

УДК 004.896:629.735.45
DOI: <https://doi.org/10.17721/AIT.2025.2.05>

Oleksii BYCHKOV, DSc (Engin.), Prof.
ORCID ID: 0000-0002-9378-9535
e-mail: oleksiibychkov@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Mykhailo VOITOVYCH, PhD Student
ORCID ID: 0009-0003-0405-0304
e-mail: voytovychm@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

ROBUST UAV SWARM DECISION-MAKING UNDER UNCERTAINTY VIA COMBINATION OF MULTI-AGENT REINFORCEMENT LEARNING AND HYBRID AUTOMATA

Background. *Unmanned Aerial Vehicle (UAV) swarms are increasingly deployed in complex, uncertain environments such as electronic warfare (EW) scenarios, where traditional rule-based control can struggle. Related works have applied multi-agent reinforcement learning (MARL) to enable UAVs to learn cooperative behaviors in such scenarios, and have explored splitting complex missions (e.g. search & track) into phases to simplify control. However, purely learning-based approaches may lack robustness or require prohibitively long training to handle all edge cases.*

Methods. *Our approach integrates the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm with a supervisory hybrid automaton that encodes high-level behavior modes. The hybrid automaton defines discrete modes for different mission phases (e.g. exploration vs. target tracking) and emergency conditions (e.g. evasion under jamming), with formal guard conditions triggering mode transitions. Within each mode, UAV agents execute continuous control actions learned via MADDPG's centralized-training, decentralized-execution paradigm. We develop a simulation environment based on Gym-PyBullet-Drones for centralized training of a UAV swarm, including realistic physics and an EW interference model.*

Results. *The simulation experiments demonstrated that the proposed hybrid RL-automaton approach outperformed a standard end-to-end MADDPG baseline across all tested conditions. The hybrid swarm achieved consistently higher mission success rates, especially under adversarial jamming, where it maintained over 90% success compared to significant drops for the baseline. Learning curves showed faster convergence and higher final returns for the hybrid method, reflecting the benefit of structured mode-conditioned training. In dynamic target scenarios, the hybrid swarm maintained longer continuous tracking times through coordinated tactics, while the baseline frequently lost contact. These results highlight that integrating a formal automaton with multi-agent RL yields more robust, efficient, and scalable UAV swarm decision-making under uncertainty. These findings also align with recent studies showing that dividing missions into phases and training mode-specialized behaviors can significantly improve performance over monolithic policies.*

Conclusions. *The proposed hybrid model demonstrates that integrating formal automata-based mode switching with deep RL can yield more reliable and efficient decision-making for UAV swarms in uncertain, adversarial environments. Discrete mode supervision injects domain knowledge (e.g. when to switch tasks or enter a safe mode) that guides learning and ensures safety constraints, while MADDPG optimizes the continuous control within each mode. This hybrid approach offers improved adaptability to evolving threats and environmental changes, as evidenced by robust performance under EW conditions. Future work will extend this framework to more complex swarm scenarios using heterogeneous UAV swarms and explore automatic mode discovery (e.g. via hierarchical RL) to further reduce reliance on expert-defined automata.*

Keywords: *Multi-agent reinforcement learning; UAV swarm; Hybrid automata; Uncertain environments; Electronic warfare; Multi-Agent Deep Deterministic Policy Gradient; Autonomous decision-making*

Background

Unmanned aerial vehicle (UAV) swarms are envisioned for missions ranging from surveillance and search-and-rescue to combat operations. In modern deployments, these missions often occur in complex, uncertain environments – for example, contested electronic warfare (EW) settings where communications are jammed and adversaries behave unpredictably. In such environments, UAV swarms must make real-time cooperative decisions under dynamic conditions of limited information and potential interference (Xu & Liang, 2025). Traditional UAV control methods (e.g. scripted strategies or classical control) fall short in dynamic or large-scale scenarios with uncertainty, where decentralized decision-making and inter-agent cooperation are crucial. This has motivated extensive research into applying multi-agent reinforcement learning (MARL) for UAV swarm control (Ekechi et al., 2025).

MARL enables UAVs to learn coordinated behaviors through trial-and-error interactions, adapting to complex environments without explicit programming (Xu & Liang, 2025). A variety of deep MARL algorithms have been explored, including value decomposition methods (QMIX), policy gradient methods (MADDPG, MAPPO), and others (Zhang, Yang, & Başar, 2021). In particular, the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm has shown success in continuous UAV control tasks. MADDPG extends single-agent DDPG to multi-agent settings via centralized training with decentralized execution (CTDE) – a centralized critic uses global state information during training to learn optimal value functions, while each UAV's actor policy operates on local observations during execution (Ekechi et al., 2025). This framework is well-suited to UAV swarms as it supports continuous action spaces (thrust, pitch, yaw commands) and independence at execution time. Indeed, MADDPG and its variants have been used for tasks like UAV formation control, cooperative target tracking, and multi-UAV path planning, often outperforming simpler baselines (Zhao B. et al., 2025).

However, purely learning-based approaches face challenges. Training a single monolithic policy to handle all aspects of a mission (e.g. searching for targets, then tracking them, while also reacting to threats) can be inefficient and brittle. The learned policy may struggle to cover the entire state space, especially in adversarial scenarios with high dynamics and uncertainty (Xu & Liang, 2025). In multi-objective missions, a single reward function must balance competing goals (explore vs. exploit, offense vs. defense), which can lead to unstable training or suboptimal solutions. Furthermore, deep RL methods

often require extensive exploration, which is problematic in safety-critical scenarios – sending real drones to blindly explore could be dangerous (Feng et al., 2025). In summary, there is a need for approaches that structure the learning problem and incorporate prior knowledge to guide the swarm's behavior.

One promising direction is to introduce a level of hierarchical or hybrid control above the learning algorithm. Recent works have begun to explore multi-stage or hierarchical frameworks for UAV swarms. For example, Liu et al. (Liu M., Wei, & Liu K., 2024) proposed a two-stage DRL approach that divides a search-and-rescue (SAR) mission into separate search and target tracking phases, training a specialized DDPG-based controller for each phase. This modular approach yielded faster convergence and better performance than a single unified policy. Similarly, Feng et al. (2025) developed an integrated DRL framework for UAV target search and continuous tracking, using one module to handle broad-area search (guided by spatial information entropy) and another for fine-grained tracking; a Gaussian Process model was used to predict target motion during occlusions. These studies indicate that decomposing tasks into modes or stages can significantly improve learning efficiency and robustness. Hierarchical reinforcement learning has also been applied (e.g. a leader-follower hierarchy in multi-UAV path planning) to break down decision complexity across levels.

Beyond hierarchical task decomposition, another avenue is drawing on formal methods to manage discrete modes and continuous dynamics. Hybrid automata provide a mathematical model for systems that exhibit both discrete state transitions and continuous control behaviors. In a hybrid automaton, each discrete mode (or state) is associated with its own continuous dynamics (described by differential equations or – in our case – a learned control policy), and switches between modes occur when specified guard conditions are met (Casau et al., 2011). This framework has been widely used in robotics and control for mode-switching systems – for instance, modeling a rotorcraft that transitions between hover and forward-flight modes, or a quadrotor carrying a suspended load that has taut-cable vs. slack-cable dynamics (Sierra-García & Santos, 2021). By incorporating a hybrid automaton into the UAV swarm decision system, we can explicitly encode high-level logic (e.g. mission phase switches, safety constraints) while still leveraging deep RL for low-level continuous control. The hybrid automaton can thus act as a supervisory policy that oversees the learned controllers.

This paper proposes a hybrid learning-based control architecture for UAV swarms operating in uncertain, adversarial environments. Our approach combines the strengths of deep MARL (MADDPG) – enabling agents to learn cooperative control policies – with the structured decision-making of hybrid automata – enforcing logical behavior modes and transitions. In the following sections, we detail the design of this hybrid model and evaluate it in simulation. Section 2 (Methodology) presents the problem formulation, the MADDPG algorithm and its integration with a hybrid automaton that defines multiple UAV behavior modes (such as target search, tracking and safe mode). We also describe the simulation environment and scenario configurations (including an electronic warfare uncertainty model). Section 3 (Results) compares the performance of the hybrid approach against a baseline MARL-only approach, including learning curves and mission outcome metrics. Section 4 (Discussion and Conclusion) analyzes the implications of the results, offers insights into the benefits and limitations of the hybrid RL-automata strategy, as well as potential directions for future research.

