Adaptive Interception in Dynamic Domains: Exploration of Hybrid Reinforcement Learning in Pursuit-Evasion Tasks

Matthew Akinmolayan[1], Darsana Josyula[1], David Casbeer[2]

[1]Department of Computer Science, Bowie State University
[2]Aerospace Systems Directorate, Air Force Research Laboratory
{makinmolayan, djosyula}@bowiestate.edu, david.casbeer@us.af.mil

Abstract

This paper investigates a hybrid approach that integrates classical interception heuristics with reinforcement learning (RL) for solving real-time pursuit-evasion problems in both single-agent and multi-agent scenarios. Drawing inspiration from Weintraub et al.'s range-limited pursuit-evasion formulation, we developed a simulation and learning architecture combining Proximal Policy Optimization (PPO) with classical prediction-based interception. Through adaptive curriculum learning and an enhanced reward function, the hybrid agