BORA: Byzantine-Observation Reinforced Attack on Cooperative Multi-Agent UAV Relay Networks

Antonio Di Maio
ETH Zurich
Zurich, Switzerland
dimaio@ethz.ch

Arwin Baur
ETH Zurich
Zurich, Switzerland
arbaur@ethz.ch

***Abstract*—Cooperative multi-agent relay networks can restore ground-user end-to-end wireless connectivity under adversarial jamming by continuously repositioning nodes according to a shared policy. Their reliance on in-band observation exchange, however, exposes an attack surface for internal Byzantine nodes. Existing Byzantine-resilient MARL approaches address training-time gradient poisoning or outsider interference, disregarding runtime falsification of broadcast semantics by insider nodes that already hold legitimate network access. We propose BORA (Byzantine-Observation Reinforced Attack), a learned adversarial policy that falsifies the broadcast state vector of Byzantine nodes to degrade network throughput, trained against the honest nodes' policy. Because every Byzantine node advertises an identical falsified vector to all recipients, its broadcasts stay mutually consistent and conform to the communication protocol, evading consistency-based defenses such as reliable broadcast that detect malicious senders only through per-recipient disagreement. Experiments across a broad range of scenarios show that BORA achieves up to 99.2% global throughput degradation when deployed on a limited fraction of corrupted relay nodes, more than doubling the degradation of the strongest heuristic baseline (47%).**

***Index Terms*—Multi-Agent Systems; Unmanned Aerial Vehicles; Adversarial Training; Mobile Networking.**

I. Introduction

Unmanned Aerial Vehicle (UAV) swarms are increasingly deployed in contested, mission-critical environments where ground communication infrastructure is unavailable or has been destroyed. In adversarial environments, disaster response applications such as search-and-rescue after earthquakes, humanitarian operations in active conflict zones, or tactical coordination under adversarial pressure, can be served through UAV relay swarm which offer a mobile communication backbone connecting ground users to command and control, and to each other. To hold this backbone together under deliberate radio-frequency jamming, proactive relay systems continuously reposition their nodes to stay ahead of mobile interferers, each guided by a shared learned cooperative policy that maximises throughput for ground users. A key enabler of this cooperation is in-band *observation exchange*: every node periodically broadcasts its local state (position, link quality, direction to users) and folds the broadcasts it receives directly into its own observation when deciding where to fly next.

This same cooperative dependence is the swarm's most insidious weakness. While the system is explicitly engineered to withstand *external* adversaries such as radio-frequency jammers, its observation-exchange channel offers no protection against a relay node captured and reprogrammed from within, one that continues to follow the protocol while falsifying the state it broadcasts. By disseminating a manipulated local state, such a *Byzantine actor* drives honest neighbors to systematically misestimate the traffic topology, steering the swarm away from the users it should serve; and because the channel is integral to routing, it cannot be disabled without breaking the coordination the swarm depends on. This internal Byzantine threat, rather than the external jamming the swarm is built to survive, is the focus of this work.

Securing proactive relay systems against such internal misinformation is a distinct challenge from existing approaches. Byzantine-robust Multi-Agent Reinforcement Learning (MARL) focuses on training-time gradient poisoning, and adversarial Reinforcement Learning (RL) addresses observation corruption from outside the system; neither framework models an insider that falsifies the state it legitimately broadcasts to cooperative neighbors at deployment time, exploiting a channel that violates no protocol rule. Several questions therefore remain open: which observation fields can be weaponised, how disruption scales with the number of Byzantine nodes, and whether a learned adversarial policy can exceed the damage of hand-crafted heuristics in this specific threat model.

This work addresses these gaps through a systematic empirical study of Byzantine observation-manipulation attacks on a cooperative UAV relay swarm. The Byzantine agent is trained adversarially against the honest cooperative policy, making the system exploitable by a reward-maximizing adversary.

This work aims to answer the following research questions:

- How can a Byzantine attacker optimally hinder the task of a cooperative relay swarm by manipulating the shared observation-exchange protocol through dissemination of falsified local-state information messages?
- How does the potential disruption of a Byzantine-communication threat change when the attacker employs a learned policy rather than a fixed, heuristic strategy?

This work's main contributions are to:

- Propose and experimentally validate Byzantine-Observation Reinforced Attack (BORA), a learned attack

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

that falsifies the full advertised state vector, trained via RL adversarially against the honest cooperative policy.
- Introduce a large suite of heuristic observation attacks and a noise baseline, demonstrating the limitations of non-data-driven methods in this threat model.
- Show that in cooperative MARL capability and vulnerability are inseparable: the same trusted observation exchange that enables coordination also lets an insider control its neighbours' perception, suggesting that robustness should target the observation channel rather than rely on node redundancy alone.

II. Related Works

Proactive relay placement under jamming. Proactive Data-driven Relocation (PADRE) [1] is the base system studied in this work. It applies Centralized Training, Decentralized Execution (CTDE)-based MARL to relocate UAV relay nodes proactively, solving for maximum-throughput routing via Multi-Commodity Concurrent Maximum Flow (MCCMF) at each step. PADRE assumes all nodes are honest; this work removes that assumption.

Byzantine faults in distributed and multi-agent learning. The Byzantine fault model originates in distributed consensus, where a faulty component may send conflicting information to different peers and reliable agreement is impossible once such faults exceed one third of the participants.The model has since been studied extensively in distributed and federated training, where Byzantine workers corrupt the learned model through manipulated gradient updates and a large body of robust-aggregation defenses has emerged in response [2]. Closer to our setting are Byzantine faults at *execution* time, where cooperating agents must act despite teammates that behave arbitrarily. Recent cooperative-MARL work studies this regime, casting Byzantine peers as adversarial types in a Bayesian game [3], regularizing agents' mutual information to bound the influence of worst-case allies [4], and learning resilient policies in a fully decentralized manner [5]; on the attack side, a single Byzantine agent can unilaterally steer a cooperative majority toward worst-case behavior through ordinary interaction [6]. These works treat the Byzantine agent as a black box emitting arbitrary actions; our threat model is more specific, with a protocol-conforming node that injects a falsified but internally consistent local state into the in-band observation-exchange channel that honest agents aggregate at decision time.

Robustness of cooperative multi-agent systems to communication attacks. A growing body of work studies adversaries that corrupt the observations or messages exchanged between honest agents at runtime. Attacks have been demonstrated against the learned inter-agent communication of cooperative perception systems [7] and against a single agent's observations in cooperative MARL [8], while defenses range from certified robustness to adversarial communication [9] to message reconstruction against an RL-trained message attacker [10] and adversarial training that hardens communication policies against learned channel perturbations [11]. The formulation here differs: the Byzantine node is itself a participant in the swarm and injects false state into the shared in-band observation-exchange channel, rather than perturbing the victim's environment observations from outside.

Adversarial attacks on RL policies. Observation poisoning in single-agent RL perturbs the victim's own perception, either through input-space adversarial examples [12], [13] or by optimizing perturbations against the entire policy distribution [14]. A complementary line studies adversarial policies, where an adversary acting within a shared environment induces natural yet adversarial observations for the victim [15]. This work proposes a different threat model: one agent corrupts *another* agent's information source by broadcasting a falsified local state vector, rather than attacking the victim's sensors directly. Table I positions this work against the surveyed approaches.

TABLE I
Comparison of related work.

Work	B	O	L	M	R
Li et al. (ICLR 2024) [3]	✓			✓	
Li et al. (AAAI 2024) [4]	✓			✓	
Meshkat & Xu [5]	✓			✓	
Li et al. (2025) [6]	✓		✓	✓	
Ma et al. (2025) [11]		✓	✓	✓	
Leuenberger et al. [1]				✓	✓
Duan et al. (2025) [14]		✓			
Gleave et al. [15]			✓		
Sun et al. (2023) [9]		✓		✓	
Lin et al. [8]		✓	✓	✓	
Xue et al. [10]		✓	✓	✓	
This work	✓	✓	✓	✓	✓

B: Byzantine fault model; O: observation/communication attack (runtime); L: RL-trained adversary; M: cooperative MARL; R: relay/UAV network.

III. System Model

A. Network Model

The scenario (Figure 1) consists of a set $\mathcal{V} = \{1, \ldots, N\}$ of N relay nodes operating in a bounded 2D playground. A set of traffic sources and sinks $\mathcal{U}$ (users) generate demand at fixed positions. A set of adversarial jammers $\mathcal{J}$ emit broadband interference; jammers are activated after an initial warm-up phase and relocate according to a cluster strategy targeting traffic concentrations.