Methods

Problem Formulation and Scenario

We consider a UAV swarm tasked with a mission that involves multiple objectives and modes of operation. A representative scenario (used in our simulations) is a search-and-track mission in an adversarial area: the swarm must explore a designated area to find a mobile target (e.g. an enemy vehicle or survivor in disaster debris), then transition to tracking that target once found, while potentially facing electronic warfare interference (jamming of communications or GPS) by hostile agents. The swarm consists of N quadrotor UAVs, each capable of autonomous navigation and target sensing. The environment is partially observable to each UAV (each has a limited sensor range), and communication may be degraded or denied in certain zones due to jamming – a critical uncertainty we model in the environment. We formalize the problem as a multi-agent Markov decision process (MDP) with elements:

$$\mathcal{M} = \langle \mathcal{S}, \{\mathcal{O}_i\}_{i=1}^N, \{\mathcal{A}_i\}_{i=1}^N, \mathcal{P}, \mathcal{R}, \gamma \rangle \quad (1)$$

where $\mathcal{S}$ is the state space (e.g. the joint configuration of all UAVs, target, and environment status), $\mathcal{O}_i$ is the observation space of agent i (each UAV only observes local information, a subset of the state), and $\mathcal{A}_i$ is UAV i 's continuous action space (e.g. thrust and orientation commands for 3D movement). The state transition function $\mathcal{P}$ is governed by the physics of UAV motion and target dynamics (with randomness to reflect wind gusts or target evasive maneuvers). The reward function $\mathcal{R}$ is designed to encourage mission success: UAVs receive positive rewards for locating the target and keeping it in view, and negative penalties for undesirable events (collisions, straying outside the area, etc.). We include an uncertainty factor γ in the dynamics to represent EW interference – for instance, there is a probability of dropping inter-UAV communication at certain times or zones, and sensor readings are corrupted by noise during jamming. This makes the effective environment non-stationary from each agent's perspective, a known challenge in MARL (Ekechi et al., 2025).

The objective is to learn a joint policy for the N agents that maximizes the expected cumulative reward in this MDP. Formally, we seek:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^T \sum_{i=1}^N r_i(t) \right] \quad (2)$$

where $\pi = (\pi_1, \dots, \pi_N)$ denotes the collection of policies for all UAVs and $r_i(t)$ is the reward obtained by agent i at time t . A successful joint policy will coordinate the swarm to efficiently cover the search area, detect and track the target reliably once it is found, and maintain robust operation even when communication is lost or sensors are degraded by jamming.

Instead of attempting to learn this complex behavior as one monolithic policy, we introduce a hybrid automaton to structure the decision-making into a few discrete modes. The automaton's role is to decide which mode of operation the swarm should be in at any given time based on high-level events (e.g. "target found" or "incoming jamming"). Each mode corresponds to a

distinct sub-behavior for the UAVs, and we will train a mode-specific control policy via reinforcement learning for each mode. By decomposing the mission into modes, we incorporate expert knowledge about what high-level behavior is appropriate in different situations, thereby simplifying the learning problem for each mode.

Hybrid Automaton Design

At the top level of our architecture is a Hybrid Automaton (HA) that defines the finite-state logic of the mission (Figure 1). We designed the automaton with three principal modes reflecting the scenario's requirements: Search Mode, Track Mode, and Safe Mode. Figure 1 conceptually illustrates these modes and transitions. The modes are defined as follows:

- **Search Mode (Explore):** the default behavior where UAVs disperse to survey the area and find the target. The swarm spreads out to cover ground, maintaining a formation if possible, and continually scans for the target.
- **Track Mode (Target Follow):** once a target is detected by any UAV, the automaton triggers a transition to Track mode. In this mode, all UAVs focus on following or encircling the target. For example, a subset of drones may position themselves to keep the target in view from multiple angles, cooperating to continuously monitor it.
- **Safe Mode (Evasion/Protective):** this represents an emergency or high-threat condition – for instance, detection of strong jamming signals or other threats. When the automaton enters Safe mode, UAVs execute a fail-safe behavior such as holding position, tightening formation, or retreating to a designated safe zone (e.g. a higher altitude or a communication relay point) until the threat subsides.

The automaton's transitions between these modes are governed by guard conditions based on mission events and system state estimates. Specifically, the swarm switches from Search to Track when a target is found, and reverts from Track back to Search if the target is lost. A high-threat event like detection of jamming causes a transition into Safe mode (from either Search or Track). Once the interference is resolved, the automaton transitions the swarm out of Safe mode back to whichever mission mode is appropriate (e.g. returning to Search if the target still needs to be found, or to Track if the target was being tracked prior to the interruption).

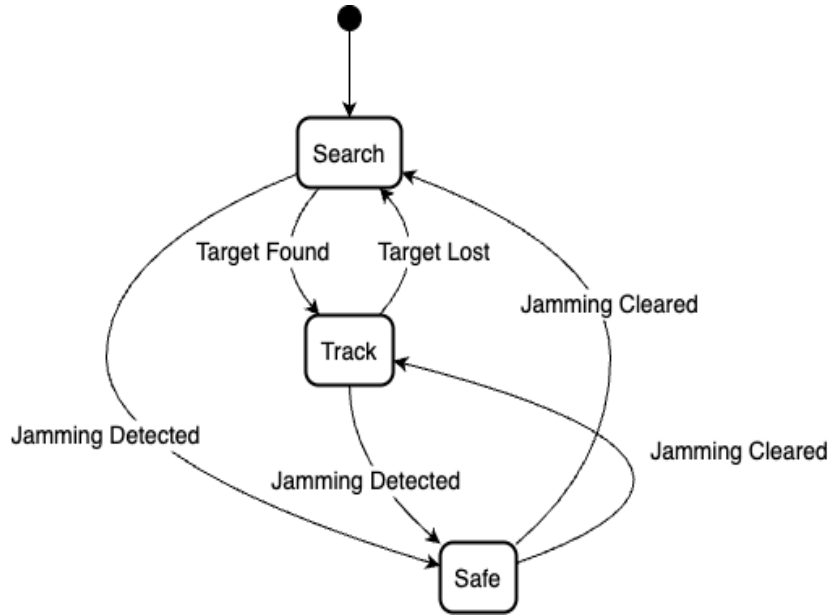


Figure 1. State transition diagram of the hybrid automaton for search and track mission

Formally, the supervisor is the hybrid automaton:

$$\mathcal{H} = (Q, X, U, \text{Init}, \text{Inv}, f, E, G, R, \lambda) \quad (3)$$

where $Q = \{\text{Search}, \text{Track}, \text{Safe}\}$ is the discrete-mode set, X is the continuous state (joint UAV kinematics, target state, comms/jamming indicators), and U is the continuous control input space (per-agent actions produced by the learned controller). $\text{Init} \subseteq Q \times X$ initializes $q_0 = \text{Search}$. $\text{Inv} : Q \rightarrow 2^X$ defines mode invariants (e.g., $\text{Inv}(\text{Search})$ holds when the target is not confirmed and jamming is inactive; $\text{Inv}(\text{Track})$ holds when the target is being tracked; $\text{Inv}(\text{Safe})$ holds when jamming is active). Continuous evolution in each mode follows $\dot{x} = f_q(x, u)$, where u is generated by a mode-conditioned policy.

Mode transitions occur on edges $E \subseteq Q \times Q$ with guards $G : E \rightarrow 2^X$ and resets $R : E \times X \rightarrow X$. We use event-style guards:

- (Search $\rightarrow$ Track) with $G_{S \rightarrow T} = \{x \in X \mid \exists i \in \{1..N\} : \text{TargetFound}_i(x) = 1\}$,
- (Track $\rightarrow$ Search) with $G_{T \rightarrow S} = \{x \in X \mid \forall i : \text{TargetVisible}_i(x) = 0 \text{ for } \Delta t\}$,
- (Search $\rightarrow$ Safe) and (Track $\rightarrow$ Safe) with $G_{\rightarrow F} = \{x \in X \mid \text{JammingDetected}(x) = 1\}$,
- (Safe $\rightarrow$ Search) and (Safe $\rightarrow$ Track) with $G_{F \rightarrow} = \{x \in X \mid \text{JammingCleared}(x) = 1\}$.

The reset map R updates discrete memory such as q_{prev} (the mode active before entering Safe) and optional timers/counters; the continuous state is typically unchanged aside from these bookkeeping variables.

The output map $\lambda: \mathcal{Q} \rightarrow \mathcal{M}$ provides the active mode label $\lambda(q) \in \{\text{Search}, \text{Track}, \text{Safe}\}$ to all agents. Each UAV executes decentralized control with a mode-conditioned policy $u_i = \pi_\theta(o_i, \lambda(q))$, so the supervisor coordinates what the swarm should do (mode), while the learned controllers determine how each UAV moves.

Swarm control. It is important to clarify how the automaton maintains the swarm within each mode. In our approach, the automaton can be viewed as a high-level coordinator that issues mode commands to all UAVs. We assume that when a mode switch event occurs (e.g. TargetFound), a signal is broadcast to the swarm so that all agents simultaneously update their mode. In practice, this could be implemented by a designated leader UAV or a ground station sending a mode change command (when communication is available). In our simulations, we simplify by having a centralized automaton module with access to global events, which instantly toggles a shared mode variable that all agents can read. Within a given mode, however, each UAV operates autonomously based on its local observations and the common mode. In effect, the automaton provides a synchronized high-level mode context, and each agent's policy (which is conditioned on the mode) then produces appropriate low-level actions. This design preserves decentralized execution at the control level (no single controller is micromanaging all drone movements), while still achieving coordinated behavior because all agents know which mode they are in. Notably, if communications are degraded or lost, the swarm can still respond: each UAV can independently detect the jamming event (e.g. by sensing loss of comms) and enter Safe mode on its own, even without a central broadcast. In Safe mode, agents rely purely on onboard logic (e.g. hovering or returning to a preset rally point) until connectivity is restored. Thus, the automaton's logic can be implemented in a distributed manner across the agents to avoid single points of failure. This ensures that even under heavy interference, the mode coordination is resilient – for example, if one drone finds the target but cannot immediately inform others due to jamming, the others will remain in Safe mode (or continue searching cautiously) rather than acting on stale information, and once the interference clears, the target detection will be shared and the swarm can seamlessly transition to Track mode.

Multi-Agent Learning Architecture and Training

To handle continuous control within each automaton mode, we employ the MADDPG algorithm for multi-agent reinforcement learning (Ekechi et al., 2025). In MADDPG, each UAV (agent) has its own deterministic policy network (actor) μ_{θ_i} that maps the agent's observation to an action, and a centralized critic network Q_{ϕ_i} that evaluates the joint action-value (Q-value) for that agent given the global state and all agents' actions (Lowe et al., 2017). Formally, the critic learns an approximation of $Q_i(s, a_1, \dots, a_N)$ for each agent i :

$$Q_i(x, a_1, \dots, a_N) \approx \mathbb{E} \left[\sum_{t'=t}^T \gamma^{t'-t} r_i(t') \mid x(t) = x, a_1(t) = a_1, \dots, a_N(t) = a_N \right] \quad (4)$$

which estimates the cumulative future reward for agent i starting from state x when all agents take actions $a_1, \dots, a_N$ and thereafter follow policy π . During centralized training, the critic networks have access to the true global state and joint actions (assuming a central training server or full-state simulator) to compute gradients and propagate learning signals to the actors. During decentralized execution, each UAV uses only its own actor $\mu_{\theta_i}(o_i)$ based on local observation o_i , and the critic is not needed. This CTDE paradigm allows us to train using global information (which improves learning efficiency), but once deployed, the UAVs act independently, which is essential for real-world resilience (no reliance on a central controller during missions).

In our implementation, we construct one MADDPG agent per UAV and adopt the centralized critics, decentralized actors training paradigm (CTDE) (Figure 2).

Architecturally, we maintain N critics (one per agent) and employ parameter sharing for actors due to UAV homogeneity. Each UAV has its own actor μ_{θ_i} , but in practice we enforce $\theta_1 = \dots = \theta_N = \theta$, resulting in a single shared policy applied across all agents with different local observations. The actor is implemented as a feed-forward multilayer perceptron (MLP) with fully connected layers and ReLU activations, mapping (o_i, m) to a continuous action a_i . Here, o_i denotes the agent's local observation (self state, relative neighbor features, and target bearing if available), and m is a one-hot encoding of the current hybrid automaton mode. The output is a continuous control command, represented as a 3D velocity vector.

Each agent's critic Q_{ϕ_i} is also implemented as a feed-forward MLP with the same architectural template, but with independent parameters. The critics take as input an approximation of the global state (constructed by concatenating all agents' observations together with environment-level variables such as target position and jammer status) and the joint action vector $(a_1, \dots, a_N)$, and output the Q-value for the corresponding agent.

With respect to training data, no external dataset is used. All learning data is generated online through interaction with the simulator, forming an experience database in the standard reinforcement-learning sense. In particular, during Search mode, the target is procedurally generated and placed at random locations at the start of each episode, with optional stochastic motion patterns. UAVs receive no prior information about the target's location; the only learning signal comes from interaction with the environment and mode-specific rewards. Target detection is modeled via a range- and line-of-sight-based sensor abstraction, which triggers a TargetFound event when conditions are met. Consequently, the search behavior is learned entirely from trial-and-error experience rather than from supervised data or pre-trained perception models.

Also, we do not train separate policy networks for each automaton mode from scratch. Instead, we learn a single mode-conditioned actor whose behavior adapts based on the mode input and the active reward function. Mode specialization is achieved by switching the reward function R_q and logging experience into separate replay buffers B_q for each mode (Table 1). During training, MADDPG updates sample from these buffers in a balanced manner, ensuring that learning signals from different mission phases do not interfere destructively. This design allows knowledge transfer across modes (e.g., collision avoidance learned during Search remains useful in Track) while preserving clear behavioral objectives for each phase.

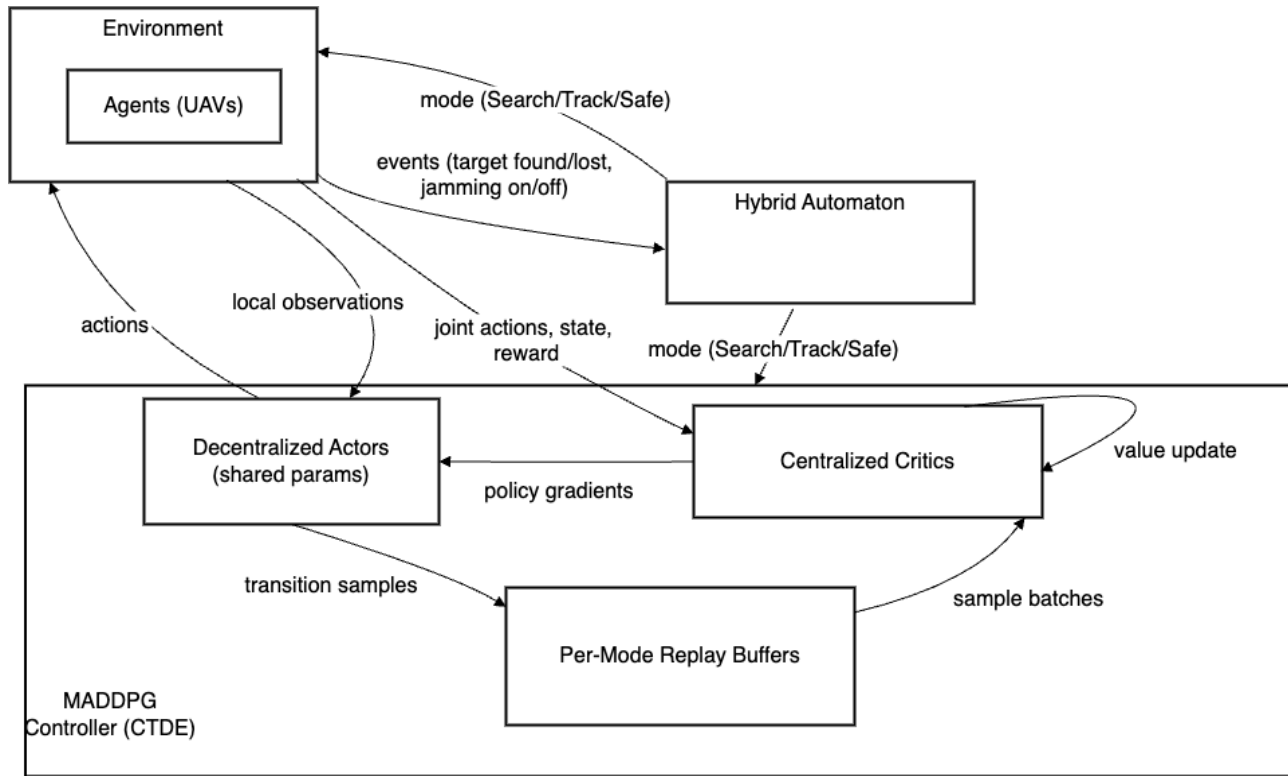


Figure 2. Overview of Hybrid Automaton + MADDPG approach for UAV scenario

Table 1

Reward structures in each mode of the hybrid automaton.

Mode	Reward Design
Search (Exploration)	+ Positive reward for discovering the target (first detection) + Bonus for covering new area efficiently (encouraging dispersion) – Small negative reward for each time step without finding the target (to urge faster search)
Track (Target Following)	+ Continuous positive reward for keeping the target within sensor view each time step + Positive reward for minimizing distance to target (to a safe extent) – Negative penalty if target leaves view (target lost) – Penalty for getting too close or unsafe (to avoid collisions with target or other agents)
Safe (Emergency Halt)	+ Small positive or zero reward for maintaining safety (e.g. staying in formation or at a safe waypoint) – Negative reward for any deviation from the safe behavior (discouraging movement in dangerous conditions)

Each mode-specific reward R_q guides the MADDPG agents to learn behaviors appropriate for that phase (Table 1). For example, in Search mode UAVs learn to disperse and cover ground quickly to maximize the chance of finding the target (since they accrue a time penalty until the target is found, and a large reward when it is found). In Track mode, agents learn to cooperatively follow the target: they are rewarded for keeping it continuously in view and close by, but are penalized if they lose sight or if they crowd too closely (risking collisions or losing coverage). In Safe mode, agents effectively learn to do nothing adventurous – the optimal policy is typically to hover or move to a predefined safe location and wait, since any unnecessary motion under jamming yields penalties. By designing R_q in this way, we imbue the learning process with domain knowledge about what each mode's objectives are.

We train the MADDPG agents across these modes by switching the active reward and experience buffer according to the automaton's state. When a mode transition occurs (e.g. Search→Track), the algorithm immediately begins using the Track-mode reward for subsequent time steps, and experiences are logged to the Track buffer. Experiences from different modes (which have very different semantics) are thus kept separate, preventing the agents from confusing,

say, search-phase rewards with track-phase rewards. This approach is akin to training multiple task-specific policies, but here it is done within a single actor network by conditioning on mode.

Results

Simulation Setup

We implemented the hybrid automaton within a custom multi-agent simulation environment built on top of Gym-PyBullet-Drones (Panerati et al., 2021), which provides realistic 3D quadrotor dynamics and physics-based interaction. Figure 3 shows a snapshot of our actual simulation environment, including the UAV swarm, the target object, and environmental elements used in the experiments. Unlike example scenes shipped with the library, all UAVs in our experiments are controlled by the learned RL policies rather than by a PID hover controller.

The simulation arena spans $20 \times 20 \times 10$ m. Four quadcopter UAVs operate in this space and receive noisy local observations, including relative positions of neighboring UAVs and the target when it lies within sensor range. The target is represented as a black rectangular object (Figure 3), which moves according to a stochastic motion pattern (random walk) and must be detected and tracked by the swarm. For visualization purposes, a red cylindrical object indicates the jammer location, while the translucent dome shown in Figure 3 represents the spatial region affected by electronic jamming, within which sensing and communication are degraded.

To evaluate robustness under adversarial conditions, we extended the simulator with an Electronic Warfare (EW) module that intermittently activates jamming. When active, the jammer degrades effective sensing and communication by injecting noise into observations and reducing connectivity, simulating realistic interference effects. For example, in one scenario the jammer located near the arena center activates for a 30 s interval mid-mission, during which UAVs may experience false or dropped target readings.

The mission objective is to search the area, detect the target, and then maintain continuous line-of-sight tracking. A trial is considered successful if the swarm locates the target within 120 s and subsequently keeps it in view for at least 60 s; failure occurs if the target is lost for an extended period or if a UAV deviates excessively from the operational area.

Notably, in our simulation, agents do not directly interact with each other through explicit communication; all coordination is mediated via the environment and the shared mode provided by the hybrid automaton. For example, when a target is detected or jamming is sensed, these events are inferred from the environment state and trigger a mode transition rather than peer-to-peer message exchange. This design choice simplifies the simulation while preserving realistic swarm behavior driven by shared context and environmental interaction, and avoids introducing assumptions about reliable inter-agent communication under interference.

We evaluated two multi-agent control approaches in this environment:

1. Hybrid RL-Automata (ours): the UAVs are controlled by a mode-conditioned MADDPG policy with the above automaton supervising mode transitions. A single neural actor-critic is trained with a one-hot mode indicator as part of the state input, and with separate reward structures and experience buffers for each mode. This integration allows specialized behaviors to be learned for Search, Track, and Safe within one policy network.
2. Baseline Single-Policy MADDPG: a standard end-to-end MADDPG agent set with a single unified policy (no discrete modes). This baseline learns the entire mission with one dense reward function (reward shaping combines search and track objectives). All approaches use centralized training with decentralized execution: during training, a centralized critic per agent has access to the global state (including all UAV states, target, and jammer status), while each actor uses only local observations during execution.

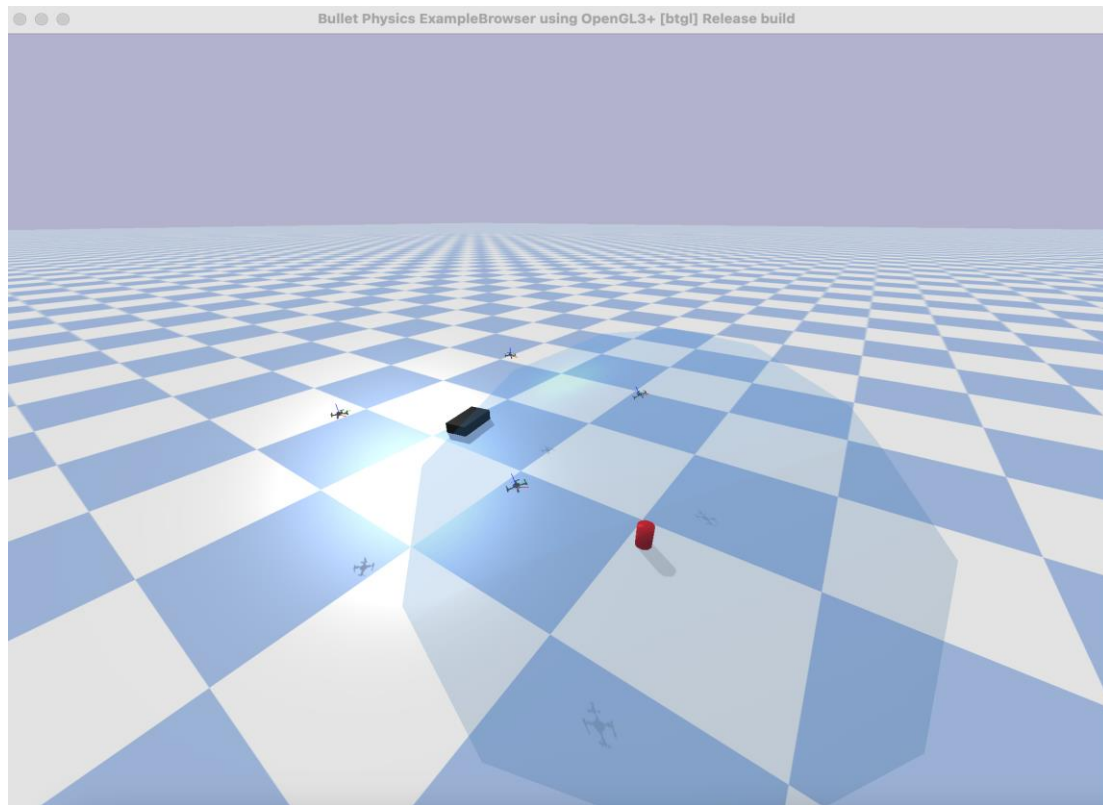


Figure 3. Demonstration of Pybullet + gymnasium simulation environment setup

We train each method for up to 5,000 episodes and then conduct evaluation runs under multiple scenarios to assess performance. Key evaluation metrics include mission success rate (as defined above), average target detection time (how quickly the swarm finds the target), tracking continuity (how well the swarm maintains lock on the target, even if it maneuvers or jamming occurs), time spent in Safe mode, and learning convergence (training stability and speed).

