35.5% of code tasks, whereas A2A-BFT commits no wrong answer on execution-validated code (0.0%).
- The collusion cell in Table 8 is the one place where the comparison does not favour us: at n=8, f=2, s=1 on GSM8K, A2A-Sim's decision rate is marginally higher (66.7% vs. 61.1%) and its wrong-commit rate marginally lower (7.8% vs. 10.0%). Both differences are well within the seed-level variability observed in comparable cells (e.g., 7.8 ± 1.9 vs. 4.5 ± 3.9 for wrong-commit under strategic rejection; descriptive at three seeds), and neither survives pooling: what separates the methods is the code domain (34.4–35.5% vs. 0.0% wrong commits) and the fact that A2A-BFT's decisions carry a proven guarantee rather than a confidence threshold. Under collusion the protocol is deliberately conservative—it forgoes a decision when validation is inconclusive—which costs decision rate in exactly the cell where an unvalidated aggregator happens to agree with itself.

Design Philosophy: While A2A-Sim focuses on evaluating LLM agent communication patterns, A2A-BFT addresses the lack of Byzantine fault tolerance in A2A networks. The two approaches are complementary: A2A-Sim provides insights into LLM consensus behavior, while A2A-BFT provides the theoretical guarantees necessary for production deployment.

A.11 Rational Byzantine Analysis

We extend the analysis to rational (rather than malicious) Byzantine agents who optimize utility rather than cause harm:

Definition A.1 (Rational Byzantine Agent)**.** *A rational Byzantine agent maximizes expected utility $U = B - C$, where B is benefit from deviation and C is cost of reputation penalty.*

Game-Theoretic Analysis: Under repeated interaction with reputation tracking, honest behavior constitutes a Nash equilibrium when:

$$\Delta r \cdot V_{consensus} > B_{deviation} \tag{17}$$

where $\Delta r = 0.25$ is the reputation penalty for aggressive misbehavior, and $V_{consensus}$ is the value of successful consensus participation.

- **Incentive Compatibility:** The reputation mechanism is *designed* to create incentives for honest behavior—agents with high reputation receive more influence in future rounds. This analysis applies to the intended design, in which the penalty falls on the deviating agent; Section 6.6 measures the released implementation and finds the opposite (penalties land predominantly on honest, correct votes), so the equilibrium below describes a *repaired* tracker rather than the one we evaluated.
- **Nash Equilibrium:** Under the reputation-based weight adjustment, honest behavior constitutes a Nash equilibrium when the long-term benefit of reputation exceeds the short-term gain from strategic rejection.
- **Repeated Game Dynamics:** The reputation tracker maintains history over the last 10 rounds, creating a repeated game context where cooperation is sustainable.

This condition is satisfied when the reputation penalty ($\Delta r = 0.25$ for aggressive misbehavior) multiplied by the consensus value exceeds any private benefit from deviation; in practice $V_{consensus}$ is large, making honest behavior the dominant strategy. The qualification is important: under the *released* tracker of Section 6.6, where the aggressive tier fires predominantly on honest correct votes and misses colluders entirely (precision and recall 0.000; Section 6.6), a utility-maximising agent would instead be pushed toward accepting Byzantine proposals to keep ρ_i low. The repaired trigger proposed in Section 6.6 is a prerequisite for the equilibrium statement, not a consequence of the current code.

A.12 Practical Deployment Considerations

- **Network Requirements:** A2A-BFT requires authenticated channels with message ordering.

- **Hardware Specifications:** Two NVIDIA A800-SXM4-80GB GPUs with 512GB RAM.
- **Scalability:** Current implementation supports $n \leq 10$; distributed consensus optimization required for larger deployments.

A.13 Design Trade-offs

A2A-BFT balances several trade-offs:

- **Security vs. Efficiency:** Higher fault tolerance requires more validators, increasing communication overhead.
- **Reactivity vs. Stability:** Fast reputation updates improve responsiveness but may cause oscillation.
- **Decentralization vs. Performance:** Fully decentralized operation increases latency but eliminates single points of failure.

A.14 Future Work

1. **Adaptive Attacks:** This paper evaluates static attack patterns (strategic rejection, collusion, Sybil). Future work should explore adaptive attacks where Byzantine agents dynamically adjust behavior based on observed consensus state.
2. **Rational Byzantine Quantification:** Appendix A.11 provides qualitative Nash equilibrium analysis for rational agents. Future work should add numerical simulations to validate the equilibrium conditions under different game parameters.
3. **LLM Parameter Sensitivity:** The $p_h \geq 0.9$ assumption is idealized: the measured judge accuracy is ≈ 0.80 on GSM8K (Appendix A.6), and sensitivity curves across a range of p_h values are future work.
4. **Distributed Consensus:** Current implementation is centralized; distributing the consensus layer would enable $n > 10$ deployments.
5. **Repairing the Reputation Trigger:** Section 6.6 shows that the released ρ_i-keyed trigger mis-attributes the bulk of aggressive penalties in the attacked cells to honest validators. Future work should design and evaluate a trigger keyed on the proposal's validated quality (execution outcomes on verifiable domains, DID-signed cross-round consistency on semantic ones) and re-measure the mis-attribution rate under the same instrumented-replay protocol.