Two nodes $i, j \in \mathcal{V}$ form a *directed link* if the received Signal-to-Interference-plus-Noise Ratio (SINR) at j from i exceeds a minimum threshold. Let us define the *link capacity* $c(i, j) \geq 0$ as the Shannon capacity of the corresponding link, where the received power is modeled by the Friis transmission equation. The *network throughput* $\tau \geq 0$ [$\mathrm{Mbit/s}$] is defined as the total data rate delivered from all sending users to all receiving users, computed as the solution of the MCCMF problem on the current network topology [1], and is the primary performance metric of this work.

B. Multi-Agent Reinforcement Learning Environment

Each relay node runs a decentralized Proximal Policy Optimization (PPO) policy trained under the CTDE paradigm [1].

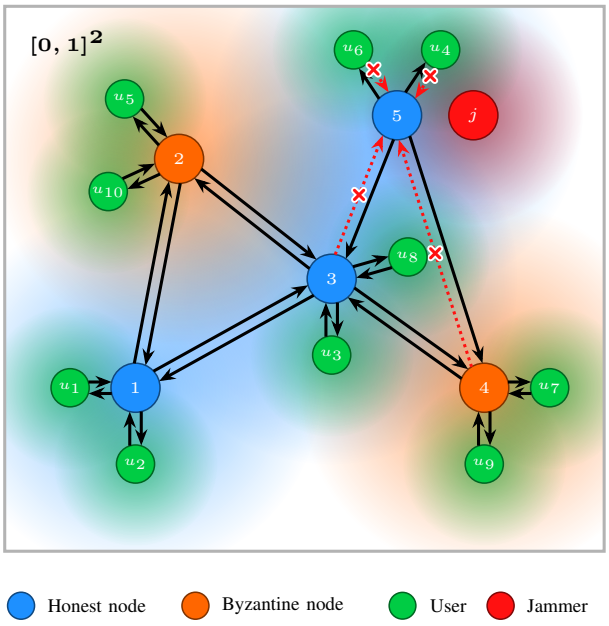


Fig. 1. System model of the scenario with MANET nodes ($f = 2$ Byzantine), users, and a jammer.

At training time, a *centralized critic* modeled by parameters ω observes the full joint state of all nodes; at deployment, each node i acts independently using only its local observation.

Local state. At each time t, each node i maintains four components of local state: (i) $S^{(i)}_{t,\text{user}}$, containing displacement vectors to the closest user, closest sending user, and closest receiving user, together with the interference level at the closest user's location; (ii) $S^{(i)}_{t,\text{netw}}$, containing aggregated in/out link capacities and sensed traffic demands; (iii) $S^{(i)}_{t,\text{topo}}$, containing edge-count summaries; and (iv) $S^{(i)}_{t,\text{phys}}$, containing position, received interference, and transmit power. We denote the normalized direction vectors to the closest user, closest sending user, and closest receiving user by $u_{\text{dis}}(i)$, $s_{\text{dis}}(i)$, and $r_{\text{dis}}(i)$, respectively; the aggregate outgoing/incoming capacities by $C_+(i)$ and $C_-(i)$; the sensed sender/receiver demands by $l_+(i)$ and $l_-(i)$; and the received interference by $I(i)$. Together, the above quantities form the per-node *local state* $S^{(i)}_t \in \mathbb{R}^d$ at time t, with $d = 19$. The *joint state* $S_t = (S^{(1)}_t, \ldots, S^{(N)}_t) \in \mathbb{R}^{Nd}$ collects the states of all relay nodes. Throughout, uppercase symbols (e.g. S_t) denote random variables, whereas the corresponding lowercase symbols (e.g. s_t) denote deterministic realizations obtained from a concrete sampling.

Observation exchange. At each step t, node i transmits its local state $S^{(i)}_t$ through integrity-preserving and authenticated messages over an in-band, multi-hop signaling channel to a *recipient set* $\mathcal{R}_t(i) \subseteq \mathcal{V}$ determined by the *communication protocol* in use. At each step, node i receives a fresh state from every node whose recipient set contains it, namely $\{\, j \in \mathcal{V} \mid i \in \mathcal{R}_t(j) \,\}$. For every other node j, it reuses the most recently received state: if no fresh packet arrives from j at step t, the previously received value is retained until a new one is delivered. The *Age of Information (AoI)* $\Delta_t(i,j) \geq 0$ represents difference between the current time step t and the time step (i.e., *timestamp*) at which node j generated the state received by node i most recently. Therefore, $S^{(j)}_{t-\Delta_t(i,j)}$ denotes the most recently received state of node j at any time t. The *AoI* $\Delta_t(i,j)$ increases by one for every time step since last state reception. Node i's observation $O^{(i)}_t = \big(S^{(j)}_{t-\Delta_t(i,j)}, \Delta_t(i,j)\big)_{j\in\mathcal{V}} \in \mathbb{R}^{N(d+1)}$ is a vector containing the most recently received states of all N nodes and their AoIs.

Action and reward. The action of each node i is a 2D velocity vector $a^{(i)} \in [-\kappa, \kappa]^2$ added to its current position at every step, where κ bounds the per-step displacement, reflecting the physical speed constraints of UAV hardware. We write $a_t = (a^{(1)}_t, \ldots, a^{(N)}_t)$ for the joint action at step t. The shared reward is the global network throughput $\tau_t[\text{bit/s}]$ at each step.

C. Byzantine Threat Model

We assume that honest nodes move according to an *honest movement policy* $\pi_{\text{M}}(\cdot;\theta^*)$, trained without Byzantine nodes and remains fixed at deployment. We assume the attacker obtains such honest movement policy by compromising a deployed relay node, acquiring $\pi_{\text{M}}(\cdot;\theta^*)$ without access to the centralized training critic. After acquiring $\pi_{\text{M}}(\cdot;\theta^*)$, the attacker trains the *Byzantine communication policy* $\pi_{\text{C}}(\cdot;\phi)$ offline in a centralized, local simulation of the target environment against the honest policy. Then, the attacker carries out the attack at decentralized execution stage by deploying the trained Byzantine communication policy $\pi_{\text{C}}(\cdot;\phi^*)$ on some compromised honest nodes (i.e., victims) in physical target environment. To focus our study on the disruption potential of Byzantine-observation manipulation only, we assume that Byzantine nodes move according to the honest movement policy $\pi_{\text{M}}(\cdot;\theta^*)$, so that the observed throughput degradation is a communication-only lower bound that could be further increased by also learning the Byzantine movement policy adversarially.

Let $\mathcal{H} = \{1, \ldots, N-f\}$ and $\mathcal{B} = \{N-f+1, \ldots, N\}$ partition $\mathcal{V}$ into the $N-f$ honest nodes and the f *Byzantine nodes*, respectively. A Byzantine node $b \in \mathcal{B}$ may broadcast to an an arbitrary recipient set $\mathcal{R}_t(b) \subseteq \mathcal{V}$, instead of its local state $S^{(b)}_t$, an arbitrary *falsified state* $\tilde{S}^{(b)}_t$ at each step. Which downstream observations carry falsified rather than honest local states is governed by $\mathbb{P}(o \mid s, \tilde{s})$: a legitimate node's observation $O^{(i)}_t$ contains the falsified state $\tilde{S}^{(b)}_t$ in the slots corresponding to the Byzantine nodes in its recipient set $\mathcal{R}_t(i)$ and honest states elsewhere. Because a Byzantine node advertises a single falsified state $\tilde{S}^{(b)}_t$ to its recipient set $\mathcal{R}_t(b)$, and this same advertised value populates its own observation slot, its observation also carries $\tilde{S}^{(b)}_t$ in place of its true state. Self-falsification does not affect the attack: Byzantine nodes move under the trained honest policy $\pi_{\text{M}}(\cdot;\theta^*)$ and therefore cannot exploit, nor

need to shield, their own observation slot. Let us define the *joint falsified state* $\tilde{S} = (\tilde{S}^{(N-f+1)}, \ldots, \tilde{S}^{(N)}) \in \mathbb{R}^{fd}$. A Byzantine node $b \in \mathcal{B}$ observes the most recently received state of every node together with its AoI, with honest senders $j \in \mathcal{H}$ contributing their true state $S^{(j)}$ and Byzantine senders $j \in \mathcal{B}$ their falsified state $\tilde{S}^{(j)}$. Therefore, under this threat model, the honest nodes' observation becomes $O_t^{(i)} = \left(S^{(j)}_{t-\Delta_t(i,j)}, \Delta_t(i,j)\right)_{j\in\mathcal{H}}, \left(\tilde{S}^{(j)}_{t-\tilde{\Delta}_t(i,j)}, \tilde{\Delta}_t(i,j)\right)_{j\in\mathcal{B}} \in \mathbb{R}^{N(d+1)}, \forall i \in \mathcal{H}$. Denoting $\tilde{\Delta}_t(i,j)$ as the falsified states' AoI, the Byzantine nodes' communication-policy observation $\forall i \in \mathcal{B}$ is $O_t^{(i)} = \Big[\left(S^{(j)}_{t-\Delta_t(i,j)}, \Delta_t(i,j)\right)_{j\in\mathcal{H}}, \left(\tilde{S}^{(j)}_{t-\tilde{\Delta}_t(i,j)}, \tilde{\Delta}_t(i,j)\right)_{j\in\mathcal{B}}, \left(S^{(j)}_{t-\Delta_t(i,j)}, \Delta_t(i,j)\right)_{j\in\mathcal{B}}\Big] \in \mathbb{R}^{(N+f)(d+1)}$.