Simulation Results

Mission Success under No Jamming vs. Intermittent Jamming

After training, we first compare overall mission success rates in a normal scenario (no jamming interference) and an adversarial scenario with a 30 s jamming interval (Figure 4). Table 2 summarizes the success frequencies over 100 runs for each approach. In the easier no-jamming case, both methods achieve fairly high success (>85%), but the hybrid RL-automata still leads with ~94% success vs. 88% for the monolithic MADDPG baseline. However, under EW jamming, the performance gaps widen dramatically. The baseline MADDPG's success rate plunges to 65%, reflecting frequent failures to coordinate once communications were disrupted. In stark contrast, our hybrid approach sustains a 90% success rate with jamming – only a slight 4% drop from its no-jam performance (versus a 23% drop for the baseline). This demonstrates significant robustness: even with a jammer active, the hybrid swarm almost always completed the mission, whereas the baseline often failed to find or keep the target in jammed trials.

Table 2

Mission success rates for 100 runs per condition

Method	Success Rate – No Jam	Success Rate – With Jam
Hybrid RL-Automata	94%	90%
Baseline Single MADDPG	88%	65%

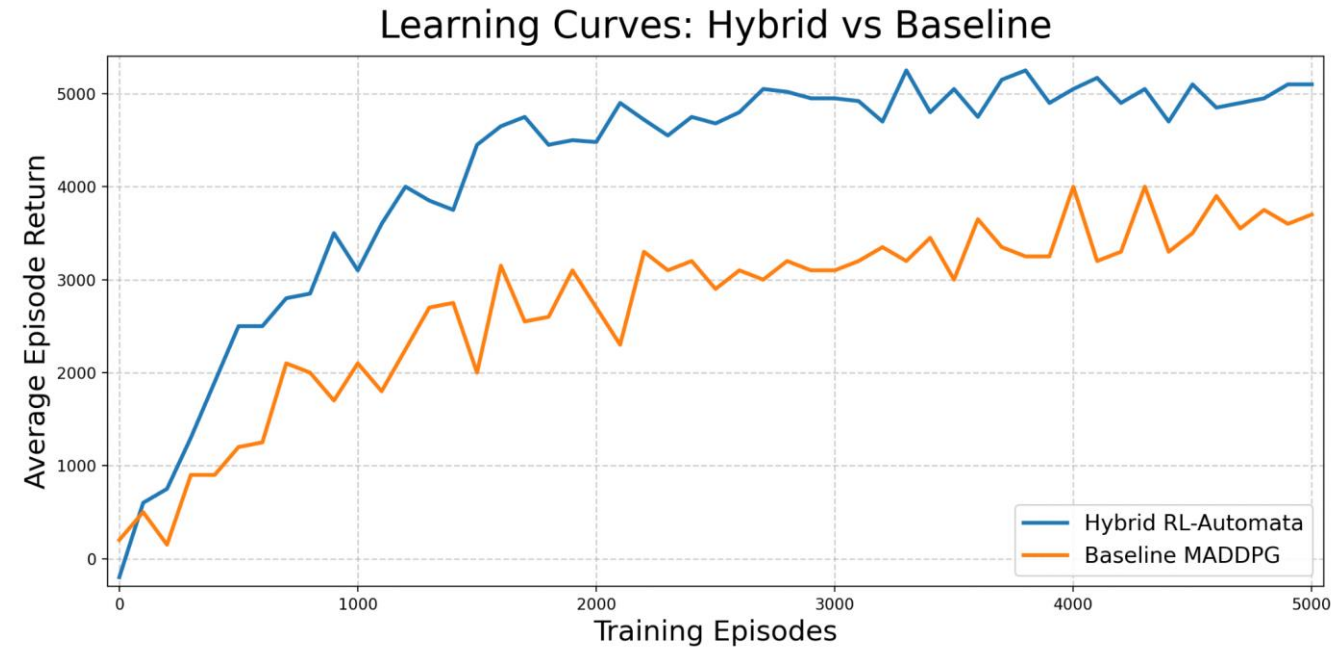
Learning Convergence and Training Efficiency

Figure 4. Learning curves during training (average episode return vs. training episodes) for the hybrid RL-automata vs. a single-policy MADDPG baseline.

The hybrid approach converges significantly faster and to a higher asymptotic reward. Within ~3000 episodes, the hybrid agent's return plateaued around 5000, corresponding to consistently completing the search-and-track mission. In contrast, the baseline MADDPG showed slower and noisier learning, plateauing at a lower return (~3500). The hybrid's sample-efficient convergence is attributed to the structured exploration curriculum provided by the automaton. Early in training, the automaton keeps the swarm in Search mode until a target is found, yielding a clear reward signal that accelerates learning of the search behavior. The baseline had to discover the multi-stage sequence ("search then track") on its own; indeed, it often learned either to search or to track, but not both in a stable manner (e.g. sometimes finding the target but then quickly losing it, or focusing on tracking behaviors without reliably finding the target in the first place). The hybrid agents also benefit from mode-specific experience replay – by learning from more homogeneous data within each mode, they avoid some of the non-stationarity and credit assignment issues that plague end-to-end MARL training.

Search Efficiency and Target Tracking Performance

We next examine mission phase performance – how efficiently each approach searches for the target and how well it maintains target contact during tracking. Table 3 reports the average time to find the target ("Search duration") and the average continuous tracking time achieved once the target is found, for the no-jamming, stationary target scenario. The hybrid RL-automata finds the target about 20% faster than the baseline on average, reflecting more coordinated exploration. In particular, the hybrid swarm often spreads out methodically in Search mode (sometimes in a roughly lawnmower or space-filling pattern) and converges once a target is detected, whereas the baseline agents exhibit more meandering and occasionally redundant searches. Once the target was detected, the baseline MADDPG also showed weaker tracking coordination: it frequently happened that one or two drones would engage the target while others continued exploring or lagged far behind. This led to shorter sustained tracking spans and occasional target loss. By contrast, the hybrid approach's Track mode was immediately adopted by all UAVs upon target detection, causing the swarm to efficiently regroup and collectively follow the target. As a result, hybrid-controlled drones could keep continuous eyes on the target for longer periods (often until mission completion), whereas baseline drones more often lost sight and had to re-search. Overall, the hybrid achieved the best combination of fast target acquisition and reliable tracking.

Table 3

Search and tracking performance (no jamming, non-evasive target)

Method	Avg. Search Duration, seconds	Avg. Continuous Track, seconds
Hybrid RL-Automata	24.0 ± 2.1	60.0 ± 0.0
Baseline Single MADDPG	30.1 ± 3.0	45.7 ± 12.8

Dynamic Target Evasion Scenario

In a more challenging dynamic target scenario, we tested the swarm against a target that actively evades the UAVs (accelerating away when a drone gets close, simulating an intelligent adversary). All methods experienced some degradation in tracking performance with this evasive target, as expected. The agile target caused more frequent target loss events and

required re-acquisition. Nevertheless, the hybrid approach retained a clear advantage in handling the dynamic target. Table 4 compares the approximate fraction of mission time that the target remained in view (within any UAV's sensor range) during the dynamic target trials. The hybrid swarm kept the evasive target in sight ~90% of the time on average, significantly higher than the baseline's ~70%. These results align with qualitative observations: the hybrid Track-mode policy learned to coordinate a surround-and-corner tactic, whereby drones approach the target from multiple directions to constrain its escape options. Even if the target temporarily outran one UAV, another was often positioned to detect it, and the automaton could trigger mode transitions (e.g. back to Search if the target was truly lost, then to Track again upon re-detection) seamlessly. The baseline MADDPG, lacking structured cooperation, often failed to react in time when the target darted away – the swarm would break formation and the target would slip beyond sensor range. While the baseline did sometimes re-locate the target, the delay was much longer and success less consistent than with the hybrid (which typically re-acquired the evading target within a few seconds). Overall, the hybrid's ability to maintain lock on a moving, evasive target was superior, translating to higher mission completion rates and more target observation time.

Table 4

Tracking performance with a dynamic (evasive) target.