B Discussion

B.1 Implications for Multi-Agent Systems

A2A-BFT provides a foundation for reliable multi-agent collaboration under adversarial conditions. The three-phase semantic consensus design balances expressiveness with formal guarantees, while the reputation analysis of Section 6.6 identifies how the trigger must be redesigned to address practical concerns about agent reliability.

B.2 Privacy Considerations

The protocol assumes authenticated channels, which may reveal agent identities. Potential extensions include:

- Zero-knowledge proofs for private verification
- Differential privacy for reputation updates
- Secure multi-party computation for vote aggregation

B.3 Extended Related Work

B.3.1 Byzantine Fault Tolerance in Distributed Systems

Classical BFT protocols, beginning with PBFT (Castro & Liskov, 1999), establish consensus in distributed systems with Byzantine faults. Subsequent work including HotStuff (Yin et al., 2019) and Tendermint (Buchman et al., 2018) improved efficiency and practicality. Foundational results establish the fundamental limits of Byzantine agreement under partial synchrony (Dwork et al., 1988) and, in fully asynchronous settings, the $3f+1$ resilience bound (Lamport et al., 1982). Our work extends these foundations to the LLM-based agent context.

B.3.2 Multi-Agent Consensus with LLMs

A2A-Sim (Berdoz et al., 2026) provided the first systematic evaluation of LLM agent consensus, revealing only a 41.6% success rate even in benign, no-stake settings, with failures dominated by liveness loss. Their work highlighted the gap between theoretical BFT guarantees and practical LLM behavior. A2A addresses agent-to-agent interoperability; the complementary Model Context Protocol (MCP) (Anthropic, 2024) standardizes connections from LLM applications to data sources and tools, and is orthogonal to the consensus layer studied here.

Recent approaches have explored confidence-aware and robust aggregation mechanisms. CP-WBFT (Zheng et al., 2026) introduces confidence probes to weight agent contributions, while SAC (Self-Anchored Consensus) (Lee et al., 2026) proposes decentralized filter-and-refine aggregation with graph-robustness conditions for decentralized LLM agent networks. Multi-agent debate showed that structured argumentation can improve reasoning accuracy (Du et al., 2024; Liang et al., 2023), and Self-Consistency demonstrated that aggregating multiple reasoning paths reduces error rates (Wang et al., 2023). Surveys of LLM-based multi-agent systems (Guo et al., 2024) provide taxonomies of collaboration patterns; frameworks such as AutoGen (Wu et al., 2024) and CAMEL (Li et al., 2023) study multi-agent conversation among LLM instances. However, these approaches pursue collaboration quality rather than A2A protocol-level safety, and do not state formal safety and liveness guarantees with attack-resistance experiments. Our work differs by providing formal safety and liveness guarantees for consensus in the presence of Byzantine and soft faults within the A2A protocol framework, together with an execution-validated safety metric (wrong-commit rate).

B.3.3 Near-Neighbour Byzantine-Robust LLM Coordination

Three further lines sit closest to our threat model. First, multi-agent consensus seeking among LLMs has been studied in *benign* settings without adversarial agents (Chen et al., 2023); A2A-Sim (Berdoz et al., 2026) first introduced adversarial agents into this setting, and our protocol adds a commit rule with a proven boundary. Second, BlockAgents (Chen et al., 2024) anchors LLM agent coordination on a blockchain for Byzantine robustness, and DecentLLMs achieves Byzantine-robust decentralized coordination of LLM agents through evaluator-scored consensus (Jo & Park, 2025); both harden the *message exchange*, whereas A2A-BFT hardens the *commitment*—validation sits inside the vote-counting rule, so wrong commits are bounded by the validation oracle's error rate (Corollary 5.2), a property that, to our knowledge, none of these designs states. Third, failure taxonomies catalogue why LLM multi-agent systems fail in practice (Cemri et al., 2025), and benchmark evidence shows coordination quality degrading as the communication graph grows (Grötschla et al., 2025); our measured liveness and safety results give a protocol-level counterpart to these empirical catalogues.

B.3.4 Protocols and Verification Infrastructure

On the infrastructure side, surveys of agent interoperability protocols (Ehtesham et al., 2025) place A2A alongside MCP (Anthropic, 2024) and the Agent Network Protocol (ANP) as complementary standards; our contribution is orthogonal to this landscape—a consensus layer over any authenticated agent fabric; the W3C Decentralized Identifiers standard (W3C, 2022) supplies the identity primitive assumed in our threat model. On verification, process- and outcome-based verifiers improve single-model reasoning by ranking or filtering candidates (Cobbe et al., 2021; Lightman et al., 2024); A2A-BFT instead places verification *inside* the commit rule, which converts validator reliability into a stated wrong-commit bound rather than a post-hoc selection heuristic.