We assume that each Byzantine node broadcasts its falsified state $\tilde{S}^{(b)}$ identically to all nodes in its recipient set $\mathcal{R}_t(b)$. Preventing any Byzantine nodes from sending different local falsified states to different recipients by design defeats reliable-broadcast-based countermeasures, which require per-recipient message inconsistency to detect a malicious sender. We consider a scenario in which strong Byzantine-resilient routing countermeasures are already in place, making traffic-layer manipulation immediately detectable and therefore unattractive to the attacker. At the observation-exchange layer, however, no malicious-state detection mechanism exists: *legitimate nodes* $i \in \mathcal{H}$ do not verify the honesty of any received state $\tilde{S}^{(b)}$ and integrate it into their observation $O^{(i)}$ without modification.

IV. BORA: Byzantine-Observation Reinforced Attack

We propose Byzantine-Observation Reinforced Attack (BORA)[1], a learned communication-based attacker adversarially trained against the optimal honest cooperative movement policies with parameters θ^*. The key insight is that the honest policy, once trained, defines a stationary deterministic mapping from observations to actions, making it a fixed, exploitable opponent against which a reward-maximizing adversary can be trained. Figure 2 presents the BORA training architecture, in which the Byzantine actors and their shared critic are trained adversarially against this fixed honest policy.

State and action space. Byzantine nodes participate in the in-band broadcast channel under the same communication protocol as honest nodes, receiving the broadcast state vectors $S^{(j)}$ sent to them. Our evaluations assume lossless delivery: every broadcast reaches the recipients in range within the step it is sent, so the measured degradation reflects the falsified state content and the protocol's communication reach rather than unreliable-channel effects such as packet loss. The subset of received broadcasts that a Byzantine policy conditions on is determined by the communication protocol. At each step, BORA samples a *state offset* $\delta^{(b)} \in \mathbb{R}^d$ from the learned policy, added to the full broadcast state vector $S^{(b)}$ of the

[1]Implementation, data, and trained models available at `https://github.com/ediamondlabs/BORA`

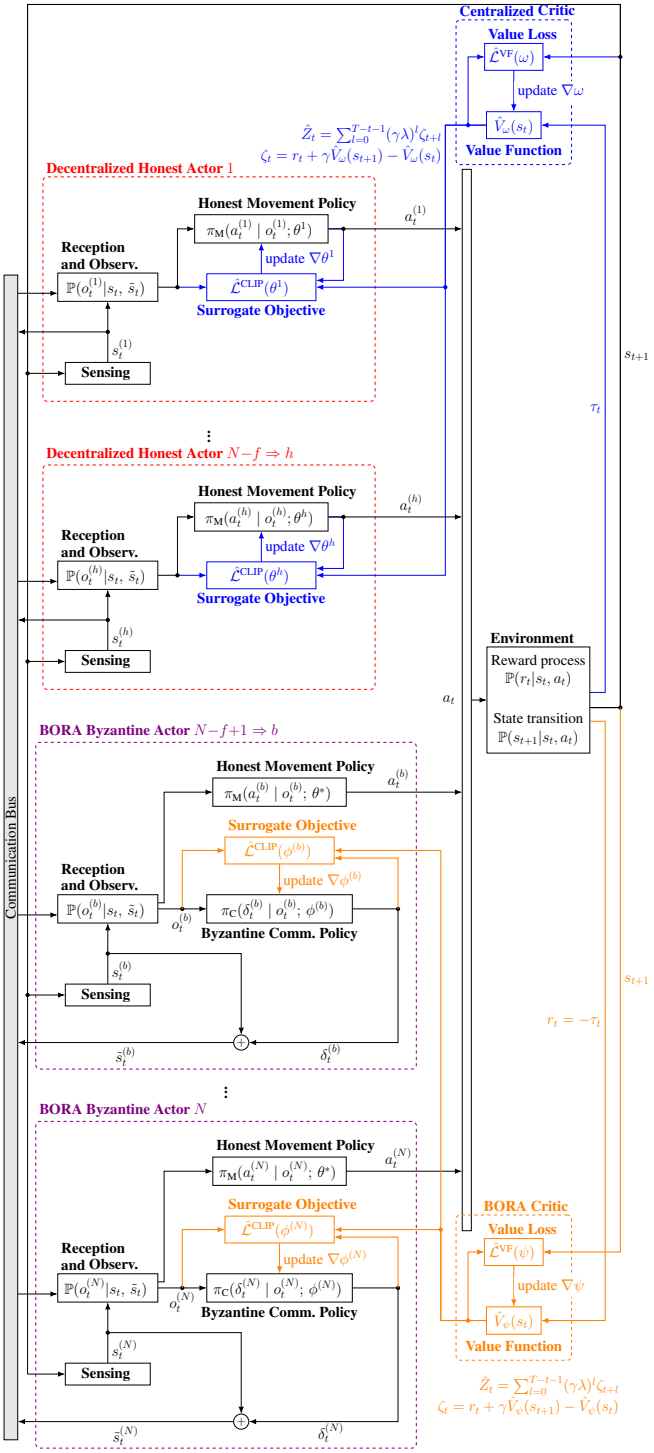


Fig. 2. Two-stage BORA architecture. Honest nodes' centralized cooperative policy pre-training in blue, for reference, followed by the Byzantine nodes' centralized adversarial policy training (BORA) in orange.

Byzantine node. The offset is learned rather than fixed, enabling time-adaptive perturbations that respond to the evolving network state.

Reward. The Byzantine agent receives reward $r_t = -\tau_t$, the negated throughput at each step, directly opposing the honest

policy's cooperative objective.

Decentralized execution. At deployment phase, the weights θ^* and ϕ^* are not updated, and no critic, reward, or parameter update remains (Algorithm 1). Each node acts from its own local observation with no central coordinator: a Byzantine node b maps its state to an offset $\delta_t^{(b)} = \pi_{\mathrm{C}}(o_t^{(b)}; \phi^*)$ and broadcasts the falsified $\tilde{S}_t^{(b)}$, while honest nodes broadcast their true state $S_t^{(i)}$. All nodes, honest and Byzantine alike, then select their movement simultaneously through the same trained honest policy $\pi_{\mathrm{M}}(\cdot; \theta^*)$. The execution stage is therefore a simultaneous-move interaction, in contrast with the policy-level Stackelberg game solved during training, where the attacker best-responds to the fixed honest policy.

Algorithm 1 defines the per-step `Step` routine (reused by the training rollout of Algorithm 2) and applies it in a loop; one such step proceeds as follows. Each node first carries over its previous observation and refreshes only the slots it receives this step (line 29). Honest nodes broadcast their true state to their recipient sets (line 8), updating the corresponding slots of every receiving node. Each Byzantine node then maps its observation to an offset $\delta_t^{(b)} = \pi_{\mathrm{C}}(o_t^{(b)}; \phi^*)$ (line 13), adds it to its true state to form the falsified broadcast (line 14), and sends it to its recipient set (line 15), updating the slots of the honest recipients. All nodes then move simultaneously under the trained honest policy (line 27) and the environment advances (line 28). In contrast to training (Algorithm 2), deployment computes no reward, queries no critic, and updates no parameters.