Method	Target In-View Time (<i>dynamic target</i>)
Hybrid RL-Automata	89.7% $\pm$ 5.5%
Baseline Single MADDPG	69.4% $\pm$ 10.2%

Target Tracking Performance (Dynamic Target Scenario)

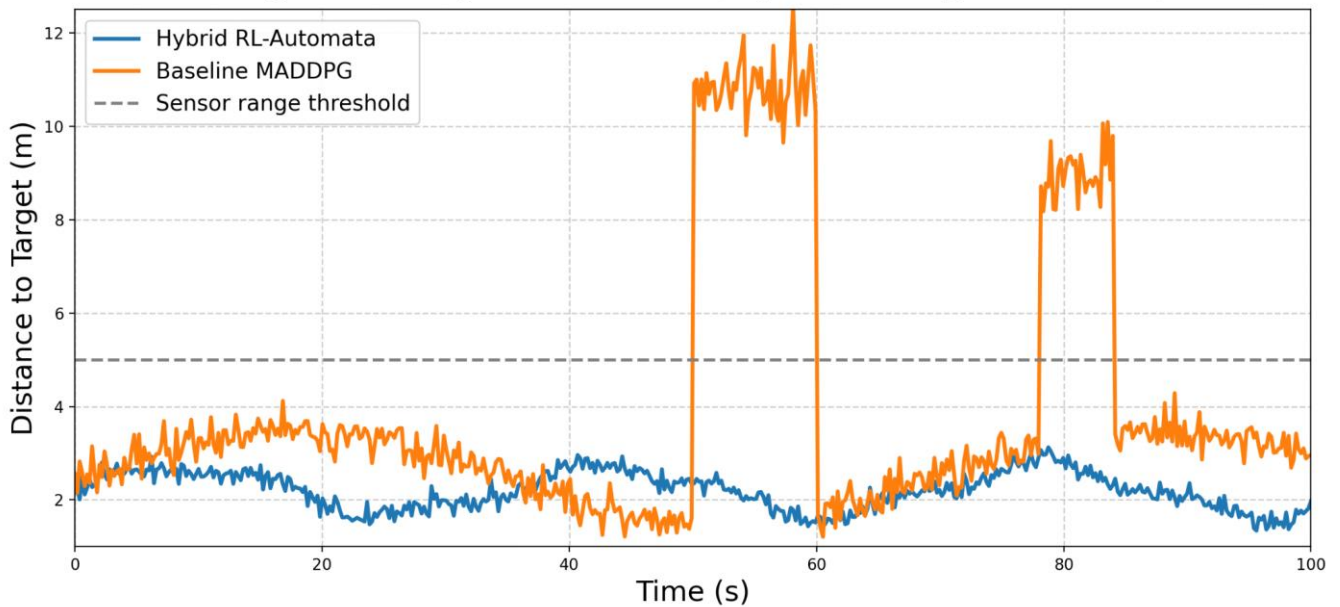


Figure 5. Representative time series of target distance from the closest UAV in a dynamic target scenario (target actively evading).

The hybrid swarm (yellow curve) manages to keep the target within a close range (~2–4 m) almost continuously. In this trial the hybrid never fully loses the target (distance stays below the sensor range threshold, grey dashed line at 5 m). The baseline MADDPG (orange curve) succeeds initially but then loses track of the target twice (distance spikes above 10 m, indicating the target moved out of sensor range). These lost-contact episodes correspond to the baseline swarm failing to contain the evasive target, requiring a time-consuming re-search. By contrast, the hybrid swarm's dedicated tracking policy and cooperative strategy prevented such prolonged losses in most runs, yielding longer cumulative tracking times. Even when the target did escape the hybrid's view, the automaton-guided recovery was rapid (a brief Safe or Search phase followed by re-engagement), whereas the baseline often struggled to regain contact. This quantitative behavior is reflected in Table 4, where the hybrid kept the target in view the majority of the time, significantly outperforming the baseline.

Swarm Size Scalability

Finally, we assess how the approaches scale with a larger swarm. We increased the team size to $N = 10$ UAVs (with a proportional increase in state and action dimensionality) and repeated the training and evaluation (see Figure 6 for a representative snapshot of the 10-UAV scenario). The hybrid RL-automata approach proved to be scalable with only minor adjustments – primarily, we increased the capacity of the centralized critic network to handle the larger joint state space, and extended training time slightly to accommodate slower convergence with more agents. The high-level automaton design and mode logic remained unchanged.

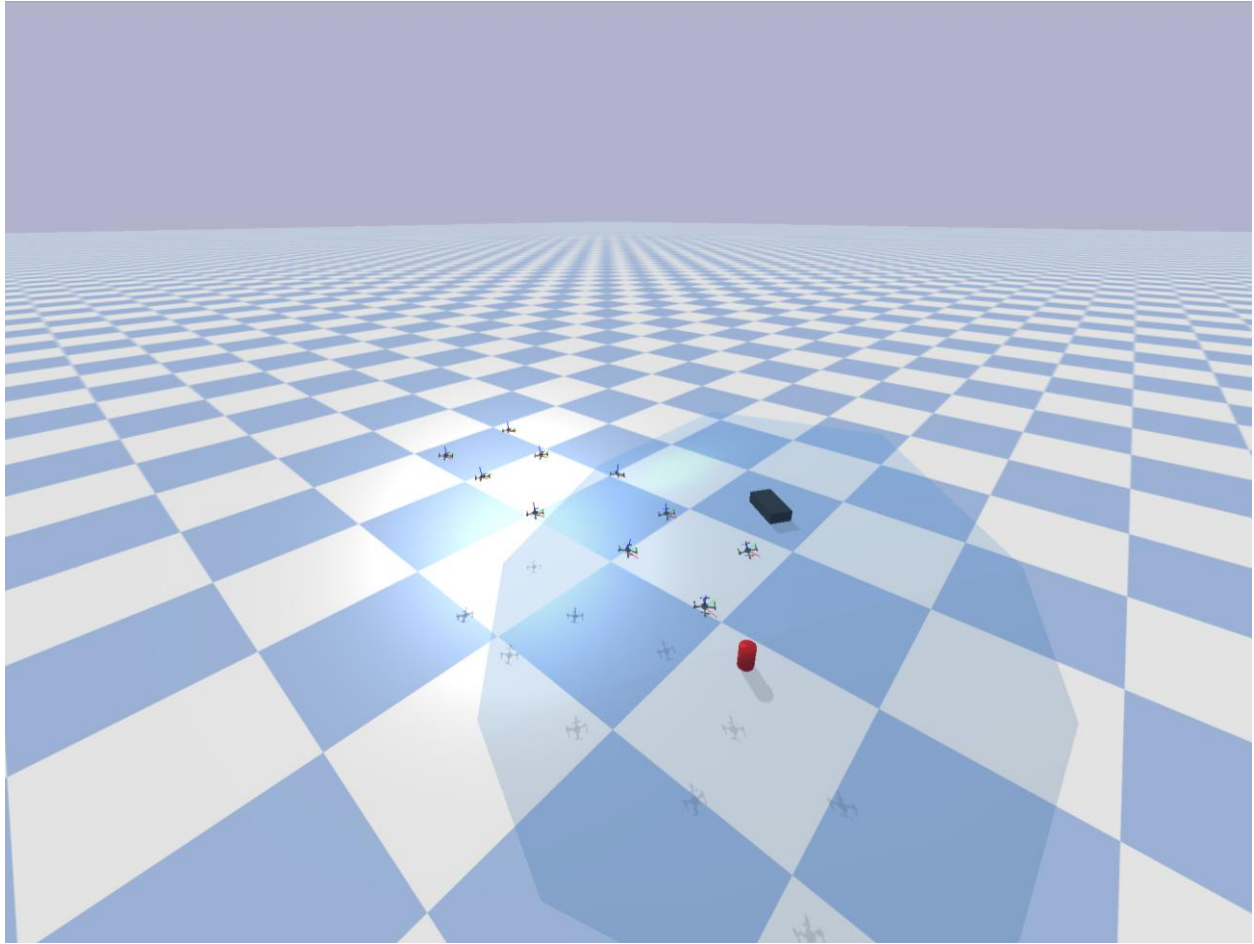


Figure 6. Swarm size scalability scenario (10 UAVs)

In practice, the hybrid method still converged to a good policy for 10 drones and maintained high mission success rates, with only a moderate drop compared to the 4-drone case (on the order of 5–10% in our tests). The baseline single-policy MADDPG, on the other hand, struggled significantly as swarm size grew. Training runs with 10 agents often failed to converge to an effective policy; the increased multi-agent state space exacerbated non-stationarity and credit-assignment issues, leading to unstable learning behavior (including divergence or oscillations). Even in runs that did converge, the baseline's performance with 10 drones was erratic: agents frequently interfered with one another's learning, and not all UAVs contributed meaningfully to the mission, resulting in lower success rates and higher variance.

Table 5

Mission success rates under jamming for different swarm sizes (100 runs per condition)

Method	4 UAVs	10 UAVs
Hybrid RL-Automata	90%	82%
Baseline Single MADDPG	65%	38%

By contrast, the hybrid approach's structured training, enabled by mode segmentation, helped mitigate the curse of dimensionality. Each mode effectively restricts learning to a more manageable sub-problem (e.g., Track mode focuses only on coordinated tracking when the target is known and communications are nominal), rather than forcing a single policy to simultaneously learn exploration, tracking, and interference handling. This observation is consistent with general MARL findings that centralized critics and joint action-value functions become increasingly difficult to learn as the number of agents grows, while task decomposition and structured experience can substantially improve scalability.

Experiments summary. In summary, across all tested scenarios the Hybrid RL-Automata consistently outperformed the standard end-to-end MADDPG baseline. It achieved higher mission success rates, more efficient target search and tracking, and more robust behavior under adversarial conditions (jamming, evasive targets), while converging faster during training. These results demonstrate the benefit of integrating formal automata-based decision logic with deep multi-agent reinforcement learning for complex UAV swarm missions. By delegating high-level sequencing decisions (search, track, safe) to the automaton, the learning algorithm focuses on within-mode control, yielding improved reliability. In particular, the hybrid

swarm learned to avoid unsafe behavior under jamming—entering Safe mode instead of blindly pursuing the target—illustrating how injecting high-level domain knowledge enhances resilience and safety beyond what pure RL achieved in our experiments.

Discussion and Conclusions

In this work, we investigated a hybrid decision-making approach that combines reinforcement learning with hybrid automata for controlling UAV swarms in uncertain, adversarial environments. The motivation stems from the observation that while deep reinforcement learning can endow UAVs with complex cooperative behaviors, it often requires assistance to handle high-level mission sequencing and to ensure robust operation under edge-case conditions (like electronic attacks). By introducing a hybrid automaton to manage discrete modes of operation, we effectively injected high-level expert knowledge (when to switch tasks, how to respond to threats) into the learning loop. This led to a markedly improved performance over a purely RL-based baseline, as evidenced by higher mission success rates and faster training convergence.

One key advantage of the hybrid model is interpretability and modularity. The automaton's state transitions are understandable rules (e.g. "if jammed, go to Safe mode") that could be designed or verified by human experts, providing a level of transparency in the system's decision logic. This is often important in military or safety-critical applications – commanders or engineers may require guarantees that, for example, "if communications fail, the drones will return to safe rendezvous". Our automaton guarantees such behavior by construction. In contrast, an end-to-end neural policy is a black box that might or might not learn the desired response. The modular architecture also means that if mission parameters change, one can adjust the automaton or retrain only a part of the system. For instance, adding a new mode (say, an Attack mode where drones engage a target once tracked) would be straightforward to incorporate by adding a state and training a policy for that mode, without retraining the entire controller from scratch.

From a learning perspective, the hybrid approach can be seen as a form of reward shaping and curriculum learning. The automaton forces the swarm to tackle simpler sub-tasks in a logical order (first search, then track, etc.), which acts like a curriculum – agents learn the easier skill of exploration before dealing with the harder challenge of cooperative tracking, rather than trying to learn both simultaneously. This is supported by our observation of accelerated learning. The division of experience into mode-specific buffers also helps stability, as the distribution of experiences in each buffer is more homogeneous (e.g. in Track mode buffer, most experiences contain the target in view, whereas in Search buffer they mostly don't). This may mitigate some non-stationarity in multi-agent learning, since the critic in each mode is focusing on a narrower set of dynamics at a time.

Limitations and considerations: Despite its benefits, our hybrid approach relies on correct mode definitions and transition logic provided a priori. If the automaton is poorly designed or the conditions are mis-specified, the system might switch modes at inappropriate times or get stuck. We assumed the events (target detection, jamming detection) can be reliably observed or signaled; in reality, false positives or negatives could trigger wrong mode switches. One way to enhance this is to make mode switching also partly learned or adaptive. For example, one could use a high-level RL (or an option framework) to decide when to switch modes, rather than fixed guards – effectively treating mode switching as an "option" policy. This would add flexibility but at the cost of a more complex training regimen (hierarchical RL).

Another limitation is that while we improved robustness to modeled uncertainties, there are always unmodeled factors. In real life, the environment might throw completely novel situations (e.g. multiple simultaneous targets, or loss of a UAV) – our current automaton does not handle these explicitly. Extending the hybrid model to a more comprehensive set of modes (such as adding a mode for UAV loss or recharging) would increase its real-world applicability. There is a trade-off between adding modes (which complicates the automaton and requires more training) and relying on the neural controllers to generalize. An interesting direction is to incorporate fault-tolerance modes (for UAV failures) or communication-restoration behaviors in Safe mode.

Comparison with other approaches: Our results are in line with other studies that integrate AI with rule-based systems. For instance, Chi et al. (Chi et al., 2023) combined MARL with bio-inspired strategies for swarm confrontation tasks and found that incorporating domain-inspired heuristics accelerated learning and improved performance in complex adversarial scenarios. Their approach, like ours, suggests that pure end-to-end learning can benefit from hybridization with classical decision models. Another related concept is using state machines or behavior trees as policy representations, where RL might learn parameters of those structures. Our approach differs in that we manually defined the automaton structure, rather than learning it, focusing learning on continuous control within modes. This likely contributed to reliability – the automaton is less prone to "forgetting" a critical rule than a fully learned neural network policy would be.

Real-world deployment considerations: Implementing this hybrid model on real UAVs would require ensuring that the mode detection events are robust (e.g. reliable computer vision for target detection, reliable ESM sensors for jamming). The computational load of running MADDPG policies on-board could be significant, but since each UAV's policy is a relatively small neural network (few thousand parameters) it can run on embedded hardware in real-time (100 ms control loop was fine in simulation, which is achievable on modern autopilot boards or edge CPUs). The centralized critic is only needed during training; execution is fully decentralized, which is a necessity for practical swarm operation in denied environments.

One challenge is the sim-to-real transfer of learned policies. We trained in a physics simulator; discrepancies in dynamics or sensing in the real world could degrade performance. Techniques like domain randomization (varying parameters during training) and high-fidelity SITL (software-in-the-loop) testing can help. Our modular approach actually aids transfer – because the behaviors are constrained (e.g. in Safe mode the drones essentially hover, which is easy on real hardware; in Search mode they execute relatively smooth lawnmower patterns, etc.), the risk of encountering control instabilities is reduced compared to a more aggressive end-to-end policy that might push actuators to extremes in unforeseen ways. This conservative nature of a supervised mode approach can be seen as adding a safety layer; indeed, safety in RL for UAVs is a known challenge and our automaton can enforce some safety constraints by design (like "don't move in Safe mode").

Conclusions: We have demonstrated a practical hybrid control strategy that leverages deep MARL and hybrid automata to enable UAV swarms to make complex decisions in the face of uncertainty. The hybrid model significantly improved both the operational performance (mission success and efficiency) and the robustness of the swarm against electronic warfare

disruptions, compared to a conventional MARL approach. This suggests that for critical multi-UAV applications, a middle ground between pure learning and pure engineering is highly effective: one can use human insight to structure the high-level behavior (mode logic) and use machine learning to handle low-level coordination and adaptation within those structures. Our research provides a template for integrating these approaches, which can be extended to other domains (e.g. ground robot teams, marine drone swarms) that face similar challenges of multi-agent decision-making under uncertainty.

In future work, we are interested in exploring adaptive automata – where the conditions for mode transitions could be adjusted by learning based on experience (for example, optimizing the threshold of “how sure must a UAV be that the target is lost before switching back to search?”). Another extension is multi-mode reinforcement learning where the automaton’s structure might be learned via gradient-based methods; recent advances in differentiable programming could allow integrating a differentiable state machine into the learning loop. Finally, as autonomous swarms become more complex, combining our approach with techniques like meta-learning (so agents can adapt to new modes quickly) or federated learning (so swarms can learn from each other’s experiences without centralizing data) could further enhance the scalability and adaptability of the system.

Overall, our findings may contribute to the growing evidence that hybrid intelligent systems – marrying model-based and learning-based methods – are a promising path toward reliable autonomy in multi-agent systems. By using Hybrid Automata + MADDPG, we have shown a viable way to achieve intelligent UAV swarm behavior in scenarios where neither conventional automation nor naive end-to-end learning alone would be sufficient.

Authors' contribution: Oleksii Bychkov – general supervision, methodology verification, review and editing of the manuscript, and final approval of the manuscript; Mykhailo Voitovych – conceptualization of the study, literature review, design and implementation, simulation set-up and evaluation, analysis of results, drafting and final approval of the manuscript.

