Robust Multi-Agent LLMs under Byzantine Faults

Anonymous ACL submission

Abstract

Large language model (LLM) agents increasingly collaborate over peer-to-peer networks to improve their reliability. However, these same interactions can also introduce vulnerability to unreliable or Byzantine agents that can propagate incorrect information and degrade overall system performance. To address this, we propose Self-Anchored Consensus (SAC), a fully decentralized filter-and-refine protocol in which agents iteratively exchange responses, locally evaluate and filter unreliable messages, and refine their own outputs. We present $(F{+}1)$-robustness conditions on the communication graph that ensure honest agents preserve and propagate reliable information despite Byzantine influence. Experiments on mathematical and commonsense reasoning benchmarks show that SAC effectively suppresses Byzantine influence and consistently improves performance across diverse communication topologies, whereas prior methods degrade significantly.

1 Introduction

Large language models (LLMs) are increasingly deployed as communicating agents in multi-agent systems (MAS), where multiple agents exchange responses, critique one another, and converge on collective answers (Guo et al., 2024; Du et al., 2024). We consider a peer-to-peer setting in which LLM agents collaboratively solve tasks with objectively verifiable answers through consensus.

As these systems move toward real-world deployment, a central challenge is *robustness* to faulty and adversarial agents. Individual agents may hallucinate, miscalculate, or strategically inject misleading responses. Since agents cannot locally distinguish trustworthy messages from Byzantine ones (Fig. 1), unreliable responses can contaminate otherwise reliable agents. The key question is therefore whether Byzantine influence can be *contained* so that honest agents preserve their base capability.

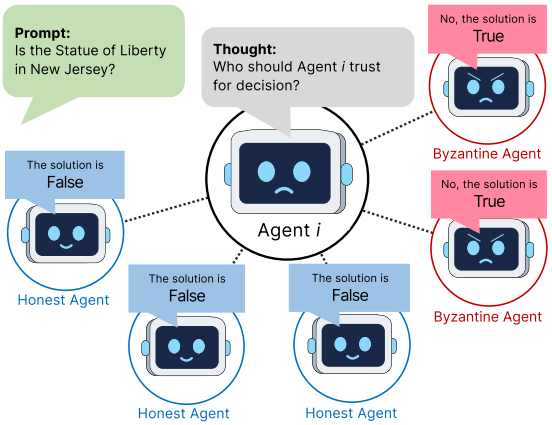

Figure 1: Decision ambiguity in the presence of Byzantine agents. An agent must decide which responses to trust among indistinguishable neighbor messages.

This setting naturally connects to classical Byzantine-resilient consensus, where some agents behave arbitrarily while honest nodes must still reach correct agreement (Lamport et al., 1982; LeBlanc et al., 2013; Su and Vaidya, 2021). Viewing LLM-MAS through this lens raises two questions: (i) how should agents decide which neighbors to trust, and (ii) how should the communication graph be designed to limit Byzantine influence?

Recent methods such as CP-WBFT (Zheng et al., 2026) aggregate neighbor responses using self-reported confidence scores. However, this implicitly assumes Byzantine agents report honest reliability signals, departing from the standard Byzantine setting where adversaries may falsify confidence values. Moreover, communication graphs are typically selected heuristically from generic topologies without principled fault-tolerance guarantees, resulting in unstable performance across different network structures (Zheng et al., 2026).

The official implementation and evaluation framework will be made publicly available upon acceptance.

We address these limitations by drawing on classical Byzantine-resilient consensus. Specifically, we adapt Mean-Subsequence-Reduced (MSR) filtering (Kieckhafer and Azadmanesh, 1994) and r-robustness (LeBlanc et al., 2013) to LLM-MAS by (i) replacing sender-reported confidence with receiver-side evaluations computed independently by each agent, and (ii) introducing an MSR-inspired filter-and-refine mechanism for iterative response refinement. We further design communication graphs under which honest agents remain robust to Byzantine influence.

Contributions. Our contributions are twofold. First, we introduce Self-Anchored Consensus (SAC), a decentralized MSR-inspired algorithm that enables robust LLM collaboration through receiver-side confidence evaluation and iterative filter-and-refine updates tailored to natural-language interactions. Second, we establish graph-theoretic robustness conditions under which honest agents contain Byzantine influence under SAC. Experiments show that our method consistently preserves honest-agent capability across diverse communication topologies, whereas prior approaches collapse catastrophically under Byzantine faults.

2 Related Works

LLM multi-agent systems. Multiple LLM instances coordinated as agents have been shown to outperform single-agent inference through debate (Du et al., 2024; Liang et al., 2024) and role-based frameworks (Wu et al., 2023; Hong et al., 2024; Guo et al., 2024). A growing body of work studies the reliability cost when such agents behave incorrectly (Tian et al., 2023; Yu et al., 2024; Zhang et al., 2024). Our work takes this failure mode as its starting point and asks what aggregation rule and graph structure are needed to contain it.

Byzantine-robust consensus in LLM-MAS. Three recent methods are closest to ours. CP-WBFT (Zheng et al., 2026) relies on self-reported confidence to select high-confidence responses, implicitly assuming Byzantine agents would honestly report their confidence. It also evaluates their method across the six hand-picked graphs, with wildly varying performance. Trusted Multi-LLMN (Luo et al., 2025) uses a leader-based BFT protocol in which consecutive Byzantine leaders force expensive re-elections. DecentLLMs (Jo and Park, 2025) removes the leader via a geometric-mean filter over evaluator scores, which is brittle under high-variance scoring and requires a fully-connected evaluator-worker graph. Our method differs on two axes at once: confidence is computed on the *receiver* side, so Byzantine agents cannot manipulate how others evaluate their output; and our communication topology is grounded on mathematically-driven robustness properties that limit Byzantine influence on general graphs.

Classical Byzantine-resilient algorithms. The Byzantine agreement problem (Lamport et al., 1982; Dolev et al., 1986) has been extensively studied to ensure reliability under arbitrary node failures, inspiring a broad line of work on fault-tolerant distributed computation. In particular, the *Weighted Mean-Subsequence-Reduced* (W-MSR) algorithm, together with r- and (r, s)-robustness conditions (LeBlanc et al., 2013), provides consensus guarantees despite Byzantine agents. These notions have been extended beyond consensus to distributed optimization (Sundaram and Gharesifard, 2019; Yuan and Ishii, 2025) and distributed learning (Xie et al., 2023; Ye et al., 2024), with applications in robotics and smart grids (Lee and Panagou, 2025; Yuan and Ishii, 2025). To our knowledge, this work is the first to bring MSR-type algorithms and robustness conditions to design Byzantine-resilient LLM-MAS.

3 Problem Setup

3.1 LLM Multi-Agent System

We consider a multi-agent system of n LLM agents, indexed by $\mathcal{V} = \{1, \ldots, n\}$, collaboratively answering a query x drawn from a task distribution $\mathcal{D}$. Agents exchange messages over an undirected, time-invariant communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where an edge $(i, j) \in \mathcal{E}$ indicates bidirectional exchange between agents i and j, and $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ denotes the neighbor set of agent i. We also define a directed graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$, where a directed edge $(i, j) \in \mathcal{E}'$ indicates that j receives messages from i. A directed graph contains a rooted out-branching if there exists $p \in \mathcal{V}$ that can reach all other nodes. Let $\mathbb{Z}_{\geq 0}$ denote the non-negative integers.

At each time step $t \in \mathbb{Z}_{\geq 0}$, agents communicate with their neighbors. Each agent i is associated with an underlying LLM $\mathcal{M}_i$ and produces a response $r_i^{(t)} \in \mathcal{Y}$ to a query x, where $\mathcal{Y}$ is the space of admissible outputs (e.g., numerical an-

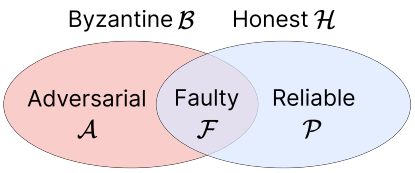

Figure 2: Venn diagram showing the partitioning of agents into reliable agents, faulty agents, and adversarial agents, among honest and Byzantine agents.

swers or binary safety labels). We assume that each query x admits a well-defined ground-truth answer $y^* \in \mathcal{Y}$, allowing responses to be classified as correct ($r_i^{(t)} = y^*$) or incorrect ($r_i^{(t)} \neq y^*$) at each round t.

3.2 Threat Model

We consider a setting in which a subset of agents are Byzantine, adapting the definition of LeBlanc et al. (2013) to LLM-MAS:

Definition 3.1 (Byzantine agent)**.** An agent $i \in \mathcal{B} \subset \mathcal{V}$ is *Byzantine* if, in some round t, it deviates from the ideal behavior and transmits unreliable information to neighbors. This may occur as follows:

- *Faulty Byzantine:* the agent $b \in \mathcal{F} \subset \mathcal{B}$ follows the prescribed protocol, but produces incorrect or noisy responses due to internal errors, limited capability, or uncertainty.
- *Adversarial Byzantine:* the agent $b \in \mathcal{A} \subset \mathcal{B}$ does not follow the prescribed protocol by acting arbitrarily, e.g., producing incorrect responses ($r_i^{(t)} \neq y^*$) or sending inconsistent information to different neighbors.

Byzantine agents consist of both faulty agents $\mathcal{F}$ and adversarial agents $\mathcal{A}$ such that $\mathcal{A} \cap \mathcal{F} = \emptyset$ and $\mathcal{A} \cup \mathcal{F} = \mathcal{B}$. Their behaviors may arise from systematic errors (e.g. hallucinations, stochastic errors) as well as adversarial manipulation, which are indistinguishable to non-Byzantine agents.

The remaining agents $\mathcal{P} := \mathcal{V} \setminus \mathcal{B}$ are referred to as reliable agents. We assume identities of Byzantine agents are unknown to reliable agents. Furthermore, we define the set of honest agents as $\mathcal{H} := \mathcal{V} \setminus \mathcal{A}$, i.e., non-adversarial agents. The relationships among these agent classes are illustrated in Fig. 2. To quantify the extent of Byzantine (both faulty and adversarial) agents, we adopt the F-local model (LeBlanc et al., 2013):

Definition 3.2 (F-local (LeBlanc et al., 2013))**.** A set $\mathcal{S} \subset \mathcal{V}$ is F-local if every node outside $\mathcal{S}$ has at most F neighbors in $\mathcal{S}$, i.e., $|\mathcal{N}_i \cap \mathcal{S}| \leq F$ for all $i \in \mathcal{V} \setminus \mathcal{S}$.

3.3 Graph Robustness

We now introduce the notion of r-robustness (LeBlanc et al., 2013):

Definition 3.3 (r-robust (LeBlanc et al., 2013))**.** A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *r-robust* if for every pair of nonempty, disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of the subsets contains a node with at least r neighbors outside itself. That is, there exists a node $i \in \mathcal{S}_k$ such that $|\mathcal{N}_i \setminus \mathcal{S}_k| \geq r$ for some $k \in \{1, 2\}$.

Higher robustness levels limit the influence of more Byzantine agents on reliable ones but require denser communication graphs. We now present a useful property of r-robustness:

Lemma 3.4. *Given an r-robust graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be a directed graph produced by removing upto k incoming edges of each agent in $\mathcal{G}$. Then, $\mathcal{G}'$ is $(r-k)$-robust.*

The notion of r-robustness is a preferable graph-theoretic condition against Byzantine agents, as it achieves equivalent resilience with sparser connectivity than classical metrics such as connectivity or minimum degree (LeBlanc et al., 2013; Pirani et al., 2023). This sparsity is particularly valuable in LLM-MAS, where each edge in $\mathcal{E}$ corresponds to an LLM-to-LLM message exchange and thus directly contributes to per-round inference cost. We later leverage this notion of robustness to design communication networks that systematically limit Byzantine influence in LLM-MAS.

3.4 Problem Statement

Given n LLM agents, a query x, and an F-local Byzantine set $\mathcal{B} = \mathcal{F} \cup \mathcal{A}$, our objective is twofold: (i) ensure that reliable agents $\mathcal{P}$ remain robust to Byzantine messages, and (ii) enable collaborative refinement that helps all honest agents $\mathcal{H} = \mathcal{P} \cup \mathcal{F}$ move toward the correct answer in a fully decentralized peer-to-peer network.

We address this in two steps: first, we develop a decentralized protocol where agents iteratively refine responses using neighbor messages while filtering Byzantine influence; second, we use r-robustness to characterize communication graphs under which honest agents can reliably refine their responses.

Remark 3.5. Adversarial agents $\mathcal{A}$ may arbitrarily deviate from the protocol, whereas faulty agents $\mathcal{F}$

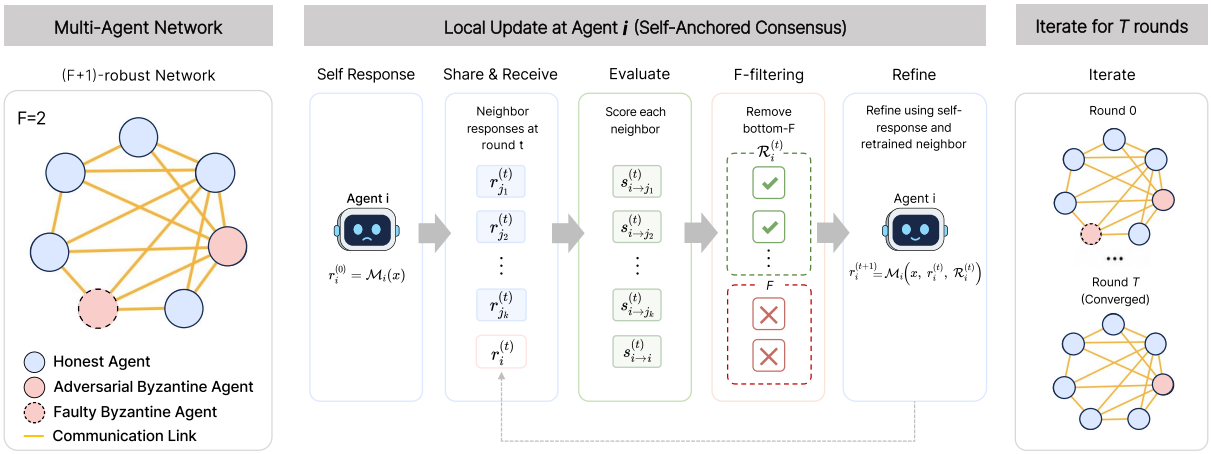

Figure 3: Overview of Self-Anchored Consensus (SAC) on an $(F{+}1)$-robust network. At each round, agent i generates its own response, exchanges responses with neighbors, scores them itself, filters out the bottom-F neighbors, and refines its response by re-prompting on the retained set. The procedure is iterated for T rounds.

follow the protocol but may produce unreliable outputs. Although both are Byzantine, faulty agents can be viewed as *temporary* Byzantine agents, since iterative peer-to-peer refinement may eventually correct their responses and restore reliable behavior.

4 Method

4.1 Self-Anchored Consensus (SAC)

Here we describe Self-Anchored Consensus (SAC), with which each agent $i \in \mathcal{V}$ iteratively refines its response by incorporating trusted neighbor responses while filtering potentially Byzantine information through four steps. Given query x, every agent $i \in \mathcal{V}$ first produces an initial response

$$r_i^{(0)} \;=\; \mathcal{M}_i(x). \tag{1}$$

At each subsequent round $t \geq 0$, every agent $i \in \mathcal{V}$ performs the following steps.

Broadcast responses. Every agent i broadcasts its current response $r_i^{(t)}$ to its neighbors.

Score neighbors. Upon receiving neighbor responses $\{r_j^{(t)}\}_{j\in\mathcal{N}_i}$, agent i computes, for each neighbor $j \in \mathcal{N}_i$, an evaluation score

$$s_{i\to j}^{(t)} \;=\; \phi_i\big(x,\, r_j^{(t)}\big) \;\in\; [0,1], \tag{2}$$

where ϕ_i is a scoring function implemented by agent i's own LLM. In our implementation, ϕ_i is realized via a verification prompt that asks $\mathcal{M}_i$ to judge the correctness of $r_j^{(t)}$ with respect to x, returning a calibrated score in $[0,1]$; the full template is given in Section B.

The key distinction from prior work (Zheng et al., 2026) is that $s_{i\to j}^{(t)}$ depends only on agent i's own computation applied to the content of $r_j^{(t)}$, and in particular does *not* rely on any score reported by agent j. This removes the possibility for a Byzantine agent to manipulate its own reliability by reporting an inflated self-confidence.

Filter. Let $\mathcal{L}_i^{(t)} \subseteq \mathcal{N}_i$ be the set of neighbors whose score is strictly below agent i's self-score,

$$\mathcal{L}_i^{(t)} \;=\; \big\{\, j \in \mathcal{N}_i \;\big|\; s_{i\to j}^{(t)} < s_{i\to i}^{(t)} \,\big\}. \tag{3}$$

Agent i then discards the $\min(F, |\mathcal{L}_i^{(t)}|)$ lowest-scoring elements of $\mathcal{L}_i^{(t)}$, yielding the retained set

$$\mathcal{R}_i^{(t)} \;=\; \mathcal{N}_i \setminus \mathrm{bottom}_F(\mathcal{L}_i^{(t)}), \tag{4}$$

where $\mathrm{bottom}_F(\cdot)$ returns the F lowest-scoring elements of its argument, or the entire argument when $|\mathcal{L}_i^{(t)}| \leq F$.

Refine. Agent i refines its response by conditioning $\mathcal{M}_i$ on its current response and the retained neighbor responses, weighted by the corresponding receiver-side scores:

$$r_i^{(t+1)} \;=\; \mathcal{M}_i\Big(x,\; r_i^{(t)},\; \big\{(r_j^{(t)},\, s_{i\to j}^{(t)})\big\}_{j\in\mathcal{R}_i^{(t)}}\Big). \tag{5}$$

The prompt template for Eq. (5) instructs $\mathcal{M}_i$ to prefer high-scored responses while retaining the option to keep its own response if none of the retained neighbor responses improves upon it; see Section B for the full template.

After T rounds (a user-specified parameter), the final response of agent $i \in \mathcal{V}$ is $r_i^{(T)}$. The overall procedure of SAC is summarized in Algorithm 1.

Algorithm 1 Self-Anchored Consensus (SAC)

Require: query x; adversary bound F; number of rounds T

1: each agent i computes $r_i^{(0)} = \mathcal{M}_i(x)$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: each agent i broadcasts $r_i^{(t)}$ to $\mathcal{N}_i$
4: compute $s_{i\to j}^{(t)}$ for all $j \in \mathcal{N}_i \cup \{i\}$ via Eq. (2)
5: $\mathcal{L}_i^{(t)} \leftarrow \{j \in \mathcal{N}_i : s_{i\to j}^{(t)} < s_{i\to i}^{(t)}\}$
6: $\mathcal{R}_i^{(t)} \leftarrow \mathcal{N}_i \setminus \mathrm{bottom}_F(\mathcal{L}_i^{(t)})$
7: update $r_i^{(t+1)}$ via Eq. (5)
8: **end for**
9: **return** $r_i^{(T)}$

Remark 4.1. Note that Algorithm 1 is fully decentralized; each LLM agent operates using only local information from its neighbors and does not require knowledge of the global network topology $\mathcal{G}$ or the identities of Byzantine agents.

4.2 Robust Topological Conditions

While Algorithm 1 is fully decentralized, its effectiveness depends on the connectivity of the underlying communication graph. Since the filtering step in line 6 of SAC removes up to F neighbors per round, an agent with $|\mathcal{N}_i| \leq F$ may have no incoming information after filtering, reducing the algorithm to repeated self-updates. We therefore design the topology to satisfy $(F+1)$-robustness, which ensures two key properties.

Property 4.2. Consider an LLM-MAS connected through a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an F-local Byzantine set $\mathcal{B} \subset \mathcal{V}$. Let $\mathcal{S}_i^{(t)} = \{\phi_i(x, r_j^{(t)}) \mid j \in \mathcal{P}\}$ denote the set of evaluation scores assigned by agent i at time t to responses from reliable agents. If $\mathcal{G}$ is $(F+1)$-robust, then for every $k \in \mathcal{R}_i^{(t)}$ and time $t \in \mathbb{Z}_{\geq 0}$, the evaluation score $s_{i\to k}^{(t)}$ is at least as high as the minimum evaluation score assigned to any reliable agent's response, i.e., $s_{i\to k}^{(t)} \geq \min \mathcal{S}_i^{(t)}$, $\forall k \in \mathcal{R}_i^{(t)}$.

Proof. By Lemma 5 of LeBlanc et al. (2013), since $\mathcal{G}$ is $(F+1)$-robust, we have $|\mathcal{N}_i| \geq F+1$ for all $i \in \mathcal{V}$. Since the filtering step of SAC removes at most F neighbor indices corresponding to the lowest evaluation scores, and $|\mathcal{N}_i| \geq F+1$, at least one neighbor remains in $\mathcal{R}_i^{(t)}$, i.e., $\mathcal{R}_i^{(t)} \neq \emptyset$. Furthermore, under the F-local model, and since at most F indices are removed from the lowest end of the ordered scores, $\mathcal{R}_i^{(t)}$ contains only agent indices whose responses have scores no lower than the minimum score among reliable agents. □

Property 4.2 guarantees that, after filtering, each agent retains at least one response whose quality is comparable to that of a reliable agent, *even if the response does not originate from a reliable agent.* This allows honest agents to safely refine their outputs using information of comparable quality to that produced by reliable agents, despite the presence of Byzantine agents.

Property 4.3. Consider an LLM-MAS connected through a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an F-local Byzantine set $\mathcal{B} \subset \mathcal{V}$. Let $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be a directed graph obtained by removing up to any F incoming edges of each agent in $\mathcal{G}$. If $\mathcal{G}$ is $(F+1)$-robust, then $\mathcal{G}'$ contains a rooted-out branching.

Property 4.3 is a direct consequence of Lemma 3.4 and Lemma 7 of (LeBlanc et al., 2013). It guarantees that F-filtering does not disconnect the network, thereby preventing fragmentation into isolated subgroups that may converge to different or incorrect solutions. Therefore, under $(F+1)$-robustness, the subgraph induced by reliable agents remains connected even after filtering out potentially malicious messages.

Together, these properties explain why $(F+1)$-robustness is a natural design choice: it provides *local resilience* against Byzantine agents through filtering and *global information-propagation guarantees* across the entire network.

5 Experimental Results

5.1 Evaluation Metrics

Let $y^\star$ denote the ground-truth answer to query $x \sim \mathcal{D}$, and $r_i^{(t)}$ the response of agent i at round t. We partition honest agents $\mathcal{H}$ into strong and weak groups, where strong and weak agents are treated as reliable and faulty (non-adversarial Byzantine) agents, respectively. Following Zheng et al. (2026), we report the following metrics (higher is better).

Initial / Final Agent Accuracy (IAA / FAA). Average per-agent accuracy before communication ($t = 0$) and after T consensus rounds ($t = T$).

Byzantine Fault Tolerance Improvement (BFTI). $\mathrm{BFTI} = \mathrm{FAA} - \mathrm{IAA}$ measures the gain or degradation introduced by the consensus protocol.

Round-level Accuracy (RA). The fraction of queries for which the majority vote across all agents

Dataset	Method	Topology	IAA	FAA	BFTI	RA	W IAA→FAA	S IAA→FAA	H-Majority
MATH	CP-WBFT	MERG	52.3%	43.6%	−8.7%	35.0%	25→58%	79→48%	38.0%
		Complete	52.7%	42.3%	−10.4%	43.0%	23→49%	81→50%	49.0%
		Erdős-Rényi	52.6%	35.1%	−17.5%	39.0%	25→42%	80→41%	42.0%
	SAC (Ours)	MERG	50.1%	53.1%	+3.0%	79.0%	22→26%	77→80%	79.0%
		Complete	53.7%	54.1%	+0.4%	80.0%	26→34%	81→78%	78.0%
		Erdős-Rényi	53.3%	55.0%	+1.7%	79.0%	25→38%	81→77%	78.0%
Commonsense	CP-WBFT	MERG	69.5%	37.1%	−32.4%	26.7%	77→52%	83→39%	26.7%
		Complete	69.0%	14.3%	−54.7%	16.7%	72→17%	85→17%	16.7%
		Erdős-Rényi	69.0%	20.5%	−48.5%	23.3%	75→25%	83→23%	23.3%
	SAC (Ours)	MERG	68.6%	66.7%	−1.9%	76.7%	73→77%	83→78%	80.0%
		Complete	69.5%	69.5%	+0.0%	83.3%	77→80%	83→82%	83.3%
		Erdős-Rényi	69.0%	66.2%	−2.8%	73.3%	73→75%	84→78%	76.7%

Table 1: Comparison of CP-WBFT and SAC across three communication topologies on two reasoning benchmarks: 100 Level 4 problems sampled from the Hendrycks MATH test set and 30 sampled instances from Commonsense170k. All settings use $n = 7$ agents (4 strong honest, 2 weak honest (faulty Byzantine), 1 adversarial Byzantine, $F = 3$, $r = 4$), with the adversarial Byzantine agent reporting a falsified self-confidence of 1.0. On MATH we use gpt-4o-mini as the strong model and gpt-3.5-turbo as the weak model; on Commonsense170k we use gpt-5 as the strong model and gpt-4o as the weak model.

		MATH			Commonsense		
Method	Round	MERG (W/S)	Complete (W/S)	Erdős-Rényi (W/S)	MERG (W/S)	Complete (W/S)	Erdős-Rényi (W/S)
CP-WBFT	Init	25.0 / 79.0	23.0 / 80.8	25.0 / 79.5	76.7 / 83.3	75.0 / 85.0	75.0 / 83.3
	Rnd 1-6	57.5 / 47.5	49.0 / 49.5	41.5 / 40.8	51.7 / 39.2	13.3 / 13.3	25.0 / 23.3
SAC (Ours)	Init	21.5 / 77.0	26.0 / 81.0	25.0 / 80.8	73.3 / 83.3	75.0 / 83.3	73.3 / 84.2
	Rnd 1	21.5 / 77.0	33.5 / 77.8	33.0 / 75.8	83.3 / 67.5	83.3 / 66.7	83.3 / 62.5
	Rnd 2	27.5 / 80.2	34.5 / 80.5	34.5 / 77.0	73.3 / 79.2	73.3 / 80.0	70.0 / 77.5
	Rnd 3	29.0 / 78.8	40.0 / 75.5	33.5 / 77.0	81.7 / 74.2	83.3 / 73.3	81.7 / 70.0
	Rnd 4	30.5 / 80.8	37.0 / 79.2	36.0 / 78.2	80.0 / 78.3	80.0 / 76.7	76.7 / 75.0
	Rnd 5	27.5 / 80.8	37.0 / 80.8	39.5 / 75.8	80.0 / 71.7	76.7 / 70.0	78.3 / 67.5
	Rnd 6	25.5 / 80.2	34.0 / 77.8	38.5 / 77.0	76.7 / 78.3	80.0 / 83.3	75.0 / 78.3

Table 2: Per-round weak / strong honest accuracy (W/S, in %) across three communication topologies on the Hendrycks MATH test set (Level 4, 100 sampled problems) and Commonsense170k (30 sampled instances).

in $\mathcal{V}$ at round T matches $y^\star$.

Group-wise Accuracy (W_IAA->FAA, S_IAA->FAA). Accuracy change of the weak and strong honest groups before and after consensus.

Honest Majority (H-Majority). The fraction of queries for which the majority vote among honest agents $\mathcal{H}$ at round T matches $y^\star$.