Training. BORA is trained with PPO, using the clipped surrogate objective $\hat{\mathcal{L}}^{\mathrm{CLIP}}(\phi) = \mathbb{E}_t\left[\min\left(\frac{\pi(a_t|o_t;\phi)}{\pi(a_t|o_t;\phi_{\mathrm{old}})} Z_t, \mathrm{clip}\left(\frac{\pi(a_t|o_t;\phi)}{\pi(a_t|o_t;\phi_{\mathrm{old}})}, 1-\epsilon, 1+\epsilon\right) Z_t\right)\right]$ where Z_t is the advantage function and ϕ_{old} are the parameters of the previous policy that generated the trajectories in the learning buffer. Training uses Generalized Advantage Estimation (GAE) advantages, as the attacker best-responds to the trained honest policy $\pi_{\mathrm{M}}(\cdot; \theta^*)$ (Algorithm 2). The environment is reset once before training begins (line 1); within each rollout an episode reset occurs automatically on termination or truncation. Each of the T rollout steps invokes the `Step` routine of Algorithm 1 with stochastic offsets $\delta_t^{(b)} \sim \pi_{\mathrm{C}}(o_t^{(b)}; \phi)$, which produces the falsified broadcasts $\tilde{s}_t^{(b)}$, the simultaneous movement of all nodes under $\pi_{\mathrm{M}}(\cdot; \theta^*)$, and the resulting throughput τ_t. Unlike the heuristic direction lie, BORA falsifies the full broadcast vector and can therefore exploit structure across all state fields. The Byzantine agent then receives reward $r_t = -\tau_t$ (line 8); each step appends the global state to the critic buffer $\mathcal{D}_\psi$ (line 9) and the Byzantine observation and offset to the actor buffer $\mathcal{D}_\phi$ (line 10). After the T steps, the centralized critic $\hat{V}_\psi(s_t)$ (parameters ψ) gives the GAE advantage $\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \zeta_{t+l}$, where $\zeta_t = r_t + \gamma \hat{V}_\psi(s_{t+1}) - \hat{V}_\psi(s_t)$ is the TD residual (line 14). The Byzantine policy is updated from $\hat{A}_t$ via $\hat{\mathcal{L}}^{\mathrm{CLIP}}$ (line 16), while the critic regresses onto the return $\hat{Z}_t = \hat{A}_t + \hat{V}_\psi(s_t)$ (line 15) and is discarded after training. Because the honest policy is a deterministic function of the observations derived

Algorithm 1: BORA Decentralized Execution

Input: Trained honest and Byz. policies' parameters θ^*, ϕ^*

// initialize start states and observations

1 $s_1^{(i)} \sim \pi_{\mathrm{s}}, \quad o_1^{(i)} \leftarrow \mathbf{0}\ \forall i \in \mathcal{V}, \quad \tilde{s}_1^{(i)} \leftarrow s_1^{(i)} \forall i \in \mathcal{B}$

// initialize AoIs

2 $(\Delta_1^{(i)})_j \leftarrow +\infty \quad \forall i, j \in \mathcal{V}$

3 **for** $t \in \mathbb{N}$ **do**

4 $\quad (s_{t+1}, \tilde{s}_{t+1}, o_{t+1}, \Delta_{t+1}, \cdot, \cdot) \leftarrow$ `Step` $(t, s_t, \tilde{s}_t, o_t, \Delta_t, \theta^*, \phi^*)$

5 **Function** `Step` $(t,\ s_t,\ \tilde{s}_t,\ o_t,\ \Delta_t, \theta^*, \phi)$ **:**

6 $\quad$ **for** $i \in \mathcal{V}$ **in parallel do**

$\quad\quad$ // observe local state

7 $\quad\quad (o_t^{(i)})_i \leftarrow s_t^{(i)}, \quad \Delta_t(i,i) \leftarrow 0$

$\quad\quad$ // honest nodes send their true state

8 $\quad\quad$ `SendTo` $(k,\ s_t^{(j)})\ \forall j \in \mathcal{H},\ k \in \mathcal{R}_t(j)$

9 $\quad\quad$ **for** $j \in \mathcal{H}$ **in parallel do**

10 $\quad\quad\quad$ **if** $i \in \mathcal{R}_t(j)$ **then**

$\quad\quad\quad\quad$ // receive honest state and timestamp

11 $\quad\quad\quad\quad \big((o_t^{(i)})_j, t_j\big) \leftarrow$ `ReceiveFrom` $(j,\ s_t^{(j)})$

$\quad\quad\quad\quad$ // timestamp t_j updates AoI

12 $\quad\quad\quad\quad (o_t^{(i)})_{N+j} \leftarrow \Delta_t(i,j) \leftarrow t - t_j$

$\quad\quad$ // Byzantine state falsification

13 $\quad\quad \delta_t^{(b)} \sim \pi_{\mathrm{C}}(o_t^{(b)};\ \phi) \quad \forall b \in \mathcal{B}$

14 $\quad\quad \tilde{s}_t^{(b)} \leftarrow s_t^{(b)} + \delta_t^{(b)} \quad \forall b \in \mathcal{B}$

$\quad\quad$ // Byzantine nodes send falsified state

15 $\quad\quad$ `SendTo` $(k,\ \tilde{s}_t^{(j)})\ \forall j \in \mathcal{B},\ k \in \mathcal{R}_t(j)$

$\quad\quad$ // collusion: Byz. peers' falsified and covert true state

16 $\quad\quad$ **for** $j \in \mathcal{B}$ **in parallel do**

17 $\quad\quad\quad$ **if** $i \in \mathcal{R}_t(j) \cap \mathcal{B}$ **then**

$\quad\quad\quad\quad$ // receive Byz. state

18 $\quad\quad\quad\quad \big((o_t^{(i)})_j, t_j\big) \leftarrow$ `ReceiveFrom` $(j,\ \tilde{s}_t^{(j)})$

19 $\quad\quad\quad\quad (o_t^{(i)})_{N+j} \leftarrow \Delta_t(i,j) \leftarrow t - t_j$

$\quad\quad\quad\quad$ // covert: receive honest state from Byz.

20 $\quad\quad\quad\quad \big((o_t^{(i)})_{2N+j}, t_j\big) \leftarrow$ `ReceiveFrom` $(j,\ s_t^{(j)})$

21 $\quad\quad\quad\quad (o_t^{(i)})_{2N+j} \leftarrow t - t_j$

22 $\quad\quad\quad\quad (o_t^{(i)})_{3N+j} \leftarrow 1$ // Byzantine flag

$\quad\quad$ // honest nodes receive the Byz. states

23 $\quad\quad$ **for** $j \in \mathcal{B}$ **in parallel do**

24 $\quad\quad\quad$ **if** $i \in \mathcal{R}_t(j) \cap \mathcal{H}$ **then**

25 $\quad\quad\quad\quad \big((o_t^{(i)})_j, t_j\big) \leftarrow$ `ReceiveFrom` $(j,\ \tilde{s}_t^{(j)})$

26 $\quad\quad\quad\quad (o_t^{(i)})_{N+j} \leftarrow \Delta_t(i,j) \leftarrow t - t_j$

$\quad\quad$ // all nodes move simultaneously honestly

27 $\quad\quad a_t^{(i)} \leftarrow \pi_{\mathrm{M}}(o_t^{(i)};\ \theta^*) \quad \forall i \in \mathcal{V}$

$\quad\quad$ // prepare next step

$\quad\quad$ // measure local states, throughput computed only in CT phase

28 $\quad\quad (s_{t+1},\ \tau_t) \sim \mathbb{P}(\cdot \mid s_t,\ a_t)$

$\quad\quad$ // carry over observation, increase AoI

29 $\quad\quad o_{t+1}^{(i)} \leftarrow o_t^{(i)}, \quad \Delta_{t+1}(i,j) \leftarrow \Delta_t(i,j) + 1 \quad \forall i, j \in \mathcal{V}$

30 $\quad$ **return** $(s_{t+1},\ \tilde{s}_{t+1},\ o_{t+1},\ \Delta_{t+1},\ \tau_t,\ (\delta_t^{(b)})_{b \in \mathcal{B}})$

from s_t, the honest actions a_t^H are implicitly available to the state-conditioned critic via θ^*. The honest weights θ^* are not updated during adversarial training, so the attacker solves $\phi^*(\theta^*) = \arg\max_\phi J(\theta^*, \phi)$, where $J(\theta^*, \phi)$ is its expected cumulative return; full access to $\pi_{\mathrm{M}}(\cdot; \theta^*)$ lets it generate unlimited offline rollouts to optimally tune ϕ.

Algorithm 2: BORA Actor-Critic Centralized Training

Input: Honest policy parameter θ^*, Rollout length T, Step budget $T_{\max}$, Learning rate α, Discount factor γ, GAE factor λ
Output: ϕ^*: learned Byzantine policy parameter

// initialize start states and observations
$s_1^{(i)} \sim \pi_{\mathrm{s}}, \quad o_1^{(i)} \leftarrow \mathbf{0}\ \forall\, i \in \mathcal{V}, \quad \tilde{s}_1^{(i)} \leftarrow s_1^{(i)} \forall i \in \mathcal{B}$
// initialize AoIs
$(\Delta_1^{(i)})_j \leftarrow +\infty \quad \forall i, j \in \mathcal{V}$
// init. Byz. critic and shared-param actors
$\phi^{(b)} = \phi \sim \rho \quad \forall b \in \mathcal{B}, \quad \psi \sim \rho$
for $n \in \{1, \ldots, T_{max}/T\}$ **do**
  $\mathcal{D}_\psi \leftarrow \emptyset, \quad \mathcal{D}_\phi \leftarrow \emptyset$
  for $t \in \{1, \ldots, T\}$ **do**
    $\left(s_{t+1}, \tilde{s}_{t+1}, o_{t+1}, \Delta_{t+1}, \tau_t, (\delta_t^{(b)})_{b\in\mathcal{B}}\right) \leftarrow$ Step $(t, s_t, \tilde{s}_t, o_t, \Delta_t, \theta^*, \phi)$
    // Byzantine reward opposes throughput
    $r_t \leftarrow -\tau_t$
    // add global info to critic train buffer
    $\mathcal{D}_\psi \leftarrow \mathcal{D}_\psi \cup \left\{(r_t, s_t)\right\}$
    // add local info to Byz. actor train buffer
    $\mathcal{D}_\phi \leftarrow \mathcal{D}_\phi \cup \left\{(r_t, o_t^{(b)}, \delta_t^{(b)})_{b\in\mathcal{B}}\right\}$
  // GAE: TD residual ζ_t, advantage, value target
  for $t \in \{1, \ldots, T\}$ **do**
    $\zeta_t \leftarrow r_t + \gamma \hat{V}_\psi(s_{t+1}) - \hat{V}_\psi(s_t)$
    $\hat{A}_t \leftarrow \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \zeta_{t+l}$
    $\hat{Z}_t \leftarrow \hat{A}_t + \hat{V}_\psi(s_t)$
  // update critic and Byzantine policy (parameter sharing)
  $\psi \leftarrow \psi - \alpha\, \nabla_\psi \hat{\mathcal{L}}^{\mathrm{VF}}(\psi;\, \mathcal{D}_\psi, (\hat{Z}_t)_{t=1}^T)$
  $\phi \leftarrow \phi + \alpha\, \nabla_\phi \hat{\mathcal{L}}^{\mathrm{CLIP}}(\phi;\, \mathcal{D}_\phi, (\hat{A}_t)_{t=1}^T)$
return ϕ

A. Communication Protocol

A Byzantine node participates in the communication protocol exactly as an honest node: at each step it broadcasts its falsified state $\tilde{S}^{(b)}$ to its recipient set $\mathcal{R}_t(b)$ and conditions its communication policy on the states it receives from $\mathcal{R}_t(b)$. The method fixes no particular protocol; we parameterize it by a single *propagation radius* $\rho \in \mathbb{N}_0 \cup \{\infty\}$, the maximum number of communication hops a broadcast may travel. Writing $\mathrm{hop}_t(i, j)$ for the hop distance between i and j in the time-t in-range graph (with $\mathrm{hop}_t(i, i) = 0$ and 1-hop neighborhood $\mathcal{N}_t(i)$), the recipient set is $\mathcal{R}_t(i) = \{\, j \in \mathcal{V} : \mathrm{hop}_t(i, j) \le \rho \,\}$.

B. Adversarial Training Defense

To evaluate the feasibility of *adversarial training* as a lightweight defense, the undefended honest policy is fine-tuned under identical hyperparameters with the trained BORA attacker active throughout. Fine-tuning from the undefended checkpoint, rather than retraining from scratch, keeps the procedure lightweight: the honest nodes are continually exposed to BORA's falsified broadcasts and must adapt their relocation strategy to remain effective, without any detection module, cryptography, or architectural change.

The Byzantine policy used for fine-tuning is the BORA attacker previously trained against the undefended honest policy, and it stays frozen throughout, so the honest policy adapts against a fixed adversary rather than a co-evolving one. All other hyperparameters and the training objective are identical to the undefended baseline.

V. Experimental Setup

A. Communication Protocols

We evaluate topology-constrained propagation radii $\rho = \infty$ and $\rho = 1$. Under $\rho = \infty$, a local state transmitted by a node traverses a sufficient number of hops to reach all nodes connected to the sender by a directed path. Under $\rho = 1$, the local state reaches only direct neighbors (i.e., $\mathcal{R}_t(i) = \mathcal{N}_t(i)$). Both protocols use perfect, lossless in-step delivery: every broadcast reaches its recipients within the step it is sent. Under $\rho = \infty$, a node refreshes the state of every node from which it is currently reachable and retains the last received state for all others. Under $\rho = 1$, a node refreshes only the slots of its current neighbors and retains states from nodes outside of range. Additionally, we consider the *Oracle* protocol, a special abstract protocol that represents an upper bound of the information available to the node, we assume that each node has direct centralized access to the full joint state $S_t \in \mathbb{R}^{Nd}$, regardless of network connectivity (i.e., $\mathcal{R}_t(i) = \mathcal{V}$).

B. Environments

First, we evaluate the vulnerability of the undefended honest policy, i.e., trained without adversaries, to heuristic and learned attacks of variable strengths, aiming at finding an upper bound on attack impact. Heuristic attack evaluations assume $\rho = \infty$ for both honest and malicious nodes, while learned attack evaluations assume the Oracle protocol for the honest nodes. Since Oracle provides more information to honest nodes than normally available in a realistic deployment, the measured disruption is a lower bound for learned attacks.

All experiments consider assume static user positions, a single jammer who stops moving upon activation at $t = 200$, $f = 1$ or $f = 2$ Byzantine nodes, and $N = \{5, 7, 10\}$ nodes, all with action scale $\kappa = 0.02$.

Second, we evaluate the effectiveness of adversarial training as a defense against frozen BORA nodes by fine-tuning the honest policy in an $f = 2$ scenario.

The policies of the honest actors are trained for 3×10^7 steps and use full parameter sharing among themselves, i.e., a single neural network shared by all honest actors. Similarly, the policies of the f Byzantine actors also use full parameter sharing, i.e., a single neural network shared by all f malicious actors. The actor and critic do not share weights. All learned

attacks are trained for 6×10^7 steps, except for BORA Oracle, which is trained for 3×10^7 steps.

Episode counts per configuration range from 1000 (the adversarial-defense conditions) through 2000 (the heuristic sweeps) to 4000 (the learned-attack evaluations).

C. Metrics

Let us denote the mean throughput under f Byzantine nodes, averaged over the last $W = 500$ steps across E episodes of length T as $\bar{\tau}(f) = \frac{1}{E \cdot W} \sum_{e=1}^{E} \sum_{t=T-W+1}^{T} \tau_t^{(e)}$ where $\tau_t^{(e)}$ is the per-step network throughput at step t of episode e. The *no-attack* mean throughput $\bar{\tau}(0)$ is evaluated in a separate run with $f = 0$ Byzantine nodes; all other parameters (random seed, user positions, jammer configuration) are held identical. We now define the *throughput degradation* as $\Delta\tau(f) = \frac{\bar{\tau}(0) - \bar{\tau}(f)}{\bar{\tau}(0)}$. For every experiment, we measure the *per-step throughput* $\tau_t[\mathrm{bit/s}]$ to visualize the temporal dynamics of each attack, particularly the pre-jammer ($t < 200$) and post-jammer ($t \geq 200$) regimes.

D. Baselines

- **No Byzantine**: the trained policy evaluated with $f = 0$; serves as the no-attack reference for $\Delta\tau(f)$.
- **Noise Baseline**: adds Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, m^2\mathbf{I})$, with zero mean and m^2 variance, to every element of the broadcast state vector; lower-bound reference requiring no structural knowledge of the observation format.
- **Direction Lie**: the Byzantine node broadcasts falsified directions, shifted by a fixed offset, to the closest user $u_{\mathrm{dis}}(b) + (m, 0)$, to closest user who needs to send data $s_{\mathrm{dis}}(b) + (m, 0)$, and to the closest user who is a destination of communication $r_{\mathrm{dis}}(b) + (m, 0)$, aiming at making honest nodes misestimate where traffic demand is concentrated and relocate away from it.
- **Capacity Lie**: the Byzantine node advertises itself as a well-connected relay hub to attract honest neighbors, i.e., scales reported in- and out-link reported capacities as $(1+m)C_+(b)$ and $(1+m)C_-(b)$,
- **Demand Lie**: inflates reported sender/receiver traffic demand as $(1+m)l_+(b)$ and $(1+m)l_-(b)$, aiming at luring honest nodes to cluster around the Byzantine node expecting traffic that does not exist.
- **Interference Lie**: reduces the reported interference $I(b)$ as $\max\{0, 1-m\}I(b)$ regardless of jammer proximity, aiming at luring honest nodes into a jammed zone by advertising low interference.
- **Temporal Replay**: the Byzantine node broadcasts its own local state from $\lfloor 50m \rfloor$ steps in the past.