Sources of funding. This study did not receive any grant from a funding institution in the public, commercial, or non-commercial sectors.

Use of Artificial Intelligence tools. During the preparation of this article, the authors used the latest GPT-5.x models for content review, formatting, and rephrasing, and selected Anthropic Claude models to generate simple code fragments. All AI-generated outputs were carefully reviewed and edited by the authors, who take full responsibility for the final content of the article.

References

- Casau, P., Cabecinhas, D., & Silvestre, C. (2011). Autonomous transition flight for a vertical take-off and landing aircraft. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)* (pp. 3974–3979). IEEE. <https://doi.org/10.1109/CDC.2011.6160819>
- Chi, P., Wei, J., Wu, K., Di, B., & Wang, Y. (2023). A bio-inspired decision-making method of UAV swarm for attack-defense confrontation via multi-agent reinforcement learning. *Biomimetics*, 8(2), 222. <https://doi.org/10.3390/biomimetics8020222>
- Ekechi, C. C., Elfouly, T., Alouani, A., & Khattab, T. (2025). A survey on UAV control with multi-agent reinforcement learning. *Drones*, 9(7), 484. <https://doi.org/10.3390/drones9070484>
- Feng, Z., Na, X., Hai, S., Sun, Q., & Shi, J. (2025). Deep reinforcement learning for UAV target search and continuous tracking in complex environments with Gaussian process regression and prior policy embedding. *Electronics*, 14(7), 1330. <https://doi.org/10.3390/electronics14071330>
- Liu, M., Wei, J., & Liu, K. (2024). A two-stage target search and tracking method for UAV based on deep reinforcement learning. *Drones*, 8(10), 544. <https://doi.org/10.3390/drones8100544>
- Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments (arXiv:1706.02275). arXiv. <https://doi.org/10.48550/arXiv.1706.02275>
- Panerati, J., Zheng, H., Zhou, S., Xu, J., Prorok, A., & Schoellig, A. P. (2021). Learning to fly — a Gym environment with PyBullet physics for reinforcement learning of multi-agent quadcopter control (arXiv:2103.02142). arXiv. <https://doi.org/10.48550/arXiv.2103.02142>
- Sierra-García, J. E., & Santos, M. (2021). Intelligent control of a UAV with a cable-suspended load using a neural network estimator. *Expert Systems with Applications*, 185, 115380. <https://doi.org/10.1016/j.eswa.2021.115380>
- Xu, J., & Liang, G. (2025). Enhanced Q-learning and deep reinforcement learning for unmanned combat intelligence planning in adversarial environments. *Scientific Reports*, 15(1), 28364. <https://doi.org/10.1038/s41598-025-13752-3>
- Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. In K. G. Vamvoudakis, Y. Wan, F. L. Lewis, & D. Cansever (Eds.), *Handbook of reinforcement learning and control* (Studies in Systems, Decision and Control, Vol. 325, pp. 321–384). Springer. https://doi.org/10.1007/978-3-030-60990-0_12
- Zhao, B., Huo, M., Li, Z., Feng, W., Feng, W.-T., Yu, Z., Qi, N., & Wang, S. (2025). Graph-based multi-agent reinforcement learning for collaborative search and tracking of multiple UAVs. *Chinese Journal of Aeronautics*, 38(3), 103214. <https://doi.org/10.1016/j.cja.2024.08.045>

Отримано редакцію журналу / Received: 09.09.2025

Прорецензовано / Revised: 20.12.2025

Схвалено до друку / Accepted: 29.12.2025

Олексій БИЧКОВ, д.т.н., проф.

ORCID ID: 0000-0002-9378-9535

e-mail: oleksiibychkov@knu.ua

Київський національний університет імені Тараса Шевченка, Київ, Україна

Михайло ВОЙТОВИЧ, асп.

ORCID ID: 0009-0003-0405-0304

e-mail: voytovychm@gmail.com

Київський національний університет імені Тараса Шевченка, Київ, Україна

НАДІЙНЕ ПРИЙНЯТТЯ РІШЕНЬ РОЄМ БПЛА В УМОВАХ НЕВИЗНАЧЕНОСТІ ШЛЯХОМ ПОЄДНАННЯ БАГАТОАГЕНТНОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ ТА ГІБРИДНИХ АВТОМАТІВ

Вступ. Рої безпілотних літальних апаратів (БПЛА) дедалі частіше застосовуються у складних та невизначених середовищах, зокрема в умовах радіоелектронної боротьби (РЕБ), де традиційні методи керування на основі жорстко заданих правил можуть бути недостатньо ефективними. В пов'язаних роботах застосовувалося багатоагентне

навчання з підкріпленням (MARL), щоб дозволити БПЛА навчатися кооперативної поведінки в таких умовах, а також досліджувалося розбиття складних місій (наприклад, пошук і супровід цілі) на фази для спрощення керування. Проте підходи, що базуються виключно на навчанні, можуть бути недостатньо робастними або вимагати надмірно тривалого навчання для врахування всіх крайових випадків.

Методи. Запропонований підхід інтегрує алгоритм Multi-Agent Deep Deterministic Policy Gradient (MADDPG) із керівним гібридним автоматом, який кодує високорівневі режими поведінки. Гібридний автомат визначає дискретні режими для різних фаз місії (наприклад, розвідка або супровід цілі) та аварійних ситуацій (наприклад, ухилення під час радіоелектронного придушення), де формальні умови переходу активують зміну режимів. У межах кожного режиму агенти-БПЛА виконують безперервні керуючі дії, навчені за допомогою MADDPG із парадигмою централізованого навчання та децентралізованого виконання. Для централізованого навчання рою БПЛА розроблено симуляційне середовище на основі Gym-PyBullet-Drones, яке включає реалістичну фізичну модель і модель впливу радіоелектронної боротьби.

Результати. Експерименти в середовищі моделювання показали, що запропонований гібридний підхід RL-автомат перевершує стандартну наскрізну базову модель MADDPG за всіма протестованими умовами. Гібридний рій демонстрував стабільно вищі показники успішності виконання місії, особливо в умовах ворожого глушіння, де рівень успіху перевищував 90%, тоді як у базовій моделі спостерігалася значне зниження ефективності. Криві навчання показали швидшу збіжність і вищі фінальні значення винагороди для гібридного методу, що відображає переваги структурованого навчання з урахуванням режимів. У сценаріях із динамічними цілями гібридний рій підтримував довший безперервний супровід завдяки скоординованій тактиці, тоді як базова модель часто втрачала контакт. Ці результати демонструють, що інтеграція формального автомата з багатоагентним навчанням з підкріпленням забезпечує більш надійне, ефективне та масштабоване прийняття рішень роєм БПЛА в умовах невизначеності. Отримані висновки також узгоджуються з сучасними дослідженнями, які показують, що поділ місії на фази та навчання спеціалізованих для режимів стратегій може суттєво підвищити продуктивність порівняно з монолітними політиками.

Висновки. Запропонована гібридна модель демонструє, що поєднання перемикачів режимів на основі формальних автоматів із глибинним навчанням з підкріпленням може забезпечити більш надійне та ефективне прийняття рішень для роїв БПЛА в умовах невизначеного та ворожого середовища. Дискретне керування режимами вводить предметно-орієнтовані знання (наприклад, коли змінювати завдання або переходити в безпечний режим), що спрямовує процес навчання та забезпечує дотримання обмежень безпеки, тоді як MADDPG оптимізує безперервне керування в межах кожного режиму. Такий гібридний підхід забезпечує кращу адаптивність до змінних загроз і умов середовища, що підтверджується стійкою роботою в умовах радіоелектронної боротьби. У подальших дослідженнях планується розширити цей підхід на більш складні сценарії із використанням гетерогенних роїв БПЛА та дослідити автоматичне виявлення режимів (наприклад, за допомогою ієрархічного навчання з підкріпленням), щоб зменшити залежність від автоматів, визначених експертами.

Ключові слова: багатоагентне навчання з підкріпленням; рій БПЛА; гібридні автомати; невизначені середовища; радіоелектронна боротьба; Multi-Agent Deep Deterministic Policy Gradient; автономне прийняття рішень.

Автори заявляють про відсутність конфлікту інтересів. Один із співавторів є членом редколегії видання, тому не брав участі у рецензуванні та прийнятті рішення щодо публікації цієї статті. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. One of the co-authors is a member of the editorial board of the publication, therefore he did not participate in the review and decision-making regarding the publication of this article. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.