5.2 Experimental Setups

Datasets. We extend the evaluation protocol of Zheng et al. (2026), which evaluates Byzantine fault tolerance on 10 hand-curated GSM8K problems. Since GSM8K mainly consists of grade-school arithmetic and amplifies the strong–weak capability gap, we additionally evaluate on more challenging benchmarks. For closed-source LLM experiments, we use 100 sampled Level 4 problems from the Hendrycks MATH test set (Hendrycks et al., 2021) and 30 sampled instances from the Commonsense170k collection introduced by Hu et al. (2023). For open-weight LLM experiments, we evaluate on five commonsense reasoning benchmarks: ARC-C (Clark et al., 2018), HellaSwag (Zellers et al., 2019), BoolQ (Clark et al., 2019), OBQA (Mihaylov et al., 2018), and RTE (Dagan et al., 2005), using 50 sampled questions from each benchmark. The full sampled problem lists are provided in Appendix C.1.

Network. Each network contains $n = 7$ agents: 4 strong honest, 2 weak honest (faulty Byzantine), and 1 adversarial Byzantine. We set $F = 3$ and evaluate three $(F+1)$-robust topologies: γ-MERG (Lee and Panagou, 2026), complete (LeBlanc et al., 2013), and Erdős–Rényi (Erdős and Rényi, 1961; Zhang et al., 2015). Additional topology details are provided in Appendix A.

Models. For closed-source experiments on Hendrycks MATH, we use `gpt-3.5-turbo` and `gpt-4o-mini` as the weak and strong models, respectively. For closed-source commonsense reasoning experiments, we use `gpt-4o` and `gpt-5` as the weak and strong models, respectively. For open-weight experiments, we use

Method	Topology	IAA	FAA	BFTI	RA	W IAA→FAA	S IAA→FAA	H-Majority
CP-WBFT	MERG	59.0%	42.5%	−16.5%	40.0%	67→55%	70→47%	40.0%
	Complete	59.1%	33.3%	−25.9%	38.8%	67→39%	70→39%	38.8%
	Erdős–Rényi	59.2%	33.1%	−26.1%	39.6%	67→38%	70→39%	39.6%
SAC (Ours)	MERG	59.1%	62.3%	+3.2%	78.0%	67→64%	70→77%	78.4%
	Complete	59.0%	62.9%	+3.9%	77.2%	67→66%	70→77%	77.6%
	Erdős–Rényi	59.0%	62.5%	+3.5%	76.8%	67→65%	70→77%	77.2%

Table 3: Average over five commonsense reasoning benchmarks (ARC-Challenge, HellaSwag, BoolQ, OpenBookQA, and RTE), using 50 sampled questions from each benchmark. All settings use n=7 agents (4 strong honest, 2 weak honest, 1 adversarial Byzantine, F=3, r=4). We use Qwen3-4B as the strong model and Qwen2.5-1.5B-Instruct as the weak model. Detailed results for each individual benchmark are provided in Appendix D.

Method	Round	MERG (W/S)	Complete (W/S)	Erdős–Rényi (W/S)
CP-WBFT	Init	66.6 / 69.9	66.6 / 70.2	66.8 / 70.2
	Rnd 1-6	54.6 / 47.1	38.8 / 38.8	37.6 / 39.1
SAC (Ours)	Init	66.8 / 70.1	66.8 / 69.9	66.6 / 69.9
	Rnd 1	61.6 / 74.2	61.6 / 74.8	61.2 / 73.8
	Rnd 2	64.8 / 77.6	65.2 / 77.6	64.8 / 76.0
	Rnd 3	62.8 / 75.7	62.6 / 75.6	61.8 / 76.3
	Rnd 4	64.6 / 76.8	66.0 / 77.3	64.8 / 76.5
	Rnd 5	63.0 / 75.2	63.0 / 75.8	62.0 / 75.6
	Rnd 6	64.2 / 77.0	66.0 / 77.1	64.6 / 77.0

Table 4: Per-round weak / strong honest accuracy (W/S, in %) averaged across five commonsense reasoning benchmarks (ARC-C, HellaSwag, BoolQ, OBQA, and RTE), using 50 sampled questions from each benchmark, across three $(F{+}1)$-robust communication topologies.

`Qwen2.5-1.5B-Instruct` as the weak model and `Qwen3-4B` as the strong model.

Byzantine model. Adversarial Byzantine agents return out-of-distribution weak-model responses with falsified self-confidence scores of 1.0, simulating dishonest-confidence attacks against CP-WBFT while remaining indistinguishable from reliable agents. Faulty Byzantine agents follow the same protocol as reliable agents.

5.3 Experimental Results

5.3.1 Closed-source LLMs

MATH. CP-WBFT consistently yields negative BFTI across all topologies (-8.7% to -17.5%; Table 1), with strong-agent accuracy decreasing after the first communication round while weak-agent accuracy increases. In contrast, SAC achieves non-negative BFTI ($+0.4\%$ to $+3.0\%$), preserves strong-agent capability, and maintains stable per-round behavior. H-Majority reaches 78–79% across all topologies.

Commonsense reasoning. A similar trend is observed on Commonsense170k. CP-WBFT suffers severe degradation under Byzantine influence (-32.4% to -54.7% BFTI), with both weak and strong groups collapsing immediately after Rnd 1. In contrast, SAC maintains near-zero degradation (-2.8% to $+0.0\%$) while preserving strong agents and stabilizing weak-agent performance. H-Majority reaches 76.7–83.3% across all topologies.

5.3.2 Open-weight LLMs

A similar trend is observed for open-weight LLMs (Tables 3 and 4). CP-WBFT consistently yields negative BFTI across all topologies (-16.5% to -26.1%), with both weak and strong groups collapsing immediately after the first communication round. In contrast, SAC consistently achieves positive BFTI ($+3.2\%$ to $+3.9\%$) with stable per-round behavior. H-Majority improves to 77.2–78.4%, compared to 38.8–40.0% for CP-WBFT. Across all topologies, SAC preserves strong-agent capability while preventing collapse of weaker agents.

5.4 Ablation Study

For computational efficiency, the ablation studies use 30 sampled Level 4 problems from the Hendrycks MATH test set. We study SAC along two axes: the composition of the Byzantine set and the network size n.

Varying byzantine-to-weak composition. Table 5 fixes the strong honest count at 4 and varies $(b, w) \in \{(1,2),(2,1),(3,0)\}$ at n=7. CP-WBFT

Method	Topology	IAA (%)			FAA (%)			BFTI (%)			RA (%)			H-Majority (%)		
		b=1 w=2	b=2 w=1	b=3 w=0	b=1 w=2	b=2 w=1	b=3 w=0	b=1 w=2	b=2 w=1	b=3 w=0	b=1 w=2	b=2 w=1	b=3 w=0	b=1 w=2	b=2 w=1	b=3 w=0
CP-WBFT	MERG	55.2	48.1	44.3	44.8	31.4	29.0	-10.4	-16.7	-15.3	30.0	40.0	50.0	40.0	43.3	50.0
	Complete	56.2	51.9	49.0	31.4	30.5	34.3	-24.8	-21.4	-14.7	33.3	43.3	60.0	36.7	43.3	60.0
	Erdős-Rényi	56.2	52.9	48.1	40.0	31.4	29.5	-16.2	-21.5	-18.6	43.3	40.0	50.0	46.7	40.0	53.3
SAC (Ours)	MERG	54.8	49.5	47.1	62.4	52.9	48.1	$+7.6$	$+3.4$	$+1.0$	90.0	83.3	86.7	96.7	83.3	83.3
	Complete	55.7	51.0	47.6	59.5	54.8	52.4	$+3.8$	$+3.8$	$+4.8$	83.3	86.7	93.3	86.7	83.3	93.3
	Erdős-Rényi	54.3	51.4	46.7	60.5	57.1	47.1	$+6.2$	$+5.7$	$+0.4$	80.0	93.3	83.3	86.7	90.0	86.7

Table 5: Ablation over byzantine-to-weak composition at n=7 on the Hendrycks MATH test set (30 sampled Level 4 problems). The byzantine agent count (b) and weak honest count (w) are varied while the strong honest count is fixed at 4 (F=3, T=6, r=4). Strong/weak agents use `gpt-4o-mini`/`gpt-3.5-turbo`.

Method	Topology	r	IAA (%)	FAA (%)	BFTI (%)	RA (%)	H-Majority (%)
CP-WBFT	MERG	5	63.7	37.0	-26.7	20.0	23.3
	Complete	5	61.9	29.6	-32.3	33.3	33.3
	Preferential	4	64.8	44.1	-20.7	26.7	26.7
	Erdős-Rényi	4	63.0	50.4	-12.6	36.7	43.3
SAC (Ours)	MERG	5	62.6	64.8	$+2.2$	90.0	90.0
	Complete	5	60.4	62.2	$+1.8$	86.7	86.7
	Preferential	4	62.6	65.6	$+3.0$	90.0	90.0
	Erdős-Rényi	4	62.6	60.4	-2.2	86.7	86.7

Table 6: Scalability to larger networks (n=9 agents) on the Hendrycks MATH test set (Level 4, 30 sampled problems). Configuration: 6 strong honest, 2 weak honest, and 1 adversarial Byzantine (F=3, T=6). Strong/weak agents use `gpt-4o-mini`/`gpt-3.5-turbo`.

yields negative BFTI across all settings, whereas SAC maintains non-negative BFTI in nearly all cases with high H-Majority (83.3–96.7%). Performance does not degrade monotonically as b increases, consistent with $(F+1)$-robustness limiting adversarial influence under the F-local assumption.

Scaling to larger networks. Table 6 scales the system to n=9 with a $6/2/1$ split at the same adversary bound F=3. While CP-WBFT still degrades substantially, SAC maintains positive BFTI on three of four topologies and H-Majority of 86.7–90.0%, including on a 4-robust preferential-attachment graph. These results suggest that our method scales to larger networks without retuning.