We also introduce the *learned direction lie* baselines, which share BORA's training procedure but restrict the offset to the user-direction component, serving as direction-only baselines that isolate the contribution of falsifying the full broadcast vector. In all learned (direction lie and BORA) attack evaluations, the honest policies observe the full global observation vector, composed of honest states and Byzantine falsified states.

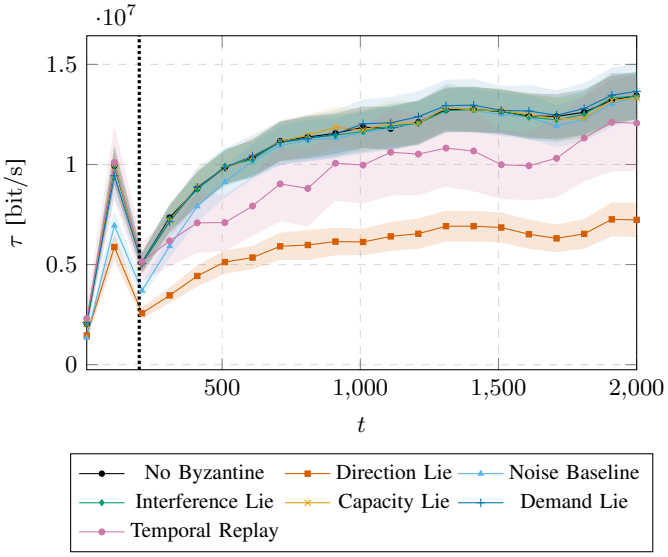


Fig. 3. Per-step throughput τ for heuristic attacks and the noise baseline with $N = 5$, $f = 2$, $m = 4$, and an active jammer. All nodes (honest and Byzantine) use $\rho = \infty$. Curves show mean throughput of 500 evaluation episodes. Jammers activate at step $t = 200$ (dotted vertical line).

- **Learned Direction Lie** ($\rho = 0$): Each Byzantine node relies only on its local state $S^{(b)}$ and does not receive any state from other nodes. It is a lower bound for the information available to the attacker.
- **Learned Direction Lie** (Oracle): Each Byzantine node observes the full global state $S = (S^{(1)}, \ldots, S^{(N)})$, regardless of network connectivity. It is an upper bound for the information available to the attacker.
- **Learned Coordinated Direction Lie** ($\rho = 1$): Each Byzantine node receives local observations from neighboring honest nodes and all other Byzantine nodes via a covert side-channel, independently of communication range. It enables inter-Byzantine coordination and provides more information to the attacker than $\rho = 0$; a more capable learned reference.

VI. Results Analysis

A. Heuristic Attacks With Active Jammers

Figure 3 shows the temporal dynamics once jammers are introduced. The honest baseline drops sharply at $t = 200$ when jammers activate, and Byzantine attacks amplify the drop to varying degrees. Among heuristic attacks, the direction lie achieves the largest post-jammer throughput reduction. After jammer activation at $t = 200$, it sustains approximately $47\,\%$ degradation relative to the honest baseline by systematically displacing honest nodes away from traffic demand.

Figure 4 presents the harm curves as a function of magnitude for all heuristic attacks, including the temporal replay attack. The direction lie reaches $47.2 \pm 5.1\,\%$ degradation at $m = 4$ (relative to the realistic-observation honest baseline $\bar{\tau}(0) = 12.79\,\mathrm{Mbit/s}$), confirming it as the dominant attack across all tested magnitudes. The interference lie remains near

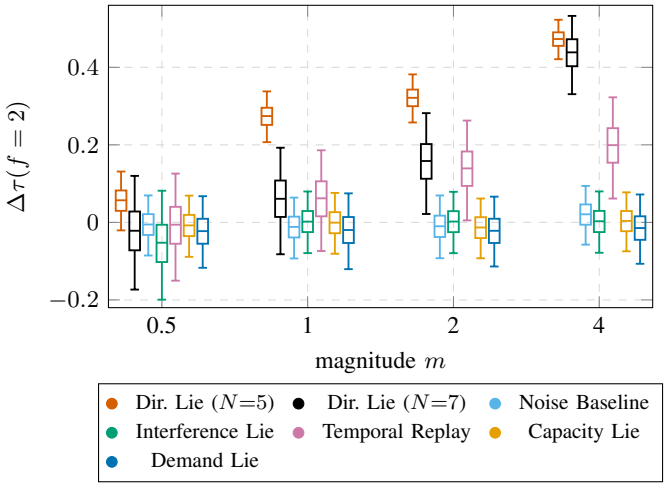


Fig. 4. Impact of attack magnitude m on throughput degradation $\Delta\tau(f=2)$ for $N=5$ and $\rho=\infty$. Each each box summarizes the distribution of mean degradations for 2000 episodes, containing an active jammer.

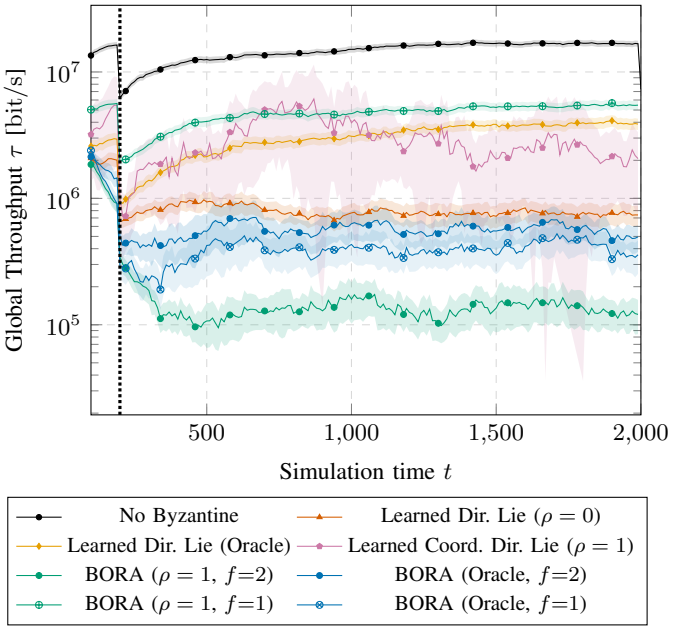


Fig. 5. Per-step throughput τ for BORA, under the $\rho=1$ and the Oracle communication protocols at $f=1$ and $f=2$, and the three learned direction-only baselines, against the honest baseline, for $N=5$. Curves average 4000 episodes for learned direction baselines and BORA $\rho=1$, and 2000 episodes for BORA Oracle.

zero across all tested magnitudes, indicating that observation-based interference manipulation is ineffective at disrupting the learned policy. The noise baseline Confidence Interval (CI) bands overlap zero at all tested magnitudes, confirming that undirected noise is effectively harmless regardless of perturbation strength. The temporal replay causes modest degradation that grows with delay magnitude, remaining well below the direction lie across all magnitudes.

B. BORA

Figure 5 shows per-step throughput for BORA and the learned direction-only baselines alongside the honest reference, whose full-state observation baseline is $\bar{\tau}(0) = 16.78\,\mathrm{Mbit/s}$. BORA $\rho=1$ degrades the global throughput by $99.2\pm0.2\,\%$, reducing it to $0.14\pm0.03\,\mathrm{Mbit/s}$, by falsifying the broadcast vector. This is the strongest attack measured, exceeding the $47.2\pm5.1\,\%$ of the heuristic direction lie by more than a factor of two. Restricting the falsification to the direction field, the learned baselines reach $95.4\pm0.7\,\%$ (Learned Direction Lie $\rho=0$), $85.3\pm2.2\,\%$ (Learned Coordinated Direction Lie), and $77.0\pm2.0\,\%$ (Learned Direction Lie Oracle), each above the heuristic direction lie. The heuristic direction lie applies a fixed, predetermined perturbation. BORA, by contrast, adapts its perturbations at every step based on the current network state, discovering time-varying strategies against which the fixed honest policy provides no adequate defense. With a single Byzantine node, BORA already degrades throughput by $67.9\pm2.3\,\%$, so severe disruption does not require multiple compromised nodes.

