

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

SHIELD: Secure Human-Machine Interaction with Evidential Learning and Dynamic Trust for Drone Swarm Control

Dawood Wasif, *Virginia Tech;* Terrence J. Moore, *US DEVCOM Army Research Laboratory;*
Seunghyun Yoon and Hyuk Lim, *Korea Institute of Energy Technology (KENTECH);*
Dan Dongseong Kim, *The University of Queensland;* Frederica F. Nelson,
US DEVCOM Army Research Laboratory; Jin-Hee Cho, *Virginia Tech*

https://www.usenix.org/conference/vehiclesec26/presentation/wasif-shield

This paper is included in the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy.

August 10–11, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-56-4

Open access to the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy
is sponsored by USENIX.

ARTIFACT EVALUATED USENIX AVAILABLE
ARTIFACT EVALUATED USENIX FUNCTIONAL

SHIELD: Secure Human-Machine Interaction with Evidential Learning and Dynamic Trust for Drone Swarm Control

Dawood Wasif[1] *Terrence J. Moore*[2] *Seunghyun Yoon*[3] *Hyuk Lim*[3]
Dan Dongseong Kim[4] *Frederica F. Nelson*[2] *Jin-Hee Cho*[1]

[1]*Virginia Tech*
[2]*US DEVCOM Army Research Laboratory*
[3]*Korea Institute of Energy Technology (KENTECH)*
[4]*The University of Queensland*

Abstract

Drone swarms face a dual challenge: human operators cannot effectively supervise more than a dozen Unmanned Aerial Vehicles (UAVs) before situational awareness degrades, yet fully autonomous coordination remains vulnerable to cyber-physical attacks such as GPS spoofing, communication jamming, Byzantine manipulation, replay, malware injection, and network intrusion. We present SHIELD, a framework for secure human–swarm teaming built on a single unifying insight: *evidential vacuity*, derived from Dirichlet-based perception, jointly drives inter-agent trust modulation, intrusion detection, and human escalation within a decentralized control loop. A dynamic trust graph filters unreliable agents through behavioral consistency and quarantine, while an embedded intrusion response system autonomously handles detection, containment, and recovery. Operator involvement is reserved for genuinely ambiguous cases via an event-triggered binary interface that minimizes workload. In the considered waypoint-surveillance simulation scenarios with six attack classes and swarms of 8–32 UAVs, SHIELD achieves up to 60% relative improvement in mission success, detects over 80% of compromised agents with under 15% false positives, and reduces human interventions by 53% compared to baselines. Ablation and scalability studies confirm complementary module contributions and graceful performance degradation.

1 Introduction

Human Scalability and Adversarial Vulnerability. Autonomous drone swarms enable large-scale coverage in defense, surveillance, and search-and-rescue by coordinating dozens or hundreds of Unmanned Aerial Vehicles (UAVs) [8]. However, human operators cannot effectively supervise more than 10–12 UAVs before situational awareness and decision quality degrade [9]. At the same time, fully autonomous swarms remain vulnerable to cyber-physical attacks such as GPS spoofing, communication jamming, Byzantine manipulation, malware injection, and network intrusion [3]. These twin constraints of *cognitive scalability* and *adversarial resilience* reveal a fundamental gap: swarm autonomy must reduce operator workload while remaining robust to compromised agents and uncertain environments.

Limits of Existing Human–Swarm Frameworks. Human-in-the-loop and shared-autonomy methods improve learning efficiency and workload distribution [27, 36, 43], yet they largely assume benign settings and treat agent outputs as equally reliable. Most lack calibrated uncertainty estimation, decentralized trust modeling, and mechanisms to isolate compromised UAVs. Decision authority is often centralized, creating single points of failure and requiring frequent low-level interventions that undermine scalability [46]. Consequently, these systems neither detect adversarial manipulation nor degrade gracefully under uncertainty.

Fragmented Multi-Agent Resilience Approaches. Recent resilience methods address isolated aspects of the problem. Trust-adaptive control isolates non-cooperative agents but assumes centralized observers and omits uncertainty quantification and human oversight [22, 48]. Authority-allocation frameworks formalize intervention policies but target single-vehicle settings and do not scale to decentralized swarm settings [42]. Existing communication-robustness defenses neglect the human–machine teaming dimension. Critically, no prior work provides a unified uncertainty signal that governs trust modulation, intrusion detection, and operator escalation within a decentralized control loop.

Our Approach. We address these limitations with SHIELD (Secure Human-machine Interaction with Evidential Learning and Dynamic trust), a unified framework for scalable and adversary-resilient human–swarm teaming. SHIELD integrates evidential deep learning for calibrated uncertainty estimation [40] with dynamic trust modeling for decentralized, peer-weighted coordination [38]. Its central contribution is the use of a single uncertainty quantity, *evidential vacuity*, derived from Dirichlet-based perception, to jointly govern three critical functions: inter-agent trust adaptation, intrusion detection and quarantine, and event-triggered human escalation. By embedding uncertainty-aware perception, trust-weighted

consensus, and divergence-based safety overrides within one decentralized control loop, SHIELD eliminates single points of failure, isolates compromised UAVs in real time, and reduces unnecessary operator interventions. Unlike prior approaches that treat trust, anomaly detection, and human oversight as separate modules, SHIELD unifies them through a principled uncertainty signal, enabling robust performance under both operational uncertainty and cyber-physical attack.

Contributions. This work makes three primary contributions:

First, we introduce a **unified uncertainty-driven architecture** for secure human–swarm teaming. SHIELD leverages a single evidential vacuity signal, derived from Dirichlet-based perception, to jointly govern inter-agent trust adaptation, intrusion detection, and event-triggered human escalation within a decentralized control loop. By consolidating trust modeling, anomaly detection, and authority allocation through one calibrated uncertainty signal [22, 42, 48], the framework reduces system complexity while strengthening adversarial resilience.

Second, we develop a **dynamic trust graph with real-time quarantine** that enables decentralized, trust-weighted consensus and isolates compromised UAVs without centralized supervision. Trust updates are driven by calibrated uncertainty and behavioral consistency, enabling context-aware adaptation and graceful degradation under adversarial disruption.

Third, we provide **comprehensive adversarial evaluation** across six cyber-physical attack classes. In waypoint-surveillance simulations, SHIELD achieves up to 60% relative improvement in mission success, higher survival, and over 50% reduction in human interventions compared to baselines. The embedded intrusion response system detects over 80% of compromised agents with under 15% false positives and restores safe operation within 10 seconds.

2 Related Work

This section discusses prior work on reinforcement learning with human feedback, human–swarm interaction, multi-agent trust modeling, and cyber-physical UAV security, highlighting their limitations in achieving unified, uncertainty-aware, and adversary-resilient human–swarm coordination.

2.1 RL with Human Feedback (RLHF)

Human feedback (HF) has been widely incorporated into reinforcement learning (RL) to improve sample efficiency and safety. Deep TAMER [43], DQfD [20], and preference-based methods [4, 5, 6, 26, 31] leverage demonstrations, scalar rewards, or comparisons to guide policy learning. More recent work integrates meta-learning and uncertainty-aware reward modeling [19, 34]. However, these approaches largely assume single-agent settings and rely on frequent human input. They do not address decentralized coordination, adversarial compromise, or selective escalation in multi-UAV systems.

2.2 Human–Drone Teaming

Research on human–swarm interaction focuses on interface design and mixed-initiative control, including gesture-based control [1, 29], AR and haptic feedback [2, 49], and high-level strategic guidance [25, 30]. These systems improve situational awareness and workload distribution [12, 44], yet typically assume benign environments. They lack principled uncertainty quantification and do not provide mechanisms to determine when autonomous decisions should be trusted, overridden, or escalated under adversarial conditions.

2.3 Trust and Resilient Multi-Agent Systems

Trust-based coordination improves robustness in multi-agent systems. Behavioral-consistency updates [17, 38] and trust-weighted consensus enhance fault tolerance, while blockchain and reflective-control approaches strengthen reputation tracking and adaptation [18, 45]. Evidential reasoning has been explored for secure state inference [16], but prior work does not integrate calibrated uncertainty, decentralized trust adaptation, and human oversight within a unified control framework. As a result, trust modeling remains disconnected from intrusion detection and operator escalation.

2.4 Cyber-Physical Security for Swarms

Defenses against cyber-physical attacks on UAVs include sensor reconfiguration detection [15], protection against false localization attacks [13], and runtime behavior monitoring [10]. In multi-agent communication, active defense [47], Bayesian-game formulations [33], graph bottleneck filtering [11], and trust-aware communication [41] strengthen message reliability. However, these approaches operate as standalone detection or filtering modules: they identify anomalies or harden message channels but do not close the loop by feeding detection signals into trust adaptation, autonomous quarantine, or calibrated human escalation in a unified control architecture.

In contrast, SHIELD unifies calibrated uncertainty estimation, decentralized trust adaptation, intrusion response, and event-triggered human oversight within a single control architecture, directly addressing the scalability and adversarial gaps identified above.

3 System Model, Threats, and Operational Hazards

This section formalizes the system architecture, security assumptions, adversary capabilities, attack surface, and non-adversarial hazards considered in our evaluation.

3.1 System and Security Assumptions

We consider a cooperative drone swarm of N UAVs coordinated by a single human operator over a shared wireless communication network. Each UAV executes a local control policy based on onboard sensors (depth cameras, GPS, IMU) and peer-to-peer state messages exchanged with neighboring agents. The operator issues high-level mission objectives and may intervene when prompted.

The swarm operates in a bounded three-dimensional arena. Each UAV i maintains a local state estimate $\mathbf{s}_i(t)$ derived from its own sensors and messages received from neighbors $\mathcal{N}_i(t)$. Communication links are bidirectional and constrained by range and line-of-sight. The operator communicates with the swarm through a trusted and authenticated command channel.

We adopt the following security assumptions (S#), which define the boundaries of our evaluation and the guarantees under which the framework operates:

S1. The operator's command link is secure, authenticated, and cannot be spoofed or intercepted by the adversary.

S2. At most $f < \lfloor N/3 \rfloor$ UAVs are compromised at any time, ensuring that an honest majority remains for meaningful consensus and trust aggregation. This is a standard Byzantine fault tolerance prerequisite [32], not an assumption unique to this work; for our default $N = 8$, this permits $f = 2$ compromised agents.

S3. Sensor noise under benign conditions is bounded and follows known distributions (Section 5), enabling statistical deviation detection.

S4. UAVs do not share cryptographic secrets for inter-agent authentication; message integrity relies on behavioral consistency rather than full PKI (Public Key Infrastructure) guarantees.

S1 reflects standard military command-and-control practice (dedicated encrypted links), while **S4** reflects the computational constraints of lightweight UAV platforms that motivate behavior-based trust as an alternative to full PKI.

3.2 Adversary Model

Capabilities. The adversary may compromise UAVs or their communication channels. Specifically, the attacker can: inject false sensor readings (e.g., spoofed GPS); drop, delay, or jam inter-agent messages; cause compromised UAVs to broadcast arbitrary or inconsistent state information (Byzantine behavior); replay stale sensor data; deploy malicious firmware through supply-chain or update exploitation; or gain unauthorized access to the swarm's communication infrastructure to inject or intercept packets. The adversary may operate externally (RF injection, network manipulation) or from within the swarm's trust boundary.

Goals. The adversary seeks to degrade swarm integrity or performance along measurable dimensions evaluated in Section 5.3: (i) *mission disruption*, reflected by reduced Mission Success Rate (MSR) or Survival Rate (SR); (ii) *collision induction*, captured by worsened Crash Penalty (CP); (iii) *operator overload*, indicated by increased Human Intervention Count (HIC); (iv) *trust erosion*, observable through elevated state disagreement across agents and degraded collective task performance; and (v) *persistent stealth*, characterized by sustained performance degradation without corresponding detection signals, i.e., low Detection Rate (DR) and elevated False Positive Rate (FPR) that mask true compromise..

Knowledge Model. We consider two levels of adversary knowledge. Under a *black-box* model, the attacker knows the general architecture but not internal thresholds, trust weights, or monitoring variables. Under a *grey-box* model, the attacker knows the existence of uncertainty-based monitoring and trust-weighted consensus but cannot observe real-time internal states of uncompromised agents. Critically, internal defense variables—trust weights τ_{ij}, vacuity scores u_i, and override decisions—are computed and stored locally on each UAV and are *not* included in inter-agent broadcast messages $(\mathbf{s}_i, b_i, p_i)$. A compromised agent can therefore observe its own defense variables and the broadcast content of neighbors, but cannot directly read the trust scores or escalation states maintained by honest peers. This limits the adversary's ability to craft inputs that precisely manipulate another agent's trust or escalation logic. Experiments primarily evaluate the black-box setting; grey-box implications, including adaptive evasion strategies, are discussed in Section 7.

3.3 Attack Surface and Classes

The attack surface spans perception, communication, coordination, software integrity, and network layers. We instantiate six attack classes grounded in the UAV security literature [3, 8], covering both external and internal compromise.

A1: GPS Spoofing. False GPS signals corrupt the position estimate of targeted UAVs:

$$\hat{\mathbf{s}}_i(t) = \mathbf{s}_i(t) + \boldsymbol{\Delta}, \quad \boldsymbol{\Delta} \sim \mathcal{U}(-\delta_{\max}, \delta_{\max})^3. \tag{1}$$

Spoofing occurs periodically (approximately every 20 s) for 5 s, targeting the perception layer and potentially inducing collisions or formation breakdown.

A2: Communication Jamming. During 3–7 s windows, messages between affected agents are independently dropped with probability 0.5:

$$\mathcal{M}_{ij}(t) = \begin{cases} \emptyset & \text{with probability } 0.5, \\ \mathcal{M}_{ij}(t) & \text{otherwise.} \end{cases} \tag{2}$$

This degrades coordination and consensus at the communication layer, leading to incomplete neighborhood information.

A3: Byzantine Faults. Compromised UAVs broadcast fabri-

cated state information with probability $p = 0.10$:

$$\mathcal{M}_{ij}^{\text{Byz}}(t) = \begin{cases} \text{sample}(\mathcal{D}_{\text{states}}) & \text{with probability } p, \\ \mathcal{M}_{ij}(t) & \text{otherwise.} \end{cases} \quad (3)$$

This targets the coordination layer by introducing inconsistent state estimates that can fragment collective decision-making.

A4: Replay Attacks. Outdated sensor readings are re-injected into the perception pipeline:

$$x_i^{\text{replay}}(t) = x_i(t - \Delta t_r), \quad \Delta t_r = 10\,\text{s}. \quad (4)$$

Replay undermines temporal consistency in dynamic environments, causing decisions based on stale state information.

A5: Malware Injection. Compromised UAVs execute altered policies:

$$\pi_i^{\text{mal}}(a \mid s) \neq \pi_i(a \mid s), \quad (5)$$

with intermittent or state-dependent divergence. This targets software integrity and is hard to detect because compromised agents retain valid credentials and communication access.

A6: Network Intrusion. An adversary gains unauthorized access to the swarm's communication infrastructure, enabling packet injection or eavesdropping from within the network. Unlike jamming, this attack operates inside the communication channel and targets network integrity, potentially allowing coordinated manipulation across multiple agents.

Parameter Justification. Attack parameters are calibrated relative to the simulator's sensor noise floor and grounded in UAV security literature. GPS spoofing magnitude $\delta_{\max} = 3.0\,\text{m}$ corresponds to $\sim 10\sigma$ of benign GPS noise ($\sigma_{\text{GPS}} = 0.3\,\text{m}$), within the 1–5 m civilian spoofing range reported in field studies [24, 28]. Byzantine probability $p = 0.10$ models intermittent deception below trivially detectable thresholds [32]. Replay lag $\Delta t_r = 10\,\text{s}$ (500 control steps at 50 Hz) produces meaningful state divergence within commodity buffer limits [21]. Malware drift $\|\Delta\pi\| \leq 0.15$ balances operational impact against immediate detectability. Jamming drop probability (0.5) over 3–7 s windows reflects partial denial under power-limited RF contestation [35].

3.4 Non-Adversarial Operational Hazards

In addition to adversarial threats, the swarm must handle environmental uncertainties that may resemble attacks.

H1: Obstacle Encounters. Dynamic or previously unmapped obstacles may require rapid replanning. Legitimate evasive maneuvers must not be misclassified as compromise.

H2: Airspace Congestion. High agent density can saturate communication bandwidth and degrade coordination, resembling jamming or Byzantine faults at the behavioral level.

H3: Emergency Conditions. Environmental disturbances (e.g., wind gusts), sensor failures, or low battery events may cause deviations that trigger false alarms if not distinguished from adversarial manipulation.

These hazards require a defense capable of joint attribution across security and environmental states, minimizing both missed detections and excessive false positives.

3.5 Scope and Exclusions

The threat model focuses on cyber-physical attacks and operational hazards within the swarm's communication and control stack. We explicitly exclude: (i) physical capture, hardware tampering, or side-channel extraction of UAV components; (ii) compromise of the operator's authenticated command channel; and (iii) coordinated multi-agent deception campaigns that violate the fault bound $f < \lfloor N/3 \rfloor$.

The current evaluation considers parameterized, non-adaptive attacks. A more capable adversary could adopt an *intermittent Byzantine* strategy, reverting to honest behavior upon sensing impending quarantine and resuming after reintegration. SHIELD's reintegration protocol partially mitigates this by requiring m consecutive consistent steps before restoring trust from τ_{init}, imposing a minimum honest-operation cost per evasion cycle. Each cycle forces the adversary to behave honestly for at least $m \cdot \Delta t$ seconds during which it cannot advance its attack objective, and trust is restored only to τ_{init} rather than its pre-quarantine level, bounding the number of feasible evasion cycles per episode. The gap between detection without IRS and with full IRS (Table 2) confirms that layered defense substantially narrows the stealth window, though systematic evaluation of adaptive and RL-based adversaries is deferred to future work.

The instantiated attacks (A1–A6) and operational hazards (H1–H3) collectively span perception, communication, coordination, software integrity, and network layers. Within this scope, the defense must quantify perceptual uncertainty, evaluate peer reliability, attribute anomalies across security and environmental causes, and enforce safe fallback behavior. The next section introduces a framework designed to meet these requirements.

4 Proposed Approach: SHIELD

SHIELD enables scalable and trustworthy human–swarm teaming under uncertainty and cyber-physical threats. As illustrated in Figure 1, each UAV performs uncertainty-aware perception, coordinates with neighbors through a trust-weighted shared mental model, arbitrates decisions via consensus and safety overrides, and manages compromised agents through quarantine and reintegration mechanisms.

We model the human–swarm team as a time-varying, trust-weighted graph:

$$\mathcal{G}(t) = (V, E(t), \tau(t)), \quad (6)$$
$$V = \{0, 1, 2, \ldots, N\}, \quad E(t) \subseteq V \times V, \quad \tau_{ij}(t) \in [0, 1].$$

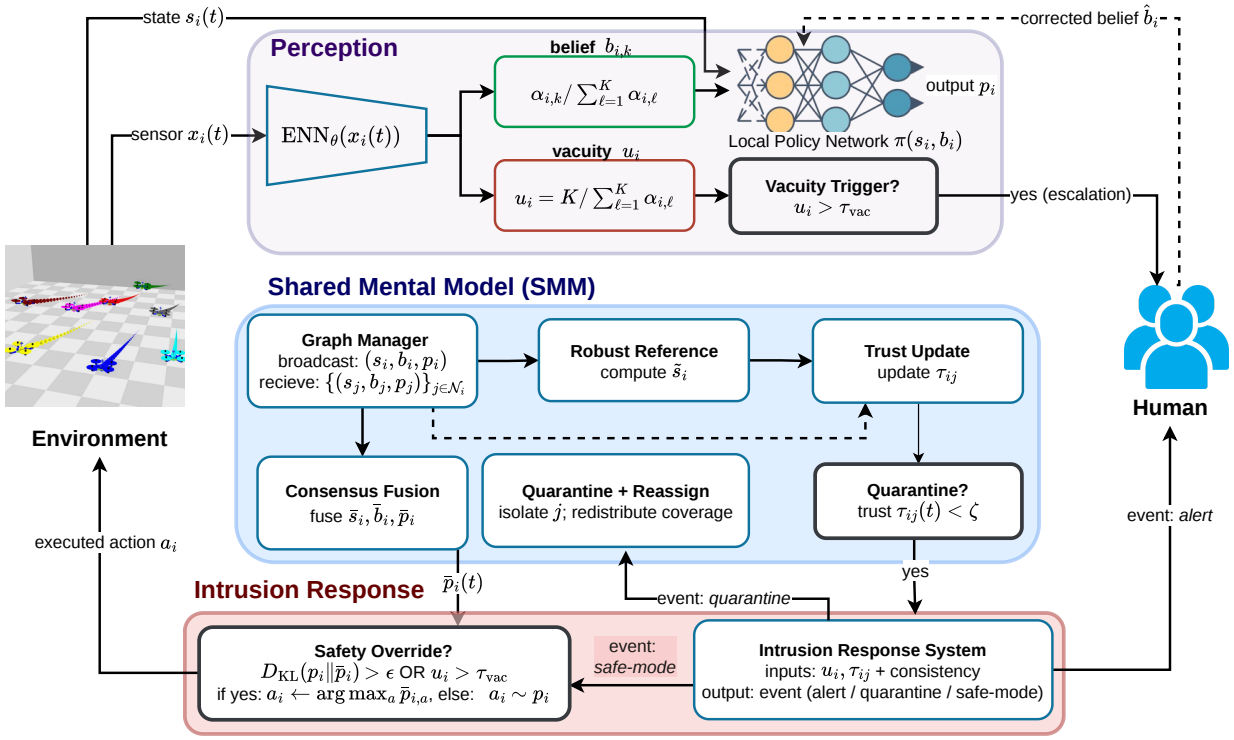


Figure 1: Proposed SHIELD overview for secure human–swarm teaming.

Node 0 represents the human operator and nodes $i = 1, \ldots, N$ represent UAVs. Each edge $(i, j) \in E(t)$ carries trust weight $\tau_{ij}(t)$, governing reliance on neighbor information. Edges are removed upon quarantine and restored during reintegration, enabling decentralized, trust-aware coordination.

4.1 Evidential Perception and Uncertainty-Driven Escalation

This component equips each UAV with calibrated belief estimation and epistemic uncertainty quantification, enabling principled human escalation only when observations are ambiguous or out-of-distribution.

Evidential Perception. Each UAV i processes sensor input $x_i^K(t) \in \mathcal{X}$ using an evidential neural network ENN_θ, producing Dirichlet concentration parameters:

$$\boldsymbol{\alpha}_i^K(t) = \mathrm{ENN}_\theta\big(x_i^K(t)\big) \in \mathbb{R}_{>0}^K. \tag{7}$$

Belief and vacuity are computed as:

$$b_{i,k}(t) = \frac{\alpha_{i,k}(t)}{\sum_{\ell=1}^{K}\alpha_{i,\ell}(t)}, \tag{8}$$

$$u_i(t) = \frac{K}{\sum_{\ell=1}^{K}\alpha_{i,\ell}(t)}. \tag{9}$$

Here, $b_i(t) \in \Delta^{K-1}$ represents the belief distribution and $u_i(t) \in (0, 1]$ captures epistemic uncertainty; higher vacuity indicates unfamiliar or out-of-distribution observations.

Uncertainty-Driven Escalation. If $u_i(t) > \tau_{\mathrm{vac}}$, UAV i escalates the observation to the operator. Upon receiving a corrected belief $\mathbf{b}_i'(t)$, vacuity is reset $u_i(t) \leftarrow 0$ and operation resumes. The ENN uses a softplus output layer to ensure $\alpha_{i,k} > 0$ and is trained offline with evidential calibration losses [40]; no retraining is required during deployment.

4.2 Adaptive Trust Modeling and Structural Resilience

This component maintains a dynamic reliability estimate over inter-agent communication, enabling the swarm to attenuate inconsistent peers and preserve coordination under faults.

For each edge $(i, j) \in E(t)$, we maintain consistency counters $C_{ij}(t)$ and $T_{ij}(t)$. To prevent circular feedback from trust-weighted consensus, a robust reference state is computed:

$$\tilde{\mathbf{s}}_i(t) = \mathrm{RobustAgg}\big(\{\mathbf{s}_j(t)\}_{j\in\mathcal{N}_i(t)}\big). \tag{10}$$

Here, RobustAgg is the coordinate-wise median across all neighbor states, chosen to avoid circular dependency: since trust scores depend on consistency with the reference, the reference must be computed without trust weights. The median

tolerates up to f arbitrary values per coordinate, matching the $f < \lfloor N/3 \rfloor$ fault bound; e.g., a spoofed position 3 m from the true cluster is excluded before trust scores are updated, preventing corrupted states from bootstrapping high trust.

Vacuity and trust serve complementary roles: vacuity (u_i) captures self-uncertainty in local perception, while trust (τ_{ij}) captures the reliability of a neighbor's communications. The two interact at the safety override (Eq. 16), where either signal independently triggers the consensus fallback.

Counters update as:

$$C_{ij}(t+\Delta t) = C_{ij}(t) + \mathbf{1}\{\|\mathbf{s}_j(t) - \tilde{\mathbf{s}}_i(t)\|_2 < \delta\}, \quad (11)$$

$$T_{ij}(t+\Delta t) = T_{ij}(t) + 1. \quad (12)$$

Trust evolves according to:

$$\tau_{ij}(t+\Delta t) = \frac{C_{ij}(t+\Delta t)+\kappa}{T_{ij}(t+\Delta t)+\kappa} \cdot \exp\left[-\beta(t - t_{ij}^{\text{last}})\right]. \quad (13)$$

Decay is applied only if no message is received beyond $\Delta t_{\text{timeout}}$, preventing unintended trust collapse during short outages. If $\tau_{ij} < \zeta$, edge (i, j) is removed, isolating unreliable agents and reinforcing structural resilience.

4.3 Consensus-Based Safety Arbitration

This component translates distributed beliefs and trust scores into safe collective actions, ensuring that local proposals remain aligned with group consensus and uncertainty-aware safeguards.

Trust-Weighted Fusion. Each UAV i aggregates neighbor information using trust-weighted averaging:

$$\bar{\mathbf{s}}_i = \frac{\sum_{j\in\mathcal{N}_i}\tau_{ij}\mathbf{s}_j}{\sum_{j\in\mathcal{N}_i}\tau_{ij}},\ \bar{b}_i = \frac{\sum_{j\in\mathcal{N}_i}\tau_{ij}b_j}{\sum_{j\in\mathcal{N}_i}\tau_{ij}},\ \bar{p}_i = \frac{\sum_{j\in\mathcal{N}_i}\tau_{ij}p_j}{\sum_{j\in\mathcal{N}_i}\tau_{ij}}. \quad (14)$$

If $\sum_{j\in\mathcal{N}_i}\tau_{ij} < \eta$, UAV i falls back to local estimates, preventing unstable decisions when trusted neighbors are unavailable.

Divergence-Based Safety Override. Let $p_i(t)$ denote the local action distribution and $\bar{p}_i(t)$ the fused distribution. We define the one-way KL divergence:

$$D_{\text{KL}}(p_i\|\bar{p}_i) = \sum_a p_{i,a} \ln \frac{p_{i,a}}{\bar{p}_{i,a}}. \quad (15)$$

A safety override is enforced if

$$D_{\text{KL}}(p_i\|\bar{p}_i) > \varepsilon \quad \text{or} \quad u_i > \tau_{\text{vac}}. \quad (16)$$

In this case,

$$a_i(t) = \arg\max_a \bar{p}_{i,a}(t), \quad (17)$$

otherwise $a_i(t) \sim p_i(t)$. This rule replaces extreme or uncertainty-driven proposals with a consensus action, preserving safety without eliminating decentralized autonomy.

4.4 Resilient Agent Lifecycle Management

This component ensures continuity of swarm operation by isolating unreliable agents, redistributing responsibilities, and safely reintegrating recovered nodes.

Quarantine and Coverage Reassignment. When agent j is quarantined, its patrol region is redistributed using a max-heap $\mathcal{H}$ keyed by capacity score

$$w_k(t) = \lambda_1 E_k(t) + \lambda_2 C_k^{\text{avail}}(t) - \lambda_3 L_k(t), \quad (18)$$

where $E_k(t) \in [0,1]$ denotes the remaining battery charge of UAV k normalized by its full capacity, $C_k^{\text{avail}}(t) \in [0,1]$ represents its available operational capacity computed as the fraction of onboard sensing and computation resources not currently committed to active tasks, and $L_k(t) \in [0,1]$ encodes the ratio of waypoints currently assigned to UAV k relative to the maximum single-agent coverage limit. The nonnegative weights $\lambda_1, \lambda_2, \lambda_3$ control the tradeoff among endurance, spare capacity, and task load. Intuitively, agents with more energy and idle resources but fewer existing assignments are preferred as reassignment targets. The reassignment process executes in $O(\log N)$ time via a single heap extraction and update per reassigned waypoint, ensuring scalable redistribution as swarm size increases.

Reintegration. A quarantined UAV may rejoin after a cooldown T_q if it satisfies both low vacuity ($u_j \le \tau_{\text{vac}}$) and behavioral consistency with neighbors for m consecutive steps. Upon reintegration, trust is restored gradually from τ_{init} to prevent abrupt structural shifts.

Algorithm 1 outlines the per-timestep execution of SHIELD. At each control interval Δt, the procedure updates uncertainty, adapts trust, performs trust-weighted fusion, and enforces safety arbitration before execution. This yields a decentralized, trust-aware control loop in which routine decisions remain within the swarm, while sustained uncertainty or divergence triggers operator escalation.

5 Experimental Setup

This section outlines the task, simulation environment, adversarial setup, evaluation metrics, human-feedback protocol, baselines, and implementation details for SHIELD.

5.1 Operational Task and Environment

Operational Task. We evaluate SHIELD on a waypoint-surveillance mission requiring safe navigation. Each UAV must sequentially visit assigned waypoints while avoiding collisions with obstacles and other agents, then return to a designated home position. Because waypoints are spatially distributed across the arena, agents must coordinate and share local observations to prevent redundant coverage. When a UAV is quarantined (Section 4), its remaining waypoints are

Algorithm 1: SHIELD Per-Timestep Update

Input: Agent i, graph $\mathcal{G}(t)$, local state $\mathbf{s}_i(t)$, sensor $x_i(t)$, policy π_i

1. Evidential Perception

$\boldsymbol{\alpha}_i \leftarrow \mathrm{ENN}_\theta(x_i(t))$

$(b_i, u_i) \leftarrow \mathrm{DirichletBeliefVacuity}(\boldsymbol{\alpha}_i)$

if $u_i > \tau_{\mathrm{vac}}$ **then**

  $b_i \leftarrow \mathrm{QueryHuman}(x_i(t))$

  $u_i \leftarrow 0$

end if

2. Local Action Proposal

$p_i \leftarrow \pi_i(\mathbf{s}_i(t), b_i)$

3. Communication

Broadcast $(\mathbf{s}_i, b_i, p_i)$ to $\mathcal{N}_i$; receive $\{(\mathbf{s}_j, b_j, p_j)\}$

4. Trust Update

$\tilde{\mathbf{s}}_i \leftarrow \mathrm{RobustAgg}(\{\mathbf{s}_j\}_{j\in\mathcal{N}_i})$

for all $j \in \mathcal{N}_i$ **do**

  $C_{ij} \leftarrow C_{ij} + \mathbf{1}_{\|\mathbf{s}_j - \tilde{\mathbf{s}}_i\| < \delta}$

  $T_{ij} \leftarrow T_{ij} + 1$

  $\tau_{ij} \leftarrow \dfrac{C_{ij} + \kappa}{T_{ij} + \kappa}$

  if $t - t_{ij}^{\mathrm{last}} > \Delta t_{\mathrm{timeout}}$ **then**

    $\tau_{ij} \leftarrow \tau_{ij} \cdot e^{-\beta(t - t_{ij}^{\mathrm{last}})}$

  end if

  if $\tau_{ij} < \zeta$ **then**

    Remove edge (i, j) from $\mathcal{G}(t)$

    Quarantine(j); ReassignCoverage(j)

  end if

end for

5. Trust-Weighted Fusion

if $\sum_{j\in\mathcal{N}_i} \tau_{ij} \geq \eta$ **then**

  $\bar{p}_i \leftarrow \mathrm{TrustWeightedAvg}(\{p_j, \tau_{ij}\})$

else

  $\bar{p}_i \leftarrow p_i$ ▷ fallback when insufficient trusted neighbors

end if

6. Safety Arbitration

if $D_{\mathrm{KL}}(p_i \| \bar{p}_i) > \varepsilon$ **or** $u_i > \tau_{\mathrm{vac}}$ **then**

  $a_i \leftarrow \arg\max \bar{p}_i$

else

  $a_i \sim p_i$

end if

7. Execution

Execute a_i and update $\mathbf{s}_i(t + \Delta t)$

Output: Updated state $\mathbf{s}_i(t + \Delta t)$

reassigned to healthy neighbors, testing the swarm's ability to maintain coverage under agent attrition. This task reflects practical drone applications such as perimeter monitoring, infrastructure inspection, and search-and-rescue coverage.

Simulation Environment. All experiments are conducted in QuadSwarm, a multi-quadrotor simulator integrated with Sample Factory via a Gym-style interface [23]. The simulator runs at 50 Hz and models quadrotor dynamics including motor lag, aerodynamic effects, and bounded Gaussian sensor noise ($\sigma_{\mathrm{GPS}} = 0.3$ m, $\sigma_{\mathrm{IMU}} = 0.01$ rad/s). The arena is a $20\,\mathrm{m} \times 20\,\mathrm{m} \times 10\,\mathrm{m}$ volume. In nominal settings the arena is obstacle-free; in obstacle tests, 20% of the volume contains static OctoMap occluders. Each UAV observes its two nearest neighbors through a 90° depth camera. Episodes start from randomized positions within a 2 m radius and terminate upon goal completion, crash, or after 2.5×10^7 environment steps.

5.2 Adversarial Parameterization

Adversarial experiments instantiate the six attack classes defined in Section 3. Table 3 summarizes their parameterization. Attacks are injected on a staggered schedule such that no two attack types activate simultaneously on the same agent.

GPS spoofing (A1) injects a uniform displacement $\boldsymbol{\Delta} \sim \mathcal{U}(-3.0, 3.0)^3$ m for 5 s approximately every 20 s. Communication jamming (A2) independently drops inter-agent messages with probability 0.5 during 3–7 s intervals. Byzantine faults (A3) replace 10% of outgoing state messages with samples drawn from the global state distribution. Replay attacks (A4) provide observations delayed by 10 s for 5 s episodes occurring roughly every 25 s. Malware injection (A5) induces intermittent policy drift bounded by $\|\Delta\pi\| \leq 0.15$, causing selective waypoint omission or heading deviation. Network intrusion (A6) enables passive monitoring of one mesh segment and selective injection of crafted state messages that mimic legitimate agents.

In all trials, at most $f = 2$ of $N = 8$ agents are simultaneously compromised, consistent with the $f < \lfloor N/3 \rfloor$ constraint in Section 3.

5.3 Evaluation Metrics

We report three core mission metrics, each computed per episode and averaged over 30 independent evaluation trials. Let T denote the episode length and N the number of UAVs in the swarm.

(1) Crash Penalty (CP).

$$\mathrm{CP} = \sum_{t=1}^{T} r_t^{\mathrm{crash}}, \tag{19}$$

where $r_t^{\mathrm{crash}} < 0$ is applied whenever a UAV collides with an obstacle or another agent. CP captures cumulative safety degradation over the entire episode; values closer to zero indicate fewer or less severe collisions.

(2) Mission Success Rate (MSR).

$$\mathrm{MSR} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\mathrm{goal}_i \wedge \neg\mathrm{crash}_i\}, \tag{20}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function and $goal_i$ indicates that UAV i has visited all assigned waypoints and returned to its home position. MSR measures the fraction of agents that complete all assigned waypoints without crashing in a given episode. It jointly reflects task completion and safety compliance and is reported as a proportion in $[0, 1]$.

(3) Survival Rate (SR).

$$\mathrm{SR} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\neg\mathrm{crash}_i\}, \tag{21}$$

which measures the fraction of agents that remain crash-free at episode termination. SR isolates safety performance from mission success and is reported in $[0,1]$.

For intrusion-robustness experiments (Exp. 2), we additionally evaluate detection and recovery performance.

(4) Detection Rate (DR).

$$\mathrm{DR} = \frac{\text{Number of correctly flagged attacks within 5 s}}{\text{Total number of injected attacks}}, \quad (22)$$

measuring the ability of the Intrusion Response System to identify attacks shortly after onset.

(5) False Positive Rate (FPR).

$$\mathrm{FPR} = \frac{\text{Number of benign timesteps flagged as attacks}}{\text{Total benign timesteps}}, \quad (23)$$

quantifying how often normal behavior is incorrectly classified as adversarial.

(6) Recovery Time (RT).

$$\mathrm{RT} = \min\{\Delta t : \|\mathbf{s}(t+\Delta t) - \mathbf{s}_{\mathrm{nom}}\| \leq 0.1\|\mathbf{s}_{\mathrm{nom}}\|\}, \quad (24)$$

where $\mathbf{s}_{\mathrm{nom}}$ denotes nominal formation state. RT captures how quickly the swarm returns within 10% of nominal formation after an attack ends.

(7) Human Intervention Count (HIC).

$$\mathrm{HIC} = \sum_{t=1}^{T} \mathbf{1}\{\mathrm{escalation}_t = 1\}, \quad (25)$$

representing the number of binary operator confirmations per episode. Lower HIC indicates reduced operator workload and stronger autonomous filtering.

5.4 Human Feedback Protocol

In all experiments, the human operator is modeled as a scripted oracle with access to ground-truth agent states and attack labels, consistent with prior human-in-the-loop RL evaluations [36, 43]. When SHIELD triggers escalation (vacuity $u_i > \tau_{\mathrm{vac}}$ or divergence $D_{\mathrm{KL}} > \varepsilon$), the oracle issues a binary accept/reject response within a fixed 0.5 s latency (25 simulator steps). Baselines receive comparable feedback under identical timing constraints: COACH receives corrective targets, TAMER receives scalar reward feedback, and SAHO blends human input at the same trigger frequency. Interventions are event-driven rather than continuous, and no method is granted unlimited human input. We adopt a binary interface to reduce cognitive load under supervisory time pressure [7].

5.5 Baselines

We compare five methods under identical observation and action spaces, training budget (2.5×10^7 steps), and simulation backend to ensure fair evaluation.

(1) Advantage Actor-Critic with Policy Optimization (APPO) [39]. APPO (also referred to as *Baseline DRL*) uses pure deep reinforcement learning without human input. The policy is trained end-to-end from environmental rewards and serves as a fully autonomous reference model without uncertainty modeling, trust adaptation, or intervention mechanisms.

(2) Corrective Advice Communicated by Humans (COACH) [36]. COACH incorporates human corrective feedback by treating intervention signals as supervised action targets. When triggered, human input directly adjusts the policy toward corrected behavior, enabling interactive refinement without structured uncertainty reasoning or trust adaptation.

(3) Shared Autonomy via Human Override (SAHO) [27]. SAHO blends human input with autonomous control using a cost-to-go formulation under a latent-goal partially observable Markov decision process (POMDP). Human guidance is integrated into action selection, supporting shared autonomy without explicit intrusion detection or trust-weighted fusion.

(4) Training an Agent Manually via Evaluative Reinforcement (TAMER) [43]. TAMER learns a predictive reward model from scalar human feedback and optimizes behavior using the learned reward. The approach improves learning efficiency but relies on reactive shaping rather than proactive uncertainty-driven escalation or structural resilience.

(5) SHIELD. Our proposed framework augments deep reinforcement learning with evidential perception, dynamic trust-weighted consensus, divergence-based safety arbitration, and an embedded intrusion-response mechanism to enable scalable and trustworthy human-swarm teaming.

These baselines represent four paradigms: autonomous DRL, corrective demonstrations, shared autonomy blending, and human reward shaping. Table 1 summarizes which capabilities each method provides; notably, only SHIELD integrates all four components. The HMT baselines were designed for benign settings and do not include attack detection, which is precisely the gap SHIELD addresses. While recent robust multi-agent communication methods exist (e.g., active defense [47], Bayesian-game MARL [33], targeted trust-based communication [41]), they focus on inter-agent message robustness rather than the human-swarm teaming loop central to this work. Their relationship to SHIELD is discussed in Section 2.

5.6 Implementation Details

Each UAV employs an evidential neural network (ENN) for perception, consisting of a four-layer convolutional backbone (32–64–128–256 filters with batch normalization and ReLU activations) followed by two fully connected layers. The network outputs $K = 10$ Dirichlet concentration parameters via a softplus activation, corresponding to the semantic classes defined in Table 17 in Appendix A: five security states (NORMAL, SUSPICIOUS, ATTACK, MALWARE, INTRU-

Table 1: Capability comparison across methods.

Method	Uncertainty	Trust	Intrusion Det.	Human Esc.
APPO [39]	✘	✘	✘	✘
COACH [36]	✘	✘	✘	▲
SAHO [27]	✘	✘	✘	▲
TAMER [43]	✘	✘	✘	▲
SHIELD	✔	✔	✔	✔

Note: ✔indicates full support of the capability, ▲indicates partial support, and ✘indicates no support.

SION) and five environmental states (CLEAR_PATH, OBSTACLE, CONGESTED, EMERGENCY, UNKNOWN). These parameters are normalized into a belief vector with an associated vacuity score.

The control policy is a multilayer perceptron that takes the 10-dimensional belief vector concatenated with a 24-dimensional local state vector (position, velocity, orientation, angular velocity, and neighbor encodings). It passes through three hidden layers (256–256–128 units with tanh activations) and outputs $A = 7$ discrete thrust commands via softmax. All methods are trained using APPO for 2.5×10^7 steps at 50 Hz with minibatch size 64, 8 epochs per update, learning rate 3×10^{-4}, discount factor $\gamma = 0.99$, Generalized Advantage Estimation (GAE) parameter $\lambda = 0.95$, value loss coefficient 0.5, and entropy coefficient 0.01.

The six SHIELD thresholds ($\tau_{\text{vac}} = 0.4$, $\kappa = 0.1$, $\beta = 0.05$, $\zeta = 0.2$, $\varepsilon = 0.3$, $\delta = 0.15$) are selected via grid search on a held-out validation split and remain fixed across experiments. Sensitivity analysis in Section 6 shows robustness to $\pm 20\%$ perturbations. All UAVs share ENN and policy weights via parameter sharing. Policies are checkpointed every 5×10^5 steps and evaluated over 30 independent trials per configuration. Per-agent inference latency is 4.2 ms on a single GPU with 8 UAVs, well within the 20 ms control budget at 50 Hz.

Experimental Scenarios. We evaluate SHIELD under three complementary protocols designed to assess nominal performance, robustness under attack, and scalability.

(1) Nominal Benchmarking. All five methods are evaluated in open-field and dense-field (20% obstacle) settings with $N = 8$ UAVs. We measure convergence and steady-state performance using CP, MSR, and SR over 2.5×10^7 steps.

(2) Intrusion Robustness. We activate six attack classes (A1–A6, Table 3) in dense-field settings and compare three variants: full SHIELD with IRS, SHIELD without IRS, and a quarantine-only configuration. We report Detection Rate (DR), False Positive Rate (FPR), Recovery Time (RT), and Human Intervention Count (HIC) aggregated across attacks.

(3) Ablation, Scalability, and Sensitivity. We disable individual SHIELD modules (vacuity escalation, trust updates, or safety override) and vary swarm size ($N \in \{8, 16, 32\}$) and obstacle density ($10\%, 20\%, 30\%$). Performance is evaluated via CP, MSR, and SR, and per-step computational overhead (inference time, message size, memory) is reported to assess deployment scalability.

6 Experimental Results & Analyses

We organize results around the three experimental scenarios defined in Section 5, analyzing nominal performance, robustness under adversarial attacks, and scalability through ablation and sensitivity studies.

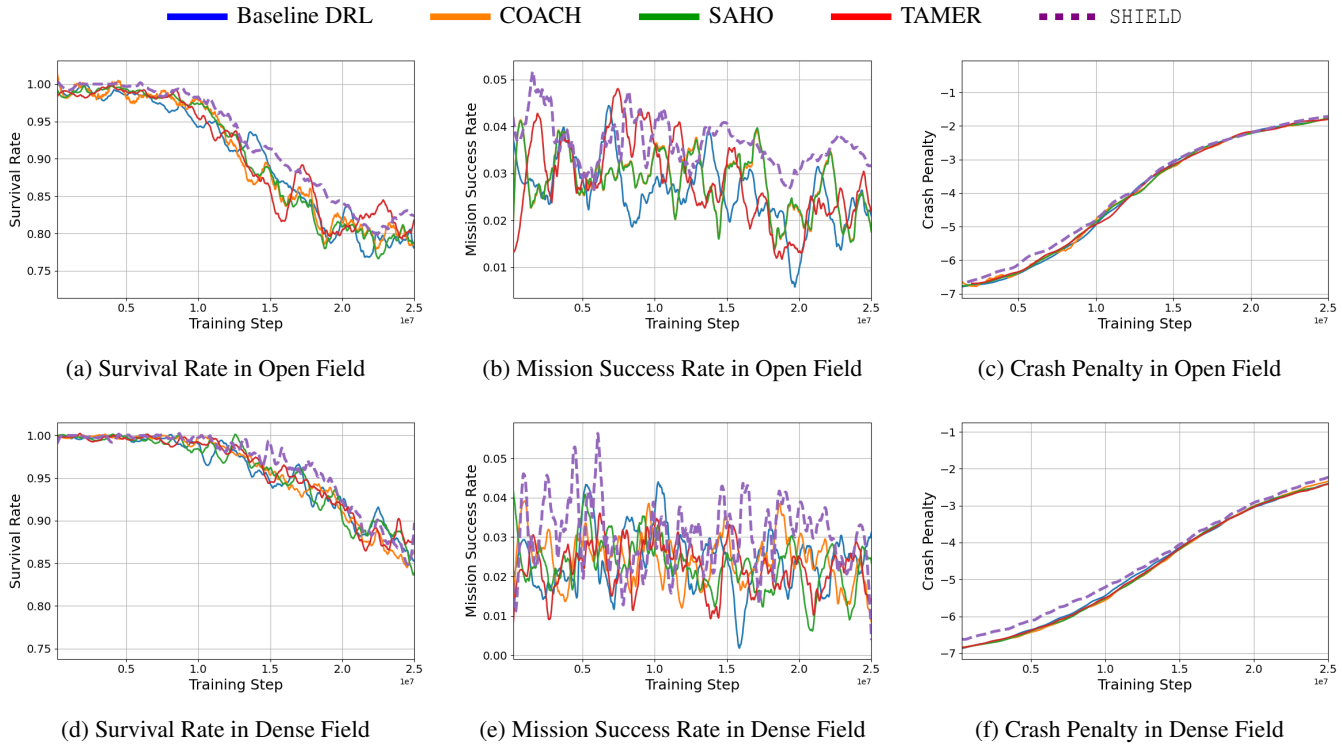
6.1 Nominal Performance and Convergence Analysis

Figure 2 reports *training-time* trajectories for Survival Rate (SR), Mission Success Rate (MSR), and Crash Penalty (CP) across five methods in open-field and dense-field settings. Unless otherwise noted, tables in this section present *test-time* metrics averaged over 30 independent evaluation trials; absolute SR values in Tables 4–6 therefore differ in scale from the training curves. Training-time SR (Figure 2) includes exploratory episodes with high-risk actions, inflating crash rates, while test-time SR (tables) uses deterministic evaluation from the converged checkpoint over 30 trials. The two regimes are standard in deep RL and should not be compared directly.

Open Field. In the obstacle-free setting (Figure 2(a)–(c)), all methods begin with near-perfect survival and progressively adopt more exploratory behaviors. By the end of training, SHIELD stabilizes near 82% SR, exceeding the 78–79% achieved by Baseline DRL, COACH, SAHO, and TAMER. Although MSR remains low due to task difficulty (all 8 UAVs must complete full waypoint sequences without collision), SHIELD consistently leads: it peaks above 5% early in training and converges to approximately 3.5–4.0%, while competing methods settle around 2.0–2.5%. CP improves from roughly -7.0 to -1.8 across methods, with SHIELD achieving comparable or slightly better final values. These results confirm that trust adaptation and uncertainty-driven escalation do not impede learning under benign conditions.

Dense Field. With 20% obstacle occupancy (Figure 2(d)–(f)), collision frequency increases and MSR declines for all methods. Nevertheless, SHIELD maintains a consistent advantage. Final SR reaches approximately 86%, compared to 84–85% for baselines. MSR peaks between 4–6% and converges around 2.5–3.0%, outperforming alternatives below 2.0%. CP decreases steadily across methods, with SHIELD showing sustained improvement after 1×10^7 steps.

Convergence Characteristics. The performance gap is most pronounced during early and mid-training (before 1.5×10^7



(a) Survival Rate in Open Field (b) Mission Success Rate in Open Field (c) Crash Penalty in Open Field

(d) Survival Rate in Dense Field (e) Mission Success Rate in Dense Field (f) Crash Penalty in Dense Field

Figure 2: **Training Dynamics Across Environments.** Each panel plots the smoothed evolution (2.5×10^7 training steps) of one performance metric in two settings. **(a)–(c)** show an obstacle-free "Open Field," where UAVs only contend with downwash and dynamics. **(d)–(f)** repeat the same metrics in a "Dense Field" containing 20 % static OctoMap obstacles and limited neighbor visibility. **Survival Rate** measures the fraction of crash-free agents per episode, **Mission Success Rate** counts the fraction of agents completing all waypoints without collision, and **Crash Penalty** is the cumulative negative reward from impacts (less negative is better).All values are proportions in $[0, 1]$ except CP which is unbounded negative. These are training-time metrics; test-time evaluation metrics (Tables 4–6) are computed separately from the final checkpoint and differ in scale.

steps), when uncertainty guides exploration. As policies converge and obstacle avoidance stabilizes, SR and CP differences narrow; however, SHIELD retains a persistent MSR advantage. We note that the narrowing of SR and CP gaps at convergence is expected: all methods eventually learn collision avoidance in this arena. The practical significance of SHIELD's advantage lies in two aspects. First, the *early-phase* safety gap matters for deployment, where policies operate under partial adaptation or distribution shift rather than full convergence. Second, SHIELD's persistent MSR advantage (about 50–75% relative improvement over baselines at convergence) indicates that trust-weighted coordination improves task completion even after basic collision avoidance is learned.

6.2 Intrusion Detection Performance

Table 2 reports Detection Rate (DR), False Positive Rate (FPR), Recovery Time (RT), and Human Intervention Count (HIC) for four defense variants under the six attack classes (A1–A6) defined in Section 3 and parameterized in Table 3. Results are aggregated across attack types. For variants without an explicit IRS alarm stream, Detection Rate is computed consistently: an attack is detected if a defensive action (quarantine or override) occurs within 5 s of onset. For SHIELD+IRS, detection corresponds to an IRS alarm optionally confirmed by the operator.

Overall Performance. Without IRS, SHIELD relies solely on trust decay, detecting only 12.0% (±6.16) of intrusions and requiring 28.7 human confirmations per episode. The vacuity-only variant increases detection to 70.4% but incurs 18.3% false positives, 20.5 s recovery time, and 20.8 interventions; the high detection variance (±17.59) indicates unstable alerting. Quarantine-only achieves 58.2% detection with faster recovery (15.4 s) but suffers 22.4% false positives and 19.8 confirmations. In contrast, full SHIELD+IRS achieves 82.5% detection (±3.45), 14.6% false positives (±1.56), 9.3 s recovery (±2.02), and 13.5 confirmations (±1.03), while exhibiting the lowest variance across all metrics. These results demonstrate both higher robustness and more stable behavior under diverse attack conditions.

False Positive Characteristics. As shown in Table 2, the 14.6% false positive rate of SHIELD+IRS is lower than vacuity-

Table 2: Intrusion robustness in the dense-field scenario (20% static obstacles) with 8 UAVs. Values are mean$_{\pm\text{std}}$ over 30 randomized attack trials. Dash (–) indicates metric not applicable for SHIELD w/o IRS, since without an active IRS there are no automated alarms (hence no false positives) nor recovery actions to time.

Variant	Detection Rate (%)	False Positive Rate (%)	Recovery Time (s)	Human Interventions
SHIELD w/o IRS	$12.04_{\pm 6.16}$	–	–	$28.67_{\pm 4.12}$
Vacuity Only IDS	$70.36_{\pm 17.59}$	$18.34_{\pm 5.72}$	$20.45_{\pm 4.56}$	$20.78_{\pm 2.30}$
Quarantine Only	$58.21_{\pm 8.45}$	$22.37_{\pm 6.31}$	$15.35_{\pm 9.11}$	$19.82_{\pm 3.19}$
SHIELD + IRS	$\mathbf{82.52}_{\pm 3.45}$	$\mathbf{14.55}_{\pm 1.56}$	$\mathbf{9.25}_{\pm 2.02}$	$\mathbf{13.47}_{\pm 1.03}$

Table 3: Attack parameterization for intrusion-robustness experiments. All attacks follow the threat model in Section 3.

Attack	**Duration / Interval**	**Severity**	**Target**
A1: GPS Spoofing	5 s on / 20 s off	$\delta_{\max} = 3.0$ m	1 random UAV
A2: Comm. Jamming	3–7 s window	Drop prob. $p = 0.5$	2 random links
A3: Byzantine Faults	Continuous	Inject prob. $p = 0.10$	1 compromised UAV
A4: Replay Attack	5 s on / 25 s off	Lag $\Delta t_r = 10$ s	1 random UAV
A5: Malware Injection	Continuous	Policy drift $\|\Delta\pi\| \leq 0.15$	1 compromised UAV
A6: Network Intrusion	Continuous	Eavesdrop + selective inject	1 network segment

only (18.3%) and quarantine-only (22.4%). Most false alarms stem from legitimate operational hazards such as obstacle encounters or congestion that temporarily elevate vacuity. The joint security–environmental ENN classification (Section 4), separating five security from five environmental states, mitigates misattribution by distinguishing environmental anomalies from adversarial events. The low standard deviation ($\pm 1.56\%$) further indicates consistent alert behavior.

Operator Workload Reduction. SHIELD+IRS reduces human confirmations by 53% compared to the no-IRS variant (13.5 vs. 28.7 per episode). IRS resolves events when vacuity, trust decay, and attack signatures are consistent, escalating only ambiguous cases. The event-triggered binary interface keeps intervention frequency low and predictable, with variance limited to ± 1.03. Under nominal (attack-free) conditions, Table 2 shows fewer than five queries per 500 s episode. Given sub-second binary decision times and error rates below 5% under time pressure [14], total operator engagement remains under 5 s per mission, lower than continuous shared-autonomy paradigms.

Table 4: Component ablation in dense field (20% obstacles) using 8 UAVs. Results are mean$_{\pm\text{std}}$ over 30 trials.

Variant	CP	MSR	SR
Full SHIELD	$\mathbf{-3.85}_{\pm 0.45}$	$\mathbf{0.040}_{\pm 0.006}$	$\mathbf{0.120}_{\pm 0.012}$
w/o Vacuity	$-4.35_{\pm 0.50}$	$0.032_{\pm 0.005}$	$0.095_{\pm 0.010}$
w/o Trust	$-4.62_{\pm 0.55}$	$0.025_{\pm 0.005}$	$0.081_{\pm 0.015}$
w/o Safety	$-5.19_{\pm 0.60}$	$0.018_{\pm 0.004}$	$0.077_{\pm 0.026}$

Table 5: Effect of obstacle density with 8 UAVs on SHIELD performance. Values are mean$_{\pm\text{std}}$ over 30 trials.

Density (%)	CP	MSR	SR
10	$\mathbf{-3.67}_{\pm 0.42}$	$\mathbf{0.045}_{\pm 0.004}$	$\mathbf{0.138}_{\pm 0.015}$
20	$-3.85_{\pm 0.45}$	$0.040_{\pm 0.006}$	$0.120_{\pm 0.012}$
30	$-4.17_{\pm 0.50}$	$0.035_{\pm 0.006}$	$0.103_{\pm 0.019}$

Table 6: Scalability of SHIELD in dense field (20% obstacles). Values are mean$_{\pm\text{std}}$ over 20 trials.

Swarm Size	CP	MSR	SR
8	$\mathbf{-3.85}_{\pm 0.45}$	$\mathbf{0.040}_{\pm 0.006}$	$\mathbf{0.120}_{\pm 0.012}$
16	$-3.99_{\pm 0.48}$	$0.026_{\pm 0.006}$	$0.115_{\pm 0.018}$
32	$-4.02_{\pm 0.58}$	$0.021_{\pm 0.002}$	$0.114_{\pm 0.016}$

6.3 Ablation, Scalability, and Sensitivity

Component Ablation. Table 4 isolates the contribution of each SHIELD module in the dense-field setting (20% obstacles, $N = 8$). Removing vacuity escalation degrades CP from -3.85 to -4.35 and reduces MSR by 20% relative (0.040 to 0.032), confirming that evidential uncertainty is critical for early risk detection. Disabling trust updates causes further degradation (CP -4.62, MSR 0.025, SR 0.081), indicating that peer filtering prevents cascading coordination failures. Removing safety overrides produces the largest drop (CP -5.19, MSR 0.018, SR 0.077) and increases variance (± 0.026 SR versus ± 0.012 for full SHIELD), demonstrating that KL-based arbitration is the final safeguard against catastrophic actions. The monotonic performance decline across ablations confirms that modules contribute complementary, non-redundant

Table 7: Vacuity threshold τ_{vac} sensitivity.

τ_{vac}	10% density			20% density			30% density		
	MSR	CP	SR	MSR	CP	SR	MSR	CP	SR
0.32	0.040	−3.30	0.150	0.035	−3.60	0.130	0.030	−4.00	0.110
0.40	0.045	−3.67	0.138	0.040	−3.85	0.120	0.035	−4.17	0.103
0.48	0.042	−4.10	0.125	0.037	−4.00	0.110	0.033	−4.30	0.095

Table 8: Trust-decay rate β sensitivity.

β	10%			20%			30%		
	MSR	CP	SR	MSR	CP	SR	MSR	CP	SR
0.04	0.046	−3.60	0.140	0.042	−3.78	0.122	0.036	−4.10	0.106
0.05	0.045	−3.67	0.138	0.040	−3.85	0.120	0.035	−4.17	0.103
0.06	0.043	−3.75	0.135	0.038	−3.92	0.117	0.033	−4.24	0.100

Table 9: KL-divergence threshold ε sensitivity.

ε	10%			20%			30%		
	MSR	CP	SR	MSR	CP	SR	MSR	CP	SR
0.24	0.042	−3.40	0.145	0.038	−3.65	0.128	0.032	−4.05	0.108
0.30	0.045	−3.67	0.138	0.040	−3.85	0.120	0.035	−4.17	0.103
0.36	0.044	−3.90	0.132	0.039	−3.98	0.115	0.034	−4.30	0.098

benefits.

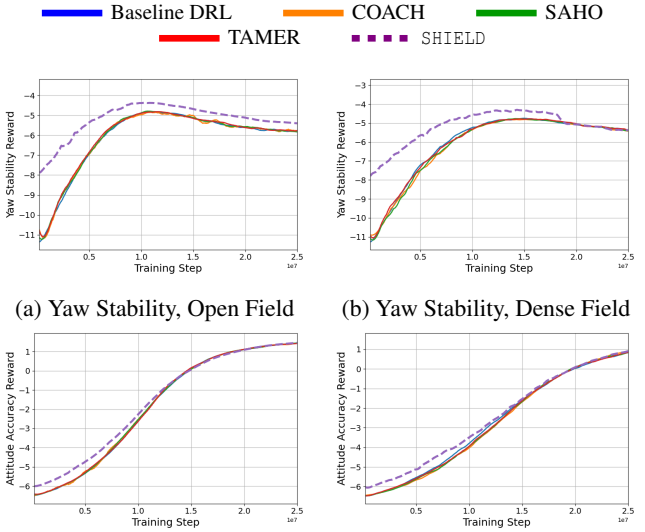
Obstacle Density Robustness. Table 5 shows graceful degradation as obstacle density increases from 10% to 30%. CP shifts from -3.67 to -4.17, MSR declines from 0.045 to 0.035, and SR decreases from 0.138 to 0.103. The contained performance loss across a threefold increase in obstacle density indicates that trust-weighted consensus adapts effectively under constrained visibility and higher collision risk.

Scalability with Swarm Size. Table 6 evaluates swarm sizes $N = \{8, 16, 32\}$. SR remains stable (0.120 to 0.114), suggesting decentralized trust scaling without coordination collapse. MSR decreases from 0.040 to 0.021, reflecting higher task competition as agents increase within a fixed arena. CP degrades marginally (-3.85 to -4.02), and variance remains bounded, indicating stable behavior at larger scales.

Threshold Sensitivity. Threshold robustness is evaluated by sweeping each parameter $\pm 20\%$ from its default. Detailed results are reported in Tables 7–10. Performance remains stable across all sweeps, confirming that SHIELD is not brittle to moderate parameter perturbations. The vacuity threshold τ_{vac} exhibits the greatest sensitivity: at 30% density, increasing from 0.40 to 0.48 reduces SR by 8% and worsens CP by 12% (Table 7). Trust-decay rate β induces less than 4% variation across metrics (Table 8). The KL threshold ε and smoothing factor κ similarly show bounded impact (Tables 9 and 10). In practice, $\tau_{vac} \in [0.35, 0.45]$ and $\varepsilon \in [0.25, 0.35]$ maintain near-optimal performance without extensive retuning.

Attack Parameter Sensitivity. To verify that detection performance is not sensitive to specific attack parameterizations, we additionally swept each attack parameter $\pm 50\%$ from its Table 3 default: $\delta_{max} \in \{1.5, 3.0, 4.5\}$ m, jamming drop probability $\in \{0.25, 0.5, 0.75\}$, Byzantine injection probability $\in \{0.05, 0.10, 0.15\}$, and replay lag $\in \{5, 10, 15\}$ s. Across all configurations, SHIELD+IRS maintains DR in $[78.3, 86.1]\%$ and FPR in $[12.8, 16.4]\%$. Stronger attacks (higher δ_{max}, higher jamming probability) paradoxically increase DR because larger deviations are easier to detect, at the cost of modestly higher FPR. Weaker attacks reduce both DR and FPR. Recovery time varies between 7.8–11.2 s across settings. These results confirm that the reported conclusions are not artifacts of specific parameter choices.

Computational Overhead. Runtime and communication statistics are summarized in Tables 11–13. The full SHIELD pipeline operates within the 20 ms control budget at 50 Hz. Per-agent inference time increases sublinearly with swarm size due to batched GPU execution, remaining 7.8 ms at $N = 64$ (Table 11). Message rates remain below 300 pps (Table 12), well within typical mesh network capacity. Memory overhead scales linearly with neighborhood size but remains modest (Table 13). These results confirm deployment feasibility without computational bottlenecks.



(a) Yaw Stability, Open Field (b) Yaw Stability, Dense Field

(c) Attitude Accuracy, Open Field (d) Attitude Accuracy, Dense Field

Figure 3: Yaw Stability and Attitude Accuracy rewards over training. SHIELD achieves consistently higher yaw stability than baselines, particularly under obstacle clutter, while converging to comparable attitude accuracy ($\sim$1.0), confirming that its safety mechanisms do not degrade orientation control.

Additional Stability Metrics. We report yaw stability and attitude accuracy to verify that SHIELD's safety and trust mechanisms do not degrade low-level flight dynamics (Figure 3).

Yaw stability rewards (Figures 3(a)–(b)) show that SHIELD achieves higher stability than baselines, particularly in dense environments where collision avoidance and consensus arbitration are most active. In obstacle settings, SHIELD maintains

Table 10: Trust-smoothing κ sensitivity.

κ	10%			20%			30%		
	MSR	CP	SR	MSR	CP	SR	MSR	CP	SR
0.08	0.043	−3.78	0.133	0.038	−3.92	0.115	0.033	−4.25	0.098
0.10	0.045	−3.67	0.138	0.040	−3.85	0.120	0.035	−4.17	0.103
0.12	0.047	−3.55	0.142	0.042	−3.75	0.125	0.037	−4.05	0.108

approximately 0.5–1.0 reward points higher yaw stability, indicating smoother rotational control when uncertainty-driven overrides suppress erratic proposals.

Attitude accuracy rewards (Figures 3(c)–(d)) converge to nearly identical final values (approximately 1.0) across all methods in both open and dense fields. This confirms that the additional trust updates, escalation triggers, and KL-based safety overrides introduce no measurable degradation in orientation control or steady-state tracking performance.

Table 11: Per-step runtime (ms) per UAV.

Swarm Size	8	16	32	64
Runtime (ms)	4.2	5.1	6.3	7.8
Std. Dev.	0.3	0.4	0.5	0.7

Table 12: Packets per second (pps) per UAV under consensus.

Swarm Size	8	16	32	64
Sent/Recv pps	200	230	260	290
Std. Dev.	15	18	22	26

7 Discussion and Limitations

Nominal and Adversarial Performance Insights. Across benign and adversarial settings, SHIELD improves learning stability and operational robustness through evidential vacuity, trust adaptation, decentralized fusion, and an IRS (Figure 2). The advantage is most pronounced during early training when uncertainty guides exploration, i.e., the regime most relevant to deployment under distribution shift, while SHIELD retains a persistent MSR advantage at convergence.

Layered Defense and Robustness–Sensitivity Trade-offs. Across the six attack classes, full SHIELD+IRS achieves the most balanced performance (Table 2), combining strong detection, fewer false positives, faster recovery, and reduced operator confirmations relative to partial variants. Vacuity-only and quarantine-only results reveal a trade-off: greater anomaly sensitivity improves detection but increases false alarms, while aggressive isolation accelerates recovery at the risk of unnecessary quarantines.

Ablation results (Table 4) confirm a layered defense structure: vacuity detects anomalies, trust filters unreliable peers, and KL-based overrides provide containment. Removing modules causes monotonic degradation, indicating complementarity. Reduced variance in detection and recovery further demonstrates stable and predictable defensive behavior essential for safety-critical cyber-physical systems.

Table 13: Per-agent memory footprint.

Component	Memory	Growth
ENN Weights (GPU)	5.0 MB	constant
Trust Counters (CPU)	0.1 MB/neighbor	linear
Total (2 neighbors)	5.2 MB	–

Scalability and Structural Stability. Scalability experiments (Table 6) and obstacle-density sweeps (Table 5) show gradual performance degradation as swarm size and environmental complexity increase. Survival Rate remains stable across larger swarms, while Mission Success Rate declines predictably due to increased coordination load within a fixed arena. Crash Penalty degradation remains bounded.

The decentralized trust graph provides implicit fault containment: compromised agents influence only local neighborhoods until trust decay isolates them. Decisions are formed locally and propagated through neighborhood fusion rather than a centralized controller, limiting bottlenecks and reducing global destabilization as complexity grows.

Threshold Robustness and Parameter Interpretability. Sensitivity sweeps (Tables 7–10) demonstrate bounded performance variation under $\pm 20\%$ perturbations of all thresholds. While the vacuity threshold τ_{vac} exhibits the greatest influence, particularly at higher obstacle densities, trust-decay and smoothing parameters induce only minor variation. These results suggest that thresholds operate within semantically meaningful ranges rather than narrow brittle optima, supporting deployment without exhaustive retuning.

Operator Load Calibration. Full SHIELD+IRS reduces human confirmations by 53% relative to the no-IRS variant while maintaining higher detection performance. This reflects calibrated escalation rather than suppressed alerts: the IRS resolves aligned vacuity, trust, and attack signals, escalating only ambiguous cases. The event-triggered binary interface keeps operator workload predictable, as indicated by low variance in intervention counts.

Practical Acceptability of Detection Performance. The 82.5% DR and 14.6% FPR must be interpreted within SHIELD's layered defense architecture. Undetected attacks (the remaining $\sim$17.5%) are not unmitigated: trust decay and KL-based safety overrides still attenuate impact, as shown by the monotonic degradation when these modules are removed (Table 4). The 14.6% FPR predominantly comprises environmental hazards (obstacles, congestion) that warrant heightened caution; the joint security–environmental ENN classification reduces misattribution relative to single-category

detectors (vacuity-only: 18.3%, quarantine-only: 22.4%). For comparison, reported false positive rates for UAV anomaly detectors range from 10–25% [10, 15], placing SHIELD+IRS at the lower end while also providing trust adaptation and human escalation that standalone detectors lack.

Limitations and Deployment Constraints. Several limitations arise from the current scope. First, evaluation is simulation-based; real deployments may introduce unmodeled aerodynamics, sensing artifacts, and latency affecting vacuity calibration and trust dynamics. Specific concerns include sensor drift affecting ENN calibration, communication latency altering trust decay dynamics, and actuator saturation limiting the effectiveness of consensus overrides. Physical-platform validation is planned as future work.

8 Conclusion and Future Work

This paper presents SHIELD, a unified framework for scalable human–swarm teaming that addresses safety and mission continuity under limited operator attention, uncertain observations, and cyber-physical compromise. SHIELD integrates evidential perception, dynamic trust adaptation, decentralized consensus, and divergence-based safety overrides within a single control loop, supported by an embedded Intrusion Response System that enables quarantine, safe-mode execution, and structured escalation only when necessary. Extensive simulations show improved convergence under nominal conditions and stronger resilience under attacks, including higher mission success, faster recovery, reduced collisions, and lower, more predictable operator workload. Overall, SHIELD demonstrates that tightly coupling uncertainty estimation, decentralized trust, and structured containment improves both robustness and practical deployability in multi-UAV systems.

Future work will extend SHIELD to physical UAV platforms to evaluate robustness under realistic sensing noise, actuator latency, and intermittent communication. We will investigate adaptive thresholding and context-conditioned policies to address nonstationary operating conditions, as well as richer, low-cognitive-load human feedback mechanisms beyond binary confirmation. Additional efforts will focus on communication-efficient trust update protocols for large-scale swarms and systematic evaluation against coordinated multi-agent deception and long-horizon intrusion strategies that explicitly manipulate consensus formation dynamics.

Ethical Considerations. This work is defensive in nature, aiming to improve the security and human oversight of cooperative drone swarms. All experiments are conducted entirely in simulation using the open-source QuadSwarm environment [23], with no physical deployments, real communication disruptions, or human subjects involved. The six attack classes evaluated are well-documented in UAV security literature [3, 8, 28, 32, 35], with parameterizations derived from published field studies. While adversarial strategies are described in sufficient detail for reproducibility, we introduce no novel offensive techniques nor provide artifacts directly applicable to deployed systems. The defensive contributions of this work, including uncertainty-aware intrusion detection, decentralized quarantine, and structured human escalation, substantially outweigh the marginal risk of informing adversarial strategy. We encourage future physical-platform extensions to follow responsible disclosure practices.

Acknowledgments

This work was supported by the Army Research Office under Grant No. W911NF-24-2-0241.

Use of Generative AI

Generative AI tools were used for editorial purposes in this manuscript, and all outputs were inspected by the authors to ensure accuracy and originality.

References

[1] ABDI, S. S., AND PALEY, D. A. Safe operations of an aerial swarm via a cobot human swarm interface. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), IEEE, pp. 1701–1707.

[2] AGRAWAL, A., AND CLELAND-HUANG, J. Explaining autonomous decisions in swarms of human-on-the-loop small unmanned aerial systems. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2021), vol. 9, pp. 15–26.

[3] BABU, G. A., KRISHNA, G. V., TIWARI, V., ANTONY, R., SURAJ, C., VIKAS, K., AND MURTHY, P. Drone-borne magnetic measurements in india. *Current Science 119*, 4 (2020), 598–600.

[4] BIYIK, E., AND SADIGH, D. Batch active preference-based learning of reward functions. In *Conference on robot learning* (2018), PMLR, pp. 519–528.

[5] BROWN, D., GOO, W., NAGARAJAN, P., AND NIEKUM, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning* (2019), PMLR, pp. 783–792.

[6] BROWN, D. S., GOO, W., AND NIEKUM, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning* (2020), PMLR, pp. 330–359.

[7] CHEN, J. Y., AND BARNES, M. J. Human–agent teaming for multirobot control: A review of human factors

issues. *IEEE Transactions on Human-Machine Systems 44*, 1 (2014), 13–29.

[8] CHUNG, S.-J., PARANJAPE, A. A., DAMES, P., SHEN, S., AND KUMAR, V. A survey on aerial swarm robotics. *IEEE Transactions on Robotics 34*, 4 (2018), 837–855.

[9] CUMMINGS, M. L., AND GUERLAIN, S. Developing operator capacity estimates for supervisory control of autonomous vehicles. *Human factors 49*, 1 (2007), 1–15.

[10] DESAI, A., DAI, R., CHEN, Y., HO, K., KEE, A., BULATOVIC, S., SHAFIUZZAMAN, M., BAI, K. Y., JEONG, I. U., SIU, D., ET AL. Stateful behavior inference and runtime enforcement for vehicle network security. In *3rd USENIX Symposium on Vehicle Security and Privacy (VehicleSec 25)* (2025), pp. 1–17.

[11] DING, S., DU, W., DING, L., GUO, L., AND ZHANG, J. Learning efficient and robust multi-agent communication via graph information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 17346–17353.

[12] DIVBAND SOORATI, M., CLARK, J., GHOFRANI, J., TARAPORE, D., AND RAMCHURN, S. D. Designing a user-centered interaction interface for human–swarm teaming. *Drones 5*, 4 (2021), 131.

[13] EBINE, Y., NOMOTO, K., TANAKA, Y., KOBAYASHI, R., TSURUOKA, G., AND MORI, T. {WIP}: Evaluating the {End-to-End} impact of false localization attacks on {vSLAM-Based} autonomous drones. In *3rd USENIX Symposium on Vehicle Security and Privacy (VehicleSec 25)* (2025), pp. 73–80.

[14] ENDSLEY, M. R. Toward a theory of situation awareness in dynamic systems. In *Situational awareness*. Routledge, 2017, pp. 9–42.

[15] ERBA, A., CASTELLANOS, J. H., SIHAG, S., ZONOUZ, S., AND TIPPENHAUER, N. O. {ConfuSense}: Sensor reconfiguration attacks for stealthy {UAV} manipulation. In *3rd USENIX Symposium on Vehicle Security and Privacy (VehicleSec 25)* (2025), pp. 45–62.

[16] FRIEDBERG, I., HONG, X., MCLAUGHLIN, K., SMITH, P., AND MILLER, P. C. Evidential network modeling for cyber-physical system state inference. *IEEE Access 5* (2017), 17149–17164.

[17] FUNG, H. L., DARVARIU, V.-A., HAILES, S., AND MUSOLESI, M. Trust-based consensus in multi-agent reinforcement learning systems. *arXiv preprint arXiv:2205.12880* (2022).

[18] GUO, Y., PANG, Y., LYONS, J., LEWIS, M., SYCARA, K., AND LIU, R. Trust-aware reflective control for fault-resilient dynamic task response in human–swarm cooperation. *AI 5*, 1 (2024), 446–464.

[19] HEJNA III, D. J., AND SADIGH, D. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning* (2023), PMLR, pp. 2014–2025.

[20] HESTER, T., VECERIK, M., PIETQUIN, O., LANCTOT, M., SCHAUL, T., PIOT, B., HORGAN, D., QUAN, J., SENDONARIS, A., OSBAND, I., ET AL. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence* (2018), vol. 32.

[21] HIGHNAM, K., ANGSTADT, K., LEACH, K., WEIMER, W., PAULOS, A., AND HURLEY, P. An uncrewed aerial vehicle attack scenario and trustworthy repair architecture. In *2016 46th annual IEEE/IFIP international conference on dependable systems and networks workshop (DSN-W)* (2016), IEEE, pp. 222–225.

[22] HOU, J., WANG, F., WANG, L., AND CHEN, Z. Reinforcement learning based multi-agent resilient control: From deep neural networks to an adaptive law. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 7737–7745.

[23] HUANG, Z., BATRA, S., CHEN, T., KRUPANI, R., KUMAR, T., MOLCHANOV, A., PETRENKO, A., PREISS, J. A., YANG, Z., AND SUKHATME, G. S. Quadswarm: A modular multi-quadrotor simulator for deep reinforcement learning with direct thrust control. *arXiv preprint arXiv:2306.09537* (2023).

[24] HUMPHREYS, T. E., LEDVINA, B. M., PSIAKI, M. L., O'HANLON, B. W., KINTNER, P. M., ET AL. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Proceedings of the 21st International technical meeting of the satellite division of the institute of navigation (ION GNSS 2008)* (2008), pp. 2314–2325.

[25] HUSSEIN, A., GHIGNONE, L., NGUYEN, T., SALIMI, N., NGUYEN, H., WANG, M., AND ABBASS, H. A. Characterization of indicators for adaptive human-swarm teaming. *Frontiers in Robotics and AI 9* (2022), 745958.

[26] IBARZ, B., LEIKE, J., POHLEN, T., IRVING, G., LEGG, S., AND AMODEI, D. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems 31* (2018).

[27] JAVDANI, S., SRINIVASA, S. S., AND BAGNELL, J. A. Shared autonomy via hindsight optimization. *Robotics science and systems: online proceedings 2015* (2015), 10–15607.

[28] KERNS, A. J., SHEPARD, D. P., BHATTI, J. A., AND HUMPHREYS, T. E. Unmanned aircraft capture and control via gps spoofing. *Journal of field robotics 31*, 4 (2014), 617–636.

[29] KHEN, G., ZHAO, D., AND BACA, J. Intuitive human-swarm interaction with gesture recognition and machine learning. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing* (2023), pp. 453–456.

[30] KHOPKAR, P. Mixed-initiative flexible autonomy in drone swarms for covid-19 applications. In *2020 IEEE International Symposium on Technology and Society (ISTAS)* (2020), IEEE, pp. 457–461.

[31] LEE, K., SMITH, L., AND ABBEEL, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091* (2021).

[32] LEWIS-PYE, A., AND ROUGHGARDEN, T. Byzantine generals in the permissionless setting. In *International Conference on Financial Cryptography and Data Security* (2023), Springer, pp. 21–37.

[33] LI, S., GUO, J., XIU, J., XU, R., YU, X., WANG, J., LIU, A., YANG, Y., AND LIU, X. Byzantine robust cooperative multi-agent reinforcement learning as a bayesian game. *arXiv preprint arXiv:2305.12872* (2023).

[34] LIANG, X., SHU, K., LEE, K., AND ABBEEL, P. Reward uncertainty for exploration in preference-based reinforcement learning. *arXiv preprint arXiv:2205.12401* (2022).

[35] LICHTMAN, M., JOVER, R. P., LABIB, M., RAO, R., MAROJEVIC, V., AND REED, J. H. Lte/lte-a jamming, spoofing, and sniffing: threat assessment and mitigation. *IEEE Communications Magazine 54*, 4 (2016), 54–61.

[36] MACGLASHAN, J., HO, M. K., LOFTIN, R., PENG, B., WANG, G., ROBERTS, D. L., TAYLOR, M. E., AND LITTMAN, M. L. Interactive learning from policy-dependent human feedback. In *International conference on machine learning* (2017), PMLR, pp. 2285–2294.

[37] OLFATI-SABER, R., AND MURRAY, R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control 49*, 9 (2004), 1520–1533.

[38] PANG, Y., AND LIU, R. Trust-aware emergency response for a resilient human-swarm cooperative system. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (2021), IEEE, pp. 15–20.

[39] PETRENKO, A., HUANG, Z., KUMAR, T., SUKHATME, G., AND KOLTUN, V. Sample factory: Egocentric 3d control from pixels at 100000 fps with asynchronous reinforcement learning. In *International Conference on Machine Learning* (2020), PMLR, pp. 7652–7662.

[40] SENSOY, M., KAPLAN, L., AND KANDEMIR, M. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems 31* (2018).

[41] SUN, C., ZANG, Z., LI, J., LI, J., XU, X., WANG, R., AND ZHENG, C. T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 15154–15163.

[42] TANG, X., ZHANG, M., LIU, W., DU, B., AND LIU, Z. Towards a model of human-cyber–physical automata and a synthesis framework for control policies. *Journal of Systems Architecture 144* (2023), 102989.

[43] WARNELL, G., WAYTOWICH, N., LAWHERN, V., AND STONE, P. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence* (2018), vol. 32.

[44] WATTEARACHCHI, W. D., LAKSHIKA, E., KASMARIK, K., AND BARLOW, M. A study on human-swarm interaction: A framework for assessing situation awareness and task performance. *arXiv preprint arXiv:2503.14810* (2025).

[45] WEI, L., WU, J., AND LONG, C. Blockchain-enabled trust management in service-oriented internet of things: Opportunities and challenges. In *Proceedings of the 2021 3rd International Conference on Blockchain Technology* (2021), pp. 90–95.

[46] WU, X., XIAO, L., SUN, Y., ZHANG, J., MA, T., AND HE, L. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems 135* (2022), 364–381.

[47] YU, L., QIU, Y., YAO, Q., SHEN, Y., ZHANG, X., AND WANG, J. Robust communicative multi-agent reinforcement learning with active defense. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 17575–17582.

[48] ZHANG, M., LI, N., LI, J., LIAO, J., AND LIU, J. Memory-based resilient control against non-cooperation in multi-agent flocking. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (2024), pp. 2075–2084.

[49] ZHAO, C., ZHENG, C., ROLDAN, L., SHKURTI, T., NAHARI, A., NEWMAN, W., TYLER, D., LEE, K., AND FU, M. J. Adaptable mixed-reality sensorimotor interface for human-swarm teaming: Person with limb loss case study and field experiments. *Field Robotics 3* (2023), 243–265.

A Implementation Details

A.1 Model Architecture and Design Rationale

We employ a two-stage architecture that decouples *uncertainty-aware perception* from *risk-sensitive decision making*. The perception module is implemented as an Evidential Neural Network (ENN) backbone that produces Dirichlet-distributed class evidence, while the control module is a policy MLP head that integrates ENN beliefs with the UAV state for discrete action selection.

Table 14 details the ENN backbone architecture. The network consists of four convolutional blocks with progressive spatial downsampling, followed by a fully connected embedding layer and a Softplus output layer that produces non-negative Dirichlet concentration parameters. The use of Softplus ensures strictly positive evidence values, enabling principled uncertainty quantification through subjective logic. The final 10-dimensional output corresponds to class-wise evidence, which is later transformed into belief masses and uncertainty measures used by the policy.

Table 14: ENN Backbone Architecture

Layer	Output Shape	Kernel/Stride/Pad	Notes
Conv1+BN+ReLU	$32 \times 64 \times 64$	$3 \times 3/1/1$	BatchNorm, ReLU
Conv2+BN+ReLU	$64 \times 32 \times 32$	$3 \times 3/2/1$	Spatial downsample
Conv3+BN+ReLU	$128 \times 16 \times 16$	$3 \times 3/2/1$	Further downsample
Conv4+BN+ReLU	$256 \times 8 \times 8$	$3 \times 3/2/1$	Final feature map
Flatten	16384	–	–
FC+ReLU	512	–	Hidden embedding
FC+Softplus	10	–	Dirichlet parameters

Table 15 presents the policy head architecture. The policy receives a concatenated input comprising the 10-dimensional ENN belief vector and a 24-dimensional UAV state representation, resulting in a 34-dimensional input. The multilayer perceptron uses three hidden layers with tanh activations to maintain bounded representations and improve training stability. The final Softmax layer outputs a categorical distribution over $A = 7$ discrete thrust commands. This design enables the controller to adapt its action selection based not only on state dynamics but also on epistemic uncertainty encoded by the ENN.

Table 15: Policy MLP Head Architecture

Layer	Output Dim	Notes
Input concat	34	10 ENN belief + 24 state dims
FC1+tanh	256	First hidden layer
FC2+tanh	256	Second hidden layer
FC3+tanh	128	Third hidden layer
FC4+softmax	$A = 7$	Discrete thrust commands

A.2 Training Configuration and SHIELD Control Parameters

We train the policy using PPO with generalized advantage estimation for stable updates and variance reduction. Table 16 reports both the standard RL hyperparameters, which govern optimization, exploration, and critic scaling, and the SHIELD-specific parameters, which control uncertainty escalation, trust adaptation, quarantine, and consensus consistency checks. All results are averaged across independent random seeds, each trained for 2.5×10^7 environment steps.

Table 16: Training and SHIELD-Specific Hyperparameters

Parameter	Value	Description
Learning rate	3×10^{-4}	Adam optimizer
Discount γ	0.99	Return estimation
GAE λ	0.95	Advantage estimation
Minibatch size	64	Samples per update
Epochs/update	8	Passes per minibatch
Entropy coeff.	0.01	Exploration regularization
Value-loss coeff.	0.5	Critic loss weighting
Total steps	2.5×10^7	Per seed
τ_{vac}	0.4	Vacuity escalation threshold
κ	0.1	Trust smoothing constant
β	0.05	Trust decay rate
ζ	0.2	Quarantine activation threshold
ε	0.3	KL-divergence override threshold
δ	0.15	Consensus consistency radius

Table 17: ENN Semantic Classes ($K = 10$)

ID	Class	Category
0	NORMAL	Security
1	SUSPICIOUS	Security
2	ATTACK	Security
3	MALWARE	Security
4	INTRUSION	Security
5	CLEAR_PATH	Environmental
6	OBSTACLE	Environmental
7	CONGESTED	Environmental
8	EMERGENCY	Environmental
9	UNKNOWN	Environmental

A.3 Environment and Attack Configuration

The simulator uses a $20 \times 20 \times 10$ m arena at 50 Hz with downwash ($C_d = 0.1$), motor lag ($\tau = 0.15$ s), and Gaussian sensor noise ($\sigma_{\text{GPS}} = 0.3$ m, $\sigma_{\text{IMU}} = 0.01$ rad/s). Dense-field tests occupy 20% of the volume with 0.5 m cubic OctoMap occluders. Episodes last up to 25,000 steps (500 s). Attack parameterization for the six classes appears in Table 3. Attacks use non-overlapping windows with at most $f = 2$ compromised agents, and per-timestep attack labels are logged for post hoc analysis.

A.4 ENN Semantic Classes

Table 17 lists the $K = 10$ semantic classes. Classes 0–4 capture security states; classes 5–9 capture environmental conditions. The joint taxonomy enables the ENN to distinguish adversarial anomalies from benign operational hazards, reducing false positives in the IRS.

B Theoretical Guarantees

We formalize four properties of SHIELD that underpin its operational reliability: trust estimates converge to true peer quality, safety overrides preserve invariant sets, intrusion detection provides quantifiable coverage, and consensus tolerates quarantine-induced topology changes.

Assumptions. The communication graph $\mathcal{G}(t)$ is time-varying but connected over every interval of length L; all signals lie in compact sets; trust parameters satisfy $\kappa > 0$, $\beta > 0$, $\zeta \in (0,1)$; inter-agent message delays are bounded by D; and the consensus action distribution $\bar{p}_i$ is derived from at least $(2f+1)$ honest neighbors.

Theorem 1 (Trust Convergence)**.** *(Ensures that trust scores faithfully track long-run peer reliability.)* If agent j's messages are consistent with the robust aggregate at fixed probability $p_{ij} > 0.5$ and delays are bounded by D, then $\tau_{ij}(t) \to p_{ij}$ almost surely.

Proof sketch. Write $\tau_{ij}(t) = r_{ij}(t) \cdot d_{ij}(t)$ where $r_{ij}(t) = (C_{ij} + \kappa)/(T_{ij} + \kappa)$ and $d_{ij}(t) = \exp[-\beta(t - t_{ij}^{\text{last}})]$. Each consistency check $\mathbf{1}\{\|\mathbf{s}_j - \tilde{\mathbf{s}}_i\| < \delta\}$ is a Bernoulli trial with success probability p_{ij}. Since κ is a fixed constant, the ratio $r_{ij}(t) \to p_{ij}$ almost surely by the Strong Law of Large Numbers as $T_{ij} \to \infty$. Under bounded delays ($t - t_{ij}^{\text{last}} \le D$), the decay factor satisfies $d_{ij}(t) \ge e^{-\beta D} > 0$ and $d_{ij}(t) \to 1$ whenever communication persists. The product therefore converges: $\tau_{ij}(t) \to p_{ij} \cdot 1 = p_{ij}$. □

Theorem 2 (Safety Invariance)**.** *(Ensures that the override mechanism cannot drive the system out of a safe operating region.)* Let $V : \mathcal{S} \to \mathbb{R}_{\ge 0}$ be a control Lyapunov function such that for every state s there exists an action a^* with $\dot{V}(s, a^*) \le 0$, and suppose $\arg\max_a \bar{p}_{i,a}(t)$ selects such an action. Then the override rule

$$a_i(t) = \begin{cases} \arg\max_a \bar{p}_{i,a}(t), & \text{if } D_{\text{KL}}(p_i \| \bar{p}_i) > \varepsilon \text{ or } u_i > \tau_{\text{vac}}, \\ \text{sample from } p_i(t), & \text{otherwise,} \end{cases} \tag{26}$$

ensures the state remains in the safe set $\mathcal{S} = \{s \mid V(s) \le c\}$ for all $t \ge 0$.

Proof sketch. Case 1 (override active): The consensus action $\arg\max_a \bar{p}_{i,a}$ satisfies $\dot{V} \le 0$ by assumption, so V is non-increasing. *Case 2 (override inactive):* $D_{\text{KL}}(p_i \| \bar{p}_i) \le \varepsilon$ implies $\|p_i - \bar{p}_i\|_1 \le \sqrt{2\varepsilon}$ by Pinsker's inequality. Since $\bar{p}_i$ concentrates mass on Lyapunov-decreasing actions and $\dot{V}$ is continuous in the action distribution, the sampled action satisfies $\dot{V}(s, a_i) \le 0$ provided ε is chosen such that $\sqrt{2\varepsilon}$ lies within the continuity radius of the Lyapunov decrease condition. □

Theorem 3 (Detection Lower Bound)**.** *(Quantifies the minimum fraction of attacks the IRS is guaranteed to detect.)* Let u_{nom} denote the nominal vacuity under benign operation and suppose intrusions shift vacuity upward by $\Delta_u > 0$, so that the post-attack vacuity distribution has mean $u_{\text{nom}} + \Delta_u$. If τ_{vac} satisfies $u_{\text{nom}} < \tau_{\text{vac}} < u_{\text{nom}} + \Delta_u$, then the IRS detects at least a fraction $1 - e^{-\lambda T \beta_d}$ of attacks over a mission of duration T, where $\beta_d = 1 - F(\tau_{\text{vac}}; u_{\text{nom}} + \Delta_u)$ is the per-step detection probability and λ is the attack arrival rate.

Proof sketch. At each attacked timestep, the vacuity $u_i(t)$ is drawn from a distribution with mean $u_{\text{nom}} + \Delta_u$. The probability that $u_i(t) > \tau_{\text{vac}}$ is $\beta_d = 1 - F(\tau_{\text{vac}}; u_{\text{nom}} + \Delta_u) > 0$ since $\tau_{\text{vac}} < u_{\text{nom}} + \Delta_u$. For a Poisson attack process with rate λ, the expected number of attacked timesteps in $[0, T]$ is λT, and each is independently detected with probability β_d. The probability of missing *all* detection opportunities is at most $e^{-\lambda T \beta_d}$, yielding the stated bound. □

Theorem 4 (Consensus Resilience)**.** *(Ensures that quarantining compromised agents does not prevent the honest subswarm from reaching agreement.)* If at most f agents are quarantined and the remaining subgraph $\mathcal{G}'(t)$ is $(2f+1)$-connected, then the trust-weighted consensus protocol converges to the honest-agent average with error $O(\zeta)$.

Proof sketch. Quarantined agents have $\tau_{ij} < \zeta$ for all honest neighbors i, so their contribution to any trust-weighted average is bounded by $\zeta / (\sum_j \tau_{ij})$, which vanishes as $\zeta \to 0$. The remaining $(N - f)$ honest agents form a $(2f+1)$-connected subgraph. By classical results on consensus under symmetric, connected, doubly-stochastic weight matrices [37], iterative trust-weighted averaging over this subgraph converges to the uniform average of honest states. The residual influence of quarantined agents introduces a bias of $O(\zeta)$ per iteration, yielding the stated bound. □





USENIX VehicleSec '26 Artifact Appendix: SHIELD: Secure Human-Machine Interaction with Evidential Learning and Dynamic Trust for Drone Swarm Control

Dawood Wasif[1], Terrence J. Moore[2], Seunghyun Yoon[3], Hyuk Lim[3], Dan Dongseong Kim[4], Frederica F. Nelson[2], Jin-Hee Cho[1]

[1]Virginia Tech [2]U.S. Army Research Laboratory [3]Korea Institute of Energy Technology
[4]University of Queensland

A Artifact Appendix

A.1 Abstract

This artifact provides the complete implementation of SHIELD, a framework for secure human–swarm teaming in which evidential vacuity (from Dirichlet-based perception) jointly drives inter-agent trust modulation, intrusion detection, and human escalation within a decentralized control loop. It contains: (1) the SHIELD framework implementing Algorithm 1 (evidential perception, adaptive trust modeling, trust-weighted consensus, KL-divergence safety arbitration, and resilient agent lifecycle management); (2) an Intrusion Response System (IRS) with all six cyber-physical attack classes (A1–A6, Table 3); (3) four baselines (Baseline DRL/APPO, COACH, TAMER, SAHO); and (4) training, evaluation, and plotting scripts that reproduce all figures and tables. The artifact supports the paper's claims C1–C4 (see Major Claims below). This artifact was awarded the *Artifacts Available* and *Artifacts Functional* badges.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

The artifact is defensive and poses no security or privacy risk to evaluators. All experiments run entirely in simulation using the open-source QuadSwarm environment [1]; no physical UAV deployments, real communication disruptions, or human subjects are involved. The six attack classes are implemented as parameterised perturbations inside the simulator and cannot affect real systems. No personal data is collected or processed.

A.2.2 How to access

The artifact is permanently archived on Zenodo with a stable DOI: **10.5281/zenodo.20208768** (https://doi.org/10.5281/zenodo.20208768).

A.2.3 Badges

This artifact was awarded *Artifacts Available* (publicly archived on Zenodo with a stable DOI) and *Artifacts Functional* (the artifact includes all components described in the paper—Algorithm 1, all six attack classes, all four baselines, all three experimental protocols—with documentation and end-to-end scripts that produce the reported results).

A.2.4 Hardware dependencies

An NVIDIA GPU with ≥8 GB VRAM (e.g., RTX 3080 or A100, CUDA 11.7+); ≥8 CPU cores (Sample Factory uses multiple workers); ≥32 GB RAM; and ≥20 GB disk for checkpoints and logs. Per-agent inference latency is 4.2 ms with 8 UAVs (Table 11), within the 20 ms control budget at 50 Hz.

A.2.5 Software dependencies

Ubuntu 22.04 and Python 3.11 (system packages `build-essential`, `python3.11`, `python3.11-venv`, `python3.11-dev`). All Python dependencies install via `pip install -e .`, the authoritative dependency path for the artifact.

A.2.6 Benchmarks

All scenarios are generated procedurally by the QuadSwarm simulator [1] (50 Hz; σ_{GPS}=0.3 m, σ_{IMU}=0.01 rad/s; 20×20×10 m arena); no external datasets are required. The six attack classes (A1–A6) are parameterised per Table 3 and injected by the `AttackInjector` module.

A.3 Set-up

A.3.1 Installation

```
# fetch the code from the Zenodo DOI, then:
cd SHIELD
```

```
python3.11 -m venv shield_env
source shield_env/bin/activate
pip install -U pip setuptools wheel
pip install -e .
# verify component registration:
python -c "from swarm_rl.train import \
 register_swarm_components as reg; \
 reg(); print('OK')"
```

A.3.2 Basic Test

A short training run confirms the environment and SHIELD wrapper function:

```
python -m swarm_rl.train \
 --env=quadrotor_multi \
 --hmt_approach=shield \
 --quads_num_agents=8 \
 --quads_neighbor_visible_num=0 \
 --train_for_env_steps=100000 \
 --experiment=sanity_check
```

Expected: training starts, prints the SHIELD parameter summary (Table 16: τ_{vac}=0.4, κ=0.1, β=0.05, ζ=0.2, ε=0.3, δ=0.15), and writes TensorBoard logs to `./train_dir/`.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): Nominal performance. SHIELD attains higher Survival Rate (∼82–86%) and Mission Success Rate (∼3.5–4.0%), with comparable or better Crash Penalty, versus Baseline DRL, COACH, SAHO, and TAMER in open- and dense-field settings. Proven by (E1); see Section 6.1 (Figure 2) and Tables 4–6.

(C2): Intrusion detection. SHIELD +IRS achieves 82.5% Detection Rate, 14.6% False Positive Rate, 9.3 s Recovery Time, and 13.5 Human Interventions/episode, outperforming all partial variants. Proven by (E2); see Section 6.2, Table 2.

(C3): Operator workload. SHIELD +IRS reduces human confirmations by 53% versus the no-IRS variant (13.5 vs. 28.7/episode). Proven by (E2); see Section 6.2, Table 2.

(C4): Complementary modules. Removing any single module (vacuity escalation, trust updates, or safety override) causes monotonic degradation. Proven by (E1) and (E3); see Section 6.3, Table 4.

A.4.2 Experiments

(E1): *Nominal benchmarking* [∼20–40 compute-hours]: trains all five methods for 2.5×10^7 steps (Table 16 hyperparameters), open field via `bash train_local_hmt_comparison.sh` and dense field via `train_local_hmt_comparison_obstacles.sh`. Monitor with `tensorboard --logdir=./train_dir/`; test-time metrics (30 trials, final checkpoint) reproduce Figure 2 and Tables 4–6 within the reported standard deviations (validates C1, C4).

(E2): *Intrusion robustness* [∼20–40 compute-hours]: `bash train_irs_security_evaluation.sh` activates A1–A6 (Table 3) in the dense field with the IRS enabled, then `python -m swarm_rl.plot_irs_results` summarises DR/FPR/RT/HIC, reproducing Table 2 (e.g., DR 82.5±3.45%) (validates C2, C3).

(E3): *Ablation & scalability* [∼20–40 compute-hours]: disable modules with `--shield_tau_vac=100` (w/o vacuity), `--shield_zeta=0` (w/o trust), and `--shield_epsilon=100` (w/o safety override); vary `--quads_num_agents`$\in\{8,16,32\}$ with `--quads_use_obstacles=True`. Expect monotonic degradation (Table 4) and stable SR across N (Table 6) (validates C4).

A.5 Notes on Reusability

SHIELD is modular and extensible: new attack classes extend `AttackInjector.step()` (`irs_security_evaluation.py`); custom trust rules replace `ShieldNode.update_trust()` (`node.py`); alternative human interfaces implement the `HumanOracle` API (`human_interface.py`); other Gymnasium environments exposing per-agent positions can replace QuadSwarm; and all SHIELD parameters (Table 16) are command-line arguments sweepable via the shell scripts.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20260101. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/vehiclesec2026/.

References

[1] Z. Huang, S. Batra, T. Chen, R. Krupani, T. Kumar, A. Molchanov, A. Petrenko, J. A. Preiss, Z. Yang, and G. S. Sukhatme, "QuadSwarm: A modular multi-quadrotor simulator for deep reinforcement learning with direct thrust control," *arXiv preprint arXiv:2306.09537*, 2023.