6 Discussion

Self-reported confidence creates a point of vulnerability. Across all evaluated topologies and benchmarks, a single adversarial agent with falsified self-confidence consistently collapses CP-WBFT, producing strongly negative BFTI and rapid degradation after the first communication round. This suggests that manipulable trust signals themselves become a critical vulnerability.

Receiver-side evaluation contains Byzantine influence. By construction, SAC prevents Byzantine agents from inflating their own reliability through receiver-side evaluation. Empirically, SAC achieves non-negative BFTI in nearly all settings and consistently maintains substantially higher H-Majority than CP-WBFT.

Strong agents are preserved while weak agents are lifted. CP-WBFT may improve weak-agent accuracy, but by dragging strong agents toward incorrect responses, degrading overall reliability. In contrast, SAC preserves strong-agent capability with only minor degradation while improving weak agents across all topologies. These results suggest that SAC acts as a *protective filtering mechanism* against Byzantine faults.

7 Conclusion

We studied robust decision-making in decentralized LLM-MAS under Byzantine faults. We proposed Self-Anchored Consensus (SAC), a fully decentralized filter-and-refine protocol with receiver-side evaluation, and established $(F+1)$-robustness conditions under which honest agents reliably refine their responses despite Byzantine influence. Experiments on mathematical and commonsense reasoning benchmarks show that SAC consistently maintains robustness across diverse topologies, whereas prior methods collapse under Byzantine attacks.

Limitations

One limitation of our evaluation is that the closed-source experiments rely on repeated GPT API calls over multiple communication rounds, which makes large-scale evaluation computationally expensive. As a result, the closed-source commonsense experiments use 30 sampled Commonsense170k instances, whereas the open-weight experiments are evaluated at a larger scale using 50 sampled questions from each of five commonsense reasoning benchmarks. Although the overall trends remain consistent across both settings, future work should further scale the evaluation to larger datasets, more diverse benchmarks, and broader collections of both closed-source and open-weight LLMs.

One of the main limitations of our method is that SAC's robustness also rests on the quality of the receiver-side scoring function ϕ_i; an adversary that crafts responses appearing correct to ϕ_i but factually wrong is not addressed by the current filtering mechanism, and we leave this attack vector to future work. In fact, the practical limitations of Properties 4.2 and 4.3 also depend on (i) the fidelity of the scoring function ϕ_i in identifying high-quality responses and (ii) the agent's ability to integrate neighbor information during refinement. Since SAC uses each agent's own output as an "anchor" for filtering, a key failure mode arises when a faulty agent is overconfident and assigns low scores to correct neighbor responses, thereby discarding useful information and remaining trapped in error despite reliable neighbors. While our results show that SAC consistently improves faulty agents in practice, this coupling between reasoning and evaluation is a limitation. Decoupling these roles (e.g., via cross-agent verification) is an interesting direction for future work.

In addition, our evaluation is limited to networks of $n \in \{7, 9\}$, a single adversarial behavior (cached weak-model replay with falsified confidence), and single-turn responses. Larger populations, dynamic topologies, multi-turn reasoning, and tool use are outside the present scope.

Ethical considerations

This work studies robustness in decentralized multi-agent LLM systems under Byzantine faults. Our experiments are conducted using publicly available benchmark datasets and commercial or open-weight language models. No human subjects, personal data, or sensitive information are involved.

Although our work considers adversarial agents that provide misleading responses or falsified confidence scores, these settings are studied solely for robustness evaluation and reliability analysis. We hope this work contributes to the development of more reliable and trustworthy collaborative LLM systems.

References

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. 1986. Reaching approximate agreement in the presence of faults. *J. ACM*, 33(3):499–516.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pages 11733–11763. PMLR.

Paul Erdős and Alfréd Rényi. 1961. On the strength of connectedness of a random graph. *Acta Mathematica Hungarica*, 12(1-2):261–267.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 8048–8057.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran,

Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yongrae Jo and Chanik Park. 2025. Byzantine-robust decentralized coordination of LLM agents. *arXiv preprint arXiv:2507.14928*.

R.M. Kieckhafer and M.H. Azadmanesh. 1994. Reaching approximate agreement with mixed-mode faults. *IEEE Transactions on Parallel and Distributed Systems*, 5(1):53–63.

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

Heath J. LeBlanc, Haotian Zhang, Xenofon Koutsoukos, and Shreyas Sundaram. 2013. Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781.

Haejoon Lee and Dimitra Panagou. 2025. Distributed resilience-aware control in multi-robot networks. pages 3868–3875.

Haejoon Lee and Dimitra Panagou. 2026. Minimal construction of graphs with maximum robustness. *arXiv preprint arXiv:2507.00415*. V2, February 2026.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2024. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Haoxiang Luo, Gang Sun, Yinqiu Liu, Dongzhao Zhao, Dusit Niyato, Han Yu, and Schahram Dustdar. 2025. A weighted Byzantine fault tolerance consensus driven trusted multiple large language models network. *arXiv preprint arXiv:2505.05103*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Mohammad Pirani, Aritra Mitra, and Shreyas Sundaram. 2023. Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey. *Automatica*, 157:111264.

Lili Su and Nitin H. Vaidya. 2021. Byzantine-resilient multiagent optimization. *IEEE Transactions on Automatic Control*, 66(5):2227–2233.

Shreyas Sundaram and Bahman Gharesifard. 2019. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3):1063–1076.

Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. 2023. Evil geniuses: Delving into the safety of LLM-based agents. *arXiv preprint arXiv:2311.11855*.

James Usevitch and Dimitra Panagou. 2020. Determining r-and (r, s)-robustness of digraphs using mixed integer linear programming. *Automatica*, 111:108586.

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. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation.

Yijing Xie, Shaoshuai Mou, and Shreyas Sundaram. 2023. Communication-efficient and resilient distributed q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):3351–3364.

Lintao Ye, Martin Figura, Yixuan Lin, Mainak Pal, Pranoy Das, Ji Liu, and Vijay Gupta. 2024. Resilient multiagent reinforcement learning with function approximation. *IEEE Transactions on Automatic Control*, 69(12):8497–8512.

Miao Yu, Shilong Wang, Guibin Zhang, Junyuan Mao, Chenlong Yin, Qijiong Liu, Qingsong Wen, Kun Wang, and Yang Wang. 2024. NetSafe: Exploring the topological safety of multi-agent networks. *arXiv preprint arXiv:2410.15686*.

Liwei Yuan and Hideaki Ishii. 2025. Resilient distributed economic dispatch in smart grids. *IEEE Transactions on Automatic Control*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Haotian Zhang, Elaheh Fata, and Shreyas Sundaram. 2015. A notion of robustness in complex networks. *IEEE Transactions on Control of Network Systems*, 2(3):310–320.

Lifan Zheng, Jiawei Chen, Qinghong Yin, Jingyuan Zhang, Xinyi Zeng, and Yu Tian. 2026. Rethinking the reliability of multi-agent system: A perspective from Byzantine fault tolerance. In *Proceedings of the AAAI Conference on Artificial Intelligence*. ArXiv:2511.10400.

Appendix

A Graph Topologies

We evaluate our method on four graph topologies, which we describe below:

- γ-MERGs (Lee and Panagou, 2026): These graphs achieve maximum robustness, i.e., $r = \lceil n/2 \rceil$, for a given number of nodes n, while using the minimal set of edges possible. We distinguish two cases for constructing a γ-MERG:
  - **If n is odd:** Let $\mathcal{X} \subset \mathcal{V}$ be a set of $\gamma + 1$ nodes forming a complete subgraph. Connect each of the remaining $\gamma - 1$ nodes to any γ nodes in $\mathcal{X}$.
  - **If n is even:** Let $\mathcal{X} \subset \mathcal{V}$ be a set of γ nodes, each adjacent to all other nodes in $\mathcal{V}$. Then select $\left\lceil \frac{\gamma-2}{2} \right\rceil$ disjoint pairs of nodes in $\mathcal{X}$ and remove the edge between each pair.

  By Theorem 1 of (Lee and Panagou, 2026), the resulting graph is $\lceil n/2 \rceil$-robust.

- Complete graphs: In this case, every agent is connected to every other agent. By Lemma 4 of LeBlanc et al. (2013), a complete graph with n nodes is $r = \lceil n/2 \rceil$-robust.

- Preferential-attachment graphs: To construct r-robust graphs, we first form a complete graph with $2r - 1$ nodes. By Lemma 4 of LeBlanc et al. (2013), this subgraph is r-robust. The remaining $n - 2r + 1$ nodes are then each connected to any r nodes in the original set of nodes in the complete graph. By Theorem 5 of LeBlanc et al. (2013), the resulting graph is then r-robust.

- Erdős–Rényi random graphs (Erdős and Rényi, 1961): These graphs are parameterized by the number of nodes n and an edge probability p. The presence of each edge is independent of every other edge, and each edge is included independently with probability p. As shown in Theorem 3 of Zhang et al. (2015), choosing $p = \frac{\ln n + (r-1)\ln\ln n}{n}$ ensures that the graph is r-robust with high probability as $n \to \infty$. We sample graphs using this probability and verify r-robustness prior to simulation using the method from Usevitch and Panagou (2020).

B Prompt Templates

We instantiate the receiver-side scoring function ϕ_i (Eq. (2)) and the refine operator (Eq. (5)) using the prompt templates below. We provide the prompts used for the sampled Hendrycks MATH (Level 4) experiments. For Commonsense170k, we use the same overall structure with minor modifications to the task instruction and answer format, as described in Section B.3.

B.1 Scoring Prompt (ϕ_i)

The scoring prompt instructs $\mathcal{M}_i$ to independently solve the query x, compare its own reasoning against $r_j^{(t)}$, and finally output a calibrated confidence score in $[0, 1]$. This procedure makes $s_{i\to j}^{(t)}$ a receiver-side evaluation: the score depends solely on $\mathcal{M}_i$'s internal reasoning applied to the content of $r_j^{(t)}$, without relying on any confidence reported by agent j.

```
System: You are an expert evaluator.
Respond using only a single number.

User:
You are evaluating the correctness of a
proposed answer to a mathematical reasoning
problem.

Assume that up to f responses may be
malicious or unreliable.

Reason carefully before producing the final
score.

Problem: {x}

Proposed answer: {r_j^(t)}

Step 1: Solve the problem independently step
by step.
Step 2: Evaluate the proposed answer.
Step 3: Output ONLY a single decimal number
between 0.0 and 1.0.
0.0 = clearly incorrect, 0.5 = uncertain,
1.0 = clearly correct.
Score:
```

The first numeric token in $\mathcal{M}_i$'s response is extracted using the regular expression `([0-9]+(?:\.[0-9]+)?)` and clipped to the interval $[0, 1]$. If parsing fails or the LLM returns an invalid response, the score defaults to $s_{i\to j}^{(t)} = 0.5$, treating the response as maximally uncertain. The self-score $s_{i\to i}^{(t)} = \phi_i(x, r_i^{(t)})$ is computed using the same prompt with $r_j^{(t)}$ replaced by $r_i^{(t)}$.

B.2 Refine Prompt (Eq. (5))

The refine prompt provides $\mathcal{M}_i$ with its current response $r_i^{(t)}$ together with the retained neighbor responses $\{(r_j^{(t)}, s_{i\to j}^{(t)})\}_{j\in\mathcal{R}_i^{(t)}}$, sorted in descend-

ing order of reliability score. The instruction *"Prefer answers with higher reliability scores, but retain your current answer if you believe it is correct"* serves as the natural-language realization of Eq. (5), encouraging $\mathcal{M}_i$ to prioritize highly scored neighbor responses while preserving its own response as an anchor when appropriate.

> **System:** You are a helpful assistant. Follow the required output format exactly.
>
> **User:**
> You are solving a mathematical reasoning problem. Review the information below and provide your best answer.
>
> Assume that up to f responses may be malicious or unreliable.
>
> Reason carefully before producing the final answer.
>
> Problem: $\{x\}$
>
> Your current answer: $\{r_i^{(t)}\}$
>
> Other agents' answers with reliability scores (higher indicates greater reliability):
> - Answer: $\{r_{j_1}^{(t)}\}$ (reliability score: $\{s_{i\to j_1}^{(t)}\}$)
> - Answer: $\{r_{j_2}^{(t)}\}$ (reliability score: $\{s_{i\to j_2}^{(t)}\}$)
> . . . (sorted in descending order of score, $j_k \in \mathcal{R}_i^{(t)}$)
>
> Prefer answers with higher reliability scores, but retain your current answer if you believe it is correct.
> Respond using the format:
> Answer: [your numerical answer]

Reliability scores are formatted to two decimal places. The retained set $\mathcal{R}_i^{(t)}$ is computed prior to rendering the prompt, so neighbors removed by $\mathrm{bottom}_F(\cdot)$ never appear in the refinement stage. If $\mathcal{R}_i^{(t)} = \emptyset$, the refinement step is skipped and $r_i^{(t+1)} = r_i^{(t)}$. The final numerical answer is extracted using a dataset-specific parser; if extraction fails, the update again defaults to $r_i^{(t+1)} = r_i^{(t)}$.

B.3 Commonsense170k Prompt Adaptation

Since the sampled Commonsense170k instances span heterogeneous formats, including true/false, multiple-choice, and sentence-completion tasks, we replace the math-specific wording with a question-agnostic formulation. In particular, "mathematical reasoning problem" is replaced with "question", and the answer-format instruction becomes `Respond strictly using the answer format specified in the question.\n Answer: [your answer]`. The receiver-side scoring procedure and descending-score neighbor presentation remain unchanged.

B.4 Byzantine Agent Behavior

Adversarial Byzantine agents (Section 5) bypass both prompting procedures. At each round t, they (i) broadcast a cached weak-model response generated from an out-of-distribution query, and (ii) report a falsified self-confidence score of 1.0 when interacting with confidence-based baselines such as CP-WBFT (Zheng et al., 2026). This simulates a strong dishonest-confidence attack while remaining indistinguishable from honest agents at the message level. In contrast, faulty Byzantine agents follow the same prompting pipeline as reliable agents, and produce unreliable outputs solely due to the limited capability of the underlying weak model.

C Experimental Setup

C.1 Dataset for Closed-source LLMs

We list the sampled problems used in the closed-source GPT API experiments reported in Tables 1 and 2, including 100 sampled Hendrycks MATH (Level 4) instances and 30 sampled Commonsense170k instances. Each row reports the dataset-specific question identifier (`qid`), an abbreviated version of the question, and the ground-truth answer $y^\star$. Full problem statements can be retrieved using the corresponding `qid` from the original public releases.

C.1.1 Hendrycks MATH (Level 4)

The 100 sampled instances span the seven question categories present in Hendrycks MATH Level 4. We group rows by category for readability. These problems correspond to the Hendrycks MATH experiments in Tables 1 and 2, where `gpt-4o-mini` and `gpt-3.5-turbo` are used as the strong and weak models, respectively.

C.1.2 Commonsense170k

The 30 sampled Commonsense170k instances span four question formats: true/false, two-option pronoun resolution, multiple-choice reasoning, and sentence completion. We group the examples by format. For non-true/false questions, the answer choices are listed below the question statement, and the ground-truth label corresponds to the option identifier in the original dataset. These problems correspond to the Commonsense170k experiments in Tables 1 and 2, where `gpt-5` and `gpt-4o` are used as the strong and weak models, respectively.

qid	Question	$y^\star$
Algebra		
32448964	Value of K for which $6x + 4y = 7,\ Kx + 8y = 7$ has no solution	12
57cbe0b7	Solve $(\sqrt{12x} + 12)(\sqrt{3x} - 6) = 4(x + 3) + x - 34$	50
984a9fb8	Express $g^4 + 12g^2 + 9 = c(g^2 + p)^2 + q$; find q	-27
b0969f24	Piecewise f with $f(-4) = -60/13$, $f(4) = 3120$; find $a + b$	28
d026abb8	Integer x in arithmetic sequence $3^2,\ x,\ 3^4$	45
9bb18dfb	Minimum of $\lvert x - 1\rvert + \lvert x - 1.5\rvert + \lvert x - 2\rvert$ over $x \in \mathbb{R}$	1
Intermediate Algebra		
40c817ff	$\log_y x + \log_x y = 7$; find $(\log_y x)^2 + (\log_x y)^2$	47
b3f26b98	Piecewise f invertible; find k	1
b11209a5	Geometric sequence with $a_5 - a_4 = 576$, $a_2 - a_1 = 9$; sum $\sum_{i=1}^{5} a_i$	1023
10aef5c7	Count of $n < 1000$ such that $\lfloor \log_2 n \rfloor$ is a positive even integer	340
Prealgebra		
2de720d4	Number of times the digit 6 appears in integers from 1 to 100	20
b6a36467	Smallest integer > 2 that leaves remainder 2 modulo 3, 4, 5, 6	62
34e64136	Five tests scored 87, 85, 87; last two differ by 3 with average 90; find max	97
f3b85d7a	Number of even perfect cubes less than 2008	6
c740506c	Box nesting: 4 large $\times$ 3 medium $\times$ 2 small; total number of boxes	40
e067503f	Unit conversion: 6 wallops = 5 ballops, 3 ballops = 11 fallops; wallops for 110 fallops	36
7bfcd56a	First odd year after 2006 whose digits split into 3-digit and 1-digit groups with common factor > 1	2013
Number Theory		
a10973bf	Smallest integer with exactly 16 divisors, including 12 and 15	120
27b01b01	Base-7 cryptarithm $\overline{AB}_7 + \overline{BA}_7 = \overline{AA0}_7$; product $A \cdot B$	6
583c9eaf	Number of even positive divisors of 252	12
37bab629	Count of $n \in [1, 29]$ for which $n/30$ has a repeating decimal expansion	20
Counting & Probability		
0a3e457d	20-member club, 3 distinct officers, with constraint Alex serves only if Bob does not	6732
d790474f	Jar with 4 red, 2 white marbles; swap-and-sample procedure; $P(\text{red}) = 11/18$	$11/18$
b98d41f7	Coefficient of x^2y^2 in $(x + y)^4 + (x + 2y)^4$	30
03694fa9	120 triangles formed by n vertices on a base; find n	16
33b5135e	10-member chess club, 900 games played; find N games per pair	20
e5787bf4	Distinct bracelets with 5 distinct beads under rotation and reflection equivalence	12
Geometry		
1efe044e	Maximum volume (cm^3) when right $\triangle$ with legs 3, 4 rotated about a leg	50
de4ec0fd	Isosceles $\triangle$ with $AB = AC = 14$, $BC = 26$; shortest angle bisector	$8\sqrt{33}/3$
Precalculus		
16be6141	$\mathbf{v} = \text{proj}_{\mathbf{a}}\mathbf{v} + \text{proj}_{\mathbf{b}}\mathbf{v}$ for all $\mathbf{v}$; find $\mathbf{a} \cdot \mathbf{b}$	0