These results point to a simple mechanism: a wider observation scope (i.e., larger ρ) helps an attacker only when its falsification is expressive enough to exploit the additional information. Because BORA falsifies the entire broadcast vector, a wider view sharpens its perturbation. Under the Oracle protocol, where the Byzantine policy can observe the whole joint state rather than only the states it hears from neighbors (i.e., as in $\rho=1$), a single node already saturates the attack, degrading throughput by $97.5\pm0.5\,\%$ ($\bar{\tau}=0.43\pm0.09\,\mathrm{Mbit/s}$) at $f=1$ against the $67.9\,\%$ that $\rho=1$ reaches with one node. Therefore, the Oracle protocol's richer information erases the one-to-two-node scaling needed by the $\rho=1$ protocol for near-total denial. In a scenario with $N=5$ nodes and $f=2$ Byzantine, the BORA Oracle degrades throughput by $96.6\pm0.7\,\%$, corresponding to a throughput during attack of $\bar{\tau}=0.57\pm0.11\,\mathrm{Mbit/s}$), whereas the BORA $\rho=1$ achieves a stronger $99.2\,\%$ throughput degradation. BORA Oracle $f=1$ and $f=2$ induce similar throughput degradation, indicating that a single Oracle node already maximize disruption under that protocol. The slightly lower degradation of BORA Oracle at $f=2$ is likely due to the increased difficulty of optimizing two coordinated policies under a fixed training budget, rather than an inherent limitation of the attack. Conversely, the Learned Direction Lie baselines perform worse as the attacker's observation scope increases, i.e., the throughput degradation of $\rho=0$ > Coordinated > Oracle. The attacker does not benefit from the additional information coming from higher-hop state exchange or full-state knowledge because the Oracle and Coordinated observations are harder to optimize under a fixed training budget and the information may not be useful for the attack.

Figure 6 compares BORA and the learned direction-only baselines against the heuristic attacks. BORA reaches $99.2\pm0.2\,\%$ degradation ($\bar{\tau}=0.14\pm0.03\,\mathrm{Mbit/s}$) at $f=2$ and $67.9\pm2.3\,\%$ ($\bar{\tau}=5.39\pm0.39\,\mathrm{Mbit/s}$) at $f=1$, the strongest

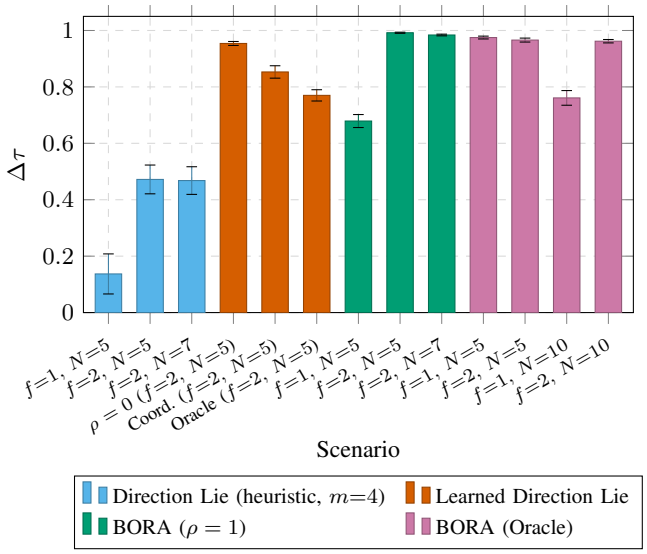


Fig. 6. Throughput degradation $\Delta\tau(f)$ grouped by attack family: the heuristic direction lie (blue, $m = 4$), the learned direction lie (red; $\rho = 0$, Oracle, Coordinated), BORA $\rho = 1$ (green), and BORA Oracle (pink).

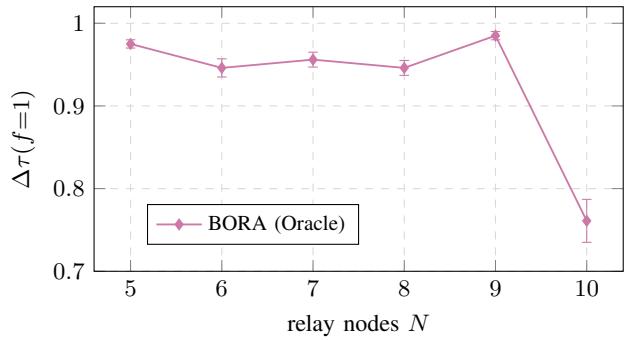


Fig. 7. BORA scalability $f = 1$ scenario: impact of the number N of nodes on throughput degradation $\Delta\tau$. Honest and BORA nodes use the Oracle protocol. Each point is normalised to the matched-N clean baseline (2000 episodes).

results. The Learned Direction Lie baselines all exceed the heuristic direction lie: Learned Direction Lie $\rho = 0$ degrades $95.4 \pm 0.7\,\%$ ($\bar{\tau} = 0.76 \pm 0.13\,\mathrm{Mbit/s}$), Learned Coordinated degrades $85.3\,\%$ ($\bar{\tau} = 2.46\,\mathrm{Mbit/s}$), and Learned Oracle degrades $77.0 \pm 2.0\,\%$ ($\bar{\tau} = 3.87 \pm 0.33\,\mathrm{Mbit/s}$). Learned attacks are evaluated over 4000 episodes and Oracle BORA over 2000 episodes under the full-state baseline. The heuristic attacks are evaluated over 2000 episodes under the realistic-observation baseline.

C. Byzantine- and Honest-Node Scalability

Figure 6 reports node-count scaling for the direction lie under the same jammer conditions used throughout. The relationship is super-linear: a single Byzantine node achieves only $13.7\,\%$ degradation at $m = 4$, as four honest nodes broadcasting correct directions swamp the single lie and the honest policy largely compensates. Adding a second Byzantine node shifts the ratio to three honest against two fake, surpassing a tipping point where the policy can no longer self-correct,

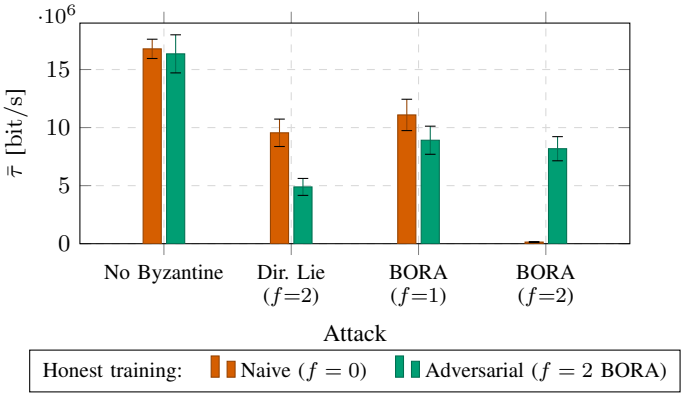


Fig. 8. Mean throughput $\bar{\tau}$ for the honest policies trained without Byzantine nodes (Naive $f = 0$, red bars) and adversarially fine-tuned (against $f = 2$ BORA, green bars), under various attack scenarios: a no-Byzantine baseline with no attacker, the heuristic direction lie ($f = 2$, $m = 4$), and BORA at $f = 1$ and $f = 2$. Each bar reports mean and confidence interval of 1000 episodes. During fine-tuning, honest actors use the Oracle protocol, while $f = 2$ frozen BORA attackers use $\rho = 1$. During evaluation, the honest actors use the Oracle protocol against BORA ($f = 1$, $f = 2$), and $\rho = \infty$ for the no-Byzantine baseline and the heuristic Direction Lie attack. During evaluation, the BORA attackers use $\rho = 1$ and the heuristic direction lie attackers use $\rho = \infty$.

yielding $47.2 \pm 5.1\,\%$ degradation, a 3.4-fold increase. At $N = 7$ (Figure 4, Figure 6) the direction lie yields $46.8 \pm 4.9\,\%$ degradation at $m = 4$ and BORA yields $98.4 \pm 0.3\,\%$, both statistically indistinguishable from $N = 5$: adding honest nodes provides no measurable resilience against either attack.

Figure 7 traces the single-node BORA attack across network size. At $f = 1$ the attack remains near-total for the smaller victims, degrading throughput by $97.5 \pm 0.5\,\%$ at $N = 5$, $94.6 \pm 1.1\,\%$ at $N = 6$, $95.6 \pm 0.9\,\%$ at $N = 7$, $94.6 \pm 0.9\,\%$ at $N = 8$, and $98.5 \pm 0.5\,\%$ at $N = 9$. At $N = 10$ a single Byzantine node still inflicts severe disruption but its relative harm falls to $76.1 \pm 2.6\,\%$ ($\bar{\tau} = 5.67\,\mathrm{Mbit/s}$ against the matched clean baseline $\bar{\tau}(0) = 23.75\,\mathrm{Mbit/s}$), the first clear departure from the near-complete-denial band: with ten relays the nine honest nodes begin to dilute the influence of one attacker. This dilution is specific to $f = 1$, however: adding a second Byzantine node at $N = 10$ restores near-total collapse at $96.2 \pm 0.6\,\%$ (Figure 6), so the harm tracks the Byzantine *fraction* f/N rather than f alone.

D. Adversarial Training Defense

Figure 8 compares the undefended honest policy against one adversarially fine-tuned against the $f = 2$ BORA attacker, both at full-state observation over 1000 episodes; each $\Delta\tau$ is taken relative to the corresponding policy's no-Byzantine throughput ($16.78 \pm 0.83\,\mathrm{Mbit/s}$ undefended versus $16.35 \pm 1.64\,\mathrm{Mbit/s}$ defended, a $2.6\,\%$ clean-performance cost). The defense is lightweight: the undefended policy is fine-tuned with Byzantine nodes present, requiring no detection module, cryptography, or architectural change. Against the exact attacker it was trained on, the defense is highly effective: it cuts $f = 2$ BORA degradation from $99.2 \pm 0.2\,\%$ to $50.0 \pm 6.4\,\%$,

raising throughput under attack from $0.14 \pm 0.04\,\mathrm{Mbit/s}$ to $8.18 \pm 1.04\,\mathrm{Mbit/s}$. It does not, however, generalize to attacks absent from fine-tuning. Under the heuristic direction lie ($f = 2$, $m = 4$) the fine-tuned policy is in fact *more* vulnerable than the undefended one, degrading by $70.1 \pm 5.4\,\%$ against $43.1 \pm 7.6\,\%$; under a single-node BORA attacker it degrades by $45.5 \pm 9.2\,\%$ against the undefended $33.9 \pm 8.7\,\%$. (This $f = 1$ BORA bar uses the $f = 2$-trained attacker deployed with a single Byzantine node and is therefore far weaker than the dedicated $f = 1$ full-vector attacker of Figure 7.) The honest policy thus overfits to the falsification pattern of its training adversary, trading away robustness to other manipulations for protection against the specific one. The attacker was moreover trained against the undefended policy and not retrained against the defended one, so an adaptive BORA would likely recover part of the $f = 2$ gain. The loss of robustness against unseen attacks shows that adversarial training against a single fixed adversary is insufficient on its own; co-training against a diverse attacker population, or combining it with structural countermeasures, is required for broad robustness.

VII. Conclusion

This work proposed BORA, a learned Byzantine attacker trained via RL adversarially against the honest cooperative relay policy, and evaluated it alongside heuristic observation-manipulation attacks under both standard and adversarial-training conditions. With two Byzantine nodes falsifying their full broadcast vector, BORA degrades throughput by $99.2 \pm 0.2\,\%$, more than doubling the $47.2 \pm 5.1\,\%$ of the strongest heuristic. BORA already degrades throughput by $67.9 \pm 2.3\,\%$ with a single Byzantine node, whereas the heuristic direction lie reaches only $13.7\,\%$ at one node and $47.2\,\%$ at two. Resilience tracks the Byzantine fraction f/N rather than network size alone. At $N = 7$ neither attack weakens, with the direction lie holding at $46.8 \pm 4.9\,\%$ and BORA at $98.4 \pm 0.3\,\%$ ($f = 2$), both statistically identical to $N = 5$, and although a lone full-state node at $N = 10$ sees its relative harm ease to $76.1 \pm 2.6\,\%$, a second node there restores near-total collapse at $96.2 \pm 0.6\,\%$. Adversarial fine-tuning against BORA halves its degradation, from $99.2\,\%$ to $50.0 \pm 6.4\,\%$ at a $2.6\,\%$ throughput cost, but this robustness does not transfer: against attacks unseen during fine-tuning the same policy is no better, and often worse, than the undefended one, showing that adversarial training against a single fixed adversary is insufficient on its own. Co-training against a diverse attacker population, combining adversarial training with structural countermeasures such as consensus-based broadcast verification, and evaluating under dynamic user mobility, remain open directions.

Acknowledgment

This work has been partially funded by the Swiss National Science Foundation project "Efficient Distributed Intelligent Applications in Mobile-Network Dynamics (eDIAMOND)" with grant number 223443.

References

[1] S. Leuenberger, A. Di Maio, and T. Braun, "Proactive decentralized multi-agent data-driven node relocation for jammed mobile networks," in *13th Wireless Days Conference (WD 2025)*. IEEE, 2025, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/11302544/

[2] Y. Li, Z. Guo, N. Yang, H. Chen, D. Yuan, and W. Ding, "Threats and defenses in federated learning life cycle: A comprehensive survey and challenges," *IEEE Transactions on Neural Networks and Learning Systems*, 2025. [Online]. Available: https://arxiv.org/abs/2407.06754

[3] S. Li, J. Guo, J. Xiu, R. Xu, X. Yu, J. Wang, A. Liu, Y. Yang, and X. Liu, "Byzantine robust cooperative multi-agent reinforcement learning as a bayesian game," in *International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=z6KS9D1dxt

[4] S. Li, R. Xu, J. Guo, P. Feng, J. Wang, A. Liu, Y. Yang, X. Liu, and W. Lv, "MIR2: Towards provably robust multi-agent reinforcement learning by mutual information regularization," in *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 2024. [Online]. Available: https://arxiv.org/abs/2310.09833

[5] S. Meshkat Alsadat and Z. Xu, "Decentralized byzantine-resilient multi-agent reinforcement learning with reward machines in temporally extended tasks," in *International Conference on Learning Representations (ICLR)*, 2026. [Online]. Available: https://openreview.net/forum?id=ydVFxjjtbA

[6] S. Li, J. Guo, J. Xiu, Y. Zheng, P. Feng, X. Yu, A. Liu, Y. Yang, B. An, W. Wu, and X. Liu, "Attacking cooperative multi-agent reinforcement learning by adversarial minority influence," *Neural Networks*, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608025006276

[7] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Adversarial attacks on multi-agent communication," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7768–7777. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Tu_Adversarial_Attacks_on_Multi-Agent_Communication_ICCV_2021_paper.html

[8] J. Lin, K. Dzeparoska, S. Q. Zhang, A. Leon-Garcia, and N. Papernot, "On the robustness of cooperative multi-agent reinforcement learning," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 62–68. [Online]. Available: https://arxiv.org/abs/2003.03722

[9] Y. Sun, R. Zheng, P. Hassanzadeh, Y. Liang, S. Feizi, S. Ganesh, and F. Huang, "Certifiably robust policy learning against adversarial communication in multi-agent systems," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=dCOL0inGl3e

[10] W. Xue, W. Qiu, B. An, Z. Rabinovich, S. Obraztsova, and C. K. Yeo, "Mis-spoke or mis-lead: Achieving robustness in multi-agent communicative reinforcement learning," in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022, pp. 1418–1426. [Online]. Available: https://dl.acm.org/doi/10.5555/3535850.3536008

[11] X. Ma, Y. Wang, J. Wang, X. Xie, B. Wu, S. Li, F. Xu, and Q. Wang, "Robust multi-agent communication based on decentralization-oriented adversarial training," *arXiv preprint arXiv:2504.21278*, 2025. [Online]. Available: https://arxiv.org/abs/2504.21278

[12] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," in *International Conference on Learning Representations (ICLR) Workshop Track*, 2017. [Online]. Available: https://arxiv.org/abs/1702.02284

[13] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 21 024–21 037. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/f0eb6568ea114ba6e293f903c34d7488-Paper.pdf

[14] T. Duan, Z. Zhang, Z. Lin, Y. Gao, L. Xiong, Y. Cui, H. Liang, X. Chen, H. Cui, and D. Huang, "Rethinking adversarial attacks in reinforcement learning from policy distribution perspective," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. [Online]. Available: https://arxiv.org/abs/2501.03562

[15] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: https://openreview.net/forum?id=HJgEMpVFwB