Table 7: Sampled Hendrycks MATH Level 4 problems used in our experiments, grouped by category. The table shows 30 representative examples from the 100 sampled instances used throughout the evaluation. These problems are additionally used in the ablation experiments reported in Tables 5 and 6. The question column gives an abbreviated statement; full problem text is recoverable from the public Hendrycks MATH release via the qid.

qid	Question (and choices, where applicable)	$y^\star$
True/False		
50816f9e	Is there a season 3 of *Wrecked*?	true
4ba7682f	Does Las Vegas have a professional football team?	true
17f50271	Can a person be jailed for civil contempt of court?	true
4d7cce65	Has Maroon 5 ever performed at a Super Bowl?	true
84e37998	Is *I Know Why the Caged Bird Sings* a memoir?	true
4bb2e906	Is *Varsity Blues* based on a true story?	false
433f24bc	Does it count if you hit the backboard on a free throw?	true
b4c59eee	Can you remove the venom glands from a snake?	true
0c7abae6	Is the Statue of Liberty in New Jersey?	false
a461f8a8	Can you send a letter without a return address?	true
f448271b	Is it illegal to pass on a solid yellow line?	false
28225993	Can you score an own goal from a direct free kick?	true
90b6886c	Do red, yellow, and orange peppers taste different?	true
a03e3147	Is there a fifth season of *Mom*?	true
07250c0c	Are *The Five Heartbeats* based on a real group?	false
Two-option pronoun resolution		
263ebb80	Embroidery: when threading the needle, the _ was too thick. *Choices:* (option1) needle; (option2) floss.	option2
c3a49074	"Felicia asked Katrina about new technology because _ was interested." *Choices:* (option1) Felicia; (option2) Katrina.	option1
60091082	Movers wanted to store the boxes in the offices, but the _ were too small. *Choices:* (option1) offices; (option2) boxes.	option1
2e8510fe	Elena loves to read books but Jessica does not; _ bought videos all the time. *Choices:* (option1) Elena; (option2) Jessica.	option2
08a09099	The public elected Jason over Randy, because _ delivered a less persuasive speech. *Choices:* (option1) Jason; (option2) Randy.	option2
Multiple choice		
11aa364a	Addison spent a month of lunches and finally found Carson a place; how would you describe Addison? *Choices:* (answer1) helpful; (answer2) meanspirited; (answer3) selfish.	answer1
bd59d9db	Austin got a PS4 Pro and bought a new TV; what will Austin want to do next? *Choices:* (answer1) play his new game console; (answer2) return the TV; (answer3) go watch a movie at the theatre.	answer1
2ce5f4f0	From which part of the plant does a bee get food? *Choices:* (answer1) flower; (answer2) seed; (answer3) stem; (answer4) root.	answer1
612a4066	Remy came up behind Jan and pushed her in the back very hard; how would you describe Remy? *Choices:* (answer1) physical; (answer2) ready to fight Jan; (answer3) very hostile towards Jan.	answer3
0cd100f4	The best way to improve future production yields on the farm is… *Choices:* (answer1) planting cabbage one year and spinach the next; (answer2) chemical fertilizers and salts; (answer3) rotating water schedules daily; (answer4) over-watering each field.	answer1
25040642	How do you treat period cramps? *Choices:* (solution1) drink some coffee; (solution2) take some Midol.	solution2
ee3abe29	Paper towel: what is it good for? *Choices:* (solution1) clean telescope; (solution2) operate telescope.	solution1
Sentence completion		
3abb847a	"How to start a giving fund" — after opening a bank account, what naturally follows? *Choices:* (ending1) choose checking vs. savings depending on payment frequency; (ending2) a separate account protects you from debt; (ending3) set up a line for the bank's payout the day before your event; (ending4) it isn't complicated — consult an individual bank.	ending1
873e845e	Javelin throw scene: as the man releases the javelin, in the background… *Choices:* (ending1) bleachers in the background, hot and sunny day; (ending2) large crowd standing around, kites flying; (ending3) a big splash, the man runs to rescue it; (ending4) another man in a black t-shirt easily catches it.	ending1
e47d4d28	Drumming and piano duet: a child drums while a woman plays piano along; they… *Choices:* (ending1) continue playing the drums and the music; (ending2) have a small audience watching them perform; (ending3) play till there is no longer a fist drumming in the background; (ending4) play and sing along intently for joy.	ending2

Table 8: Sampled Commonsense170k problems used in our experiments, grouped by format. These problems correspond to the closed-source commonsense reasoning experiments reported in Tables 1 and 2, where `gpt-5` and `gpt-4o` are used as the strong and weak models, respectively. Each instance preserves its native answer format, and ground-truth answers are reported using the corresponding label as released. For multi-option items, we list the choices in the question column.

C.2 Dataset for Open-weight LLMs

For the open-weight LLM experiments, we evaluate on five commonsense reasoning benchmarks: ARC-Challenge, HellaSwag, BoolQ, OpenBookQA, and RTE. From each benchmark, we randomly sample 50 questions, resulting in a total of 250 evaluation instances used throughout the experiments. These benchmarks cover a range of reasoning settings, including commonsense inference, scientific reasoning, natural language understanding, and multiple-choice question answering. All experiments use Qwen3-4B as the strong model and Qwen2.5-1.5B-Instruct as the weak model.

D Additional Experimental Results

In this section, we report benchmark-level and per-round results for the open-weight LLM experiments summarized in Section 5.3.2. Tables 9 and 10 expand the averaged results reported in Tables 3 and 4.

Per-benchmark robustness. Across all five benchmarks and all evaluated topologies, SAC achieves non-negative BFTI on four of the five benchmarks, with improvements of up to $+10.3\%$ on OpenBookQA and consistent gains on ARC-Challenge, HellaSwag, and BoolQ. On RTE, where the initial strong-agent accuracy is already relatively high (S IAA $\approx 84\%$), SAC still maintains H-Majority around 82% across all topologies while avoiding the large degradation observed in CP-WBFT. By contrast, CP-WBFT consistently produces strongly negative BFTI across benchmarks and topologies, with particularly severe degradation on HellaSwag and OpenBookQA. The largest gap appears on OpenBookQA under the Complete topology, where CP-WBFT yields strongly negative BFTI while SAC achieves over $+10\%$ improvement. Overall, these results suggest that SAC's receiver-side evaluation generalizes robustly across benchmarks with substantially different reasoning formats and difficulty levels.

Per-group behavior. The per-group accuracies reported in Table 9 show that SAC consistently preserves or improves strong-agent accuracy while broadly maintaining weak-agent performance. Representative examples include $47 \rightarrow 62\%$ on HellaSwag with Erdős–Rényi, $60 \rightarrow 77\%$ on OpenBookQA with MERG, and $72 \rightarrow 78\%$ on ARC-Challenge with MERG. Under CP-WBFT, the corresponding strong-group accuracies collapse to substantially lower values, often near random-guessing performance on the harder multiple-choice benchmarks. Weak-agent accuracy under SAC is also largely preserved, with several settings showing modest gains (e.g., OpenBookQA and BoolQ), whereas CP-WBFT frequently degrades both weak and strong groups simultaneously. H-Majority follows the same trend: SAC consistently maintains substantially higher honest-majority correctness across all benchmarks and topologies, while CP-WBFT often collapses despite the honest agents forming a clear majority in the network.

Per-round dynamics. Table 10 reports per-round weak/strong accuracies. Under SAC, strong-agent accuracy typically improves within the first one or two communication rounds and then remains stable through round 6. For example, strong-agent accuracy increases from $49.0 \rightarrow 60.0\%$ on HellaSwag with MERG and from $60.0 \rightarrow 79.0\%$ on OpenBookQA with MERG. Weak-agent accuracy is largely preserved throughout the refinement process, with only small fluctuations across rounds. On BoolQ, strong-agent accuracy oscillates between roughly 83% and 89% across rounds in a pattern that appears consistently across topologies, likely reflecting the binary nature of the task. In contrast, CP-WBFT collapses immediately after the first communication round and exhibits little recovery thereafter, so we aggregate rounds 1–6 into a single row. Averaged across benchmarks and topologies, SAC improves final-round strong-agent accuracy by roughly 30 percentage points compared to CP-WBFT.

Effect of topology. SAC exhibits relatively small performance variation across topologies within the same benchmark. For example, FAA differs by only a few percentage points across topologies on ARC-Challenge, HellaSwag, and BoolQ. In contrast, CP-WBFT shows substantially larger instability depending on the graph structure, particularly on HellaSwag and OpenBookQA, where FAA varies dramatically across topologies while remaining consistently poor overall. This relatively small across-topology variance of SAC is consistent with our theoretical analysis, where the qualitative robustness of SAC primarily depends on satisfying the $(F+1)$-robustness condition rather than the specific topology construction.

Dataset	Method	Topology	IAA	FAA	BFTI	RA	W IAA→FAA	S IAA→FAA	H-Maj.
ARC-Challenge	CP-WBFT	MERG	58.0	47.4	−10.6	46.0	58→60	72→53	46.0
		Complete	58.0	37.7	−20.3	44.0	58→44	72→44	44.0
		Erdős–Rényi	58.0	39.4	−18.6	46.0	58→44	72→47	46.0
	SAC (Ours)	MERG	58.0	60.6	+2.6	82.0	58→57	72→78	82.0
		Complete	59.1	62.9	+3.7	82.0	58→59	74→80	82.0
		Erdős–Rényi	58.0	59.1	+1.1	74.0	58→57	72→75	74.0
HellaSwag	CP-WBFT	MERG	42.0	21.7	−20.3	16.0	52→31	48→22	16.0
		Complete	43.1	13.7	−29.4	16.0	52→16	50→16	16.0
		Erdős–Rényi	43.4	14.0	−29.4	16.0	53→17	50→16	16.0
	SAC (Ours)	MERG	43.1	48.6	+5.4	60.0	53→50	49→60	60.0
		Complete	42.0	48.0	+6.0	58.0	53→51	47→58	58.0
		Erdős–Rényi	41.7	49.4	+7.7	62.0	52→49	47→62	62.0
BoolQ	CP-WBFT	MERG	71.7	55.7	−16.0	58.0	83→68	84→64	58.0
		Complete	71.4	48.0	−23.4	56.0	83→56	84→56	56.0
		Erdős–Rényi	71.4	46.3	−25.1	56.0	83→53	84→54	56.0
	SAC (Ours)	MERG	72.3	73.4	+1.1	90.0	83→78	85→90	92.0
		Complete	70.9	73.1	+2.3	88.0	83→79	82→88	88.0
		Erdős–Rényi	71.4	72.9	+1.4	88.0	83→79	84→88	90.0
OpenBookQA	CP-WBFT	MERG	54.0	31.4	−22.6	24.0	66→43	62→34	24.0
		Complete	54.0	18.9	−35.1	22.0	66→22	62→22	22.0
		Erdős–Rényi	54.0	20.6	−33.4	24.0	66→23	62→24	24.0
	SAC (Ours)	MERG	53.1	63.4	+10.3	76.0	66→68	60→77	76.0
		Complete	54.0	64.3	+10.3	78.0	66→70	62→78	78.0
		Erdős–Rényi	54.0	64.0	+10.0	78.0	66→69	62→78	78.0
RTE	CP-WBFT	MERG	69.1	56.3	−12.9	56.0	74→71	84→63	56.0
		Complete	69.1	48.0	−21.1	56.0	74→56	84→56	56.0
		Erdős–Rényi	69.1	45.1	−24.0	56.0	74→51	84→54	56.0
	SAC (Ours)	MERG	69.1	65.7	−3.4	82.0	74→68	84→81	82.0
		Complete	69.1	66.3	−2.9	80.0	74→71	84→80	82.0
		Erdős–Rényi	69.7	66.9	−2.9	82.0	74→69	85→82	82.0

Table 9: Full benchmark-level results corresponding to the averaged open-weight LLM results reported in Table 3. We report per-dataset comparisons of CP-WBFT and SAC across three $(F+1)$-robust communication topologies on five commonsense reasoning benchmarks (50 questions each). Each network contains $n=7$ agents (4 strong honest, 2 weak honest, 1 adversarial Byzantine, $F=3$, $r=4$). Qwen3-4B and Qwen2.5-1.5B-Instruct are used as the strong and weak models, respectively. W IAA→FAA and S IAA→FAA denote per-agent accuracy of the weak and strong honest groups before and after consensus; H-Majority is the fraction of queries for which the majority answer among the honest agents matches the ground truth at the final round.

Dataset	Method	Round	MERG (W/S)	Complete (W/S)	Erdős–Rényi (W/S)
ARC-C	CP-WBFT	Init	58.0 / 72.5	58.0 / 72.5	58.0 / 72.5
		Rnd 1-6	60.0 / 53.0	44.0 / 44.0	44.0 / 47.0
	SAC (Ours)	Init	58.0 / 72.5	58.0 / 74.5	58.0 / 72.5
		Rnd 1	57.0 / 78.0	57.0 / 81.0	57.0 / 78.0
		Rnd 2	58.0 / 79.5	58.0 / 80.5	57.0 / 76.0
		Rnd 3	58.0 / 79.0	57.0 / 80.5	57.0 / 76.0
		Rnd 4	58.0 / 78.5	59.0 / 80.5	57.0 / 75.0
		Rnd 5	58.0 / 77.0	57.0 / 80.5	57.0 / 74.5
		Rnd 6	57.0 / 77.5	59.0 / 80.5	57.0 / 75.0
HellaSwag	CP-WBFT	Init	52.0 / 47.5	52.0 / 49.5	53.0 / 49.5
		Rnd 1-6	31.0 / 22.5	16.0 / 16.0	17.0 / 16.0
	SAC (Ours)	Init	53.0 / 49.0	53.0 / 47.0	52.0 / 47.0
		Rnd 1	50.0 / 58.5	51.0 / 56.5	49.0 / 57.5
		Rnd 2	50.0 / 60.0	51.0 / 58.5	49.0 / 61.0
		Rnd 3	50.0 / 60.0	51.0 / 58.5	49.0 / 62.5
		Rnd 4	50.0 / 60.0	51.0 / 58.5	49.0 / 62.0
		Rnd 5	50.0 / 60.0	51.0 / 58.5	49.0 / 62.0
		Rnd 6	50.0 / 60.0	51.0 / 58.5	49.0 / 62.0
BoolQ	CP-WBFT	Init	83.0 / 84.0	83.0 / 83.5	83.0 / 83.5
		Rnd 1-6	68.0 / 63.5	56.0 / 56.0	53.0 / 54.5
	SAC (Ours)	Init	83.0 / 85.0	83.0 / 82.5	83.0 / 83.5
		Rnd 1	67.0 / 79.0	61.0 / 79.5	64.0 / 80.0
		Rnd 2	79.0 / 89.5	76.0 / 89.5	80.0 / 87.5
		Rnd 3	70.0 / 83.5	64.0 / 83.5	65.0 / 85.5
		Rnd 4	79.0 / 88.0	79.0 / 89.5	80.0 / 86.5
		Rnd 5	70.0 / 83.0	66.0 / 85.5	66.0 / 83.0
		Rnd 6	78.0 / 89.5	79.0 / 88.5	79.0 / 88.0
OBQA	CP-WBFT	Init	66.0 / 61.5	66.0 / 61.5	66.0 / 61.5
		Rnd 1-6	43.0 / 33.5	22.0 / 22.0	23.0 / 24.5
	SAC (Ours)	Init	66.0 / 60.0	66.0 / 61.5	66.0 / 61.5
		Rnd 1	66.0 / 81.0	68.0 / 79.5	65.0 / 74.0
		Rnd 2	68.0 / 79.0	70.0 / 79.0	68.0 / 75.5
		Rnd 3	68.0 / 78.0	70.0 / 77.5	69.0 / 76.0
		Rnd 4	68.0 / 77.5	70.0 / 77.5	69.0 / 77.0
		Rnd 5	68.0 / 77.5	70.0 / 77.5	69.0 / 77.5
		Rnd 6	68.0 / 77.0	70.0 / 77.5	69.0 / 77.5
RTE	CP-WBFT	Init	74.0 / 84.0	74.0 / 84.0	74.0 / 84.0
		Rnd 1-6	71.0 / 63.0	56.0 / 56.0	51.0 / 53.5
	SAC (Ours)	Init	74.0 / 84.0	74.0 / 84.0	74.0 / 85.0
		Rnd 1	68.0 / 74.5	71.0 / 77.5	71.0 / 79.5
		Rnd 2	69.0 / 80.0	71.0 / 80.5	70.0 / 80.0
		Rnd 3	68.0 / 78.0	71.0 / 78.0	69.0 / 81.5
		Rnd 4	68.0 / 80.0	71.0 / 80.5	69.0 / 82.0
		Rnd 5	69.0 / 78.5	71.0 / 77.0	69.0 / 81.0
		Rnd 6	68.0 / 81.0	71.0 / 80.5	69.0 / 82.5

Table 10: Benchmark-level per-round results corresponding to the averaged open-weight LLM results reported in Table 4. We report per-round weak / strong honest accuracy (W/S, in %) across three $(F{+}1)$-robust communication topologies on five commonsense reasoning benchmarks (50 questions each). Each cell shows the average accuracy of the weak honest group (Qwen2.5-1.5B-Instruct, 2 agents) and the strong honest group (Qwen3-4B, 4 agents) at the corresponding round under $n{=}7$ agents and $F{=}3$.

E Licenses and Terms of Use

All datasets and models used in this work are publicly available and used in compliance with their respective licenses and terms of use. The Hendrycks MATH dataset (Hendrycks et al., 2021) is released under the MIT License. Commonsense170k (Hu et al., 2023) is released under the Apache 2.0 License. ARC-C (Clark et al., 2018) is released under the CC BY-SA 4.0 License. HellaSwag (Zellers et al., 2019) is released under the MIT License. BoolQ (Clark et al., 2019) is released under the CC BY-SA 3.0 License. OBQA (Mihaylov et al., 2018) is released under the Apache 2.0 License. RTE (Dagan et al., 2005) is distributed for research use as part of the PASCAL RTE challenge. The Qwen2.5-1.5B-Instruct and Qwen3-4B models are released under the Apache 2.0 License. The GPT models (gpt-3.5-turbo, gpt-4o-mini, gpt-4o, gpt-5) are accessed via the OpenAI API in compliance with OpenAI's Terms of Use. All artifacts are used solely for non-commercial research purposes, consistent with their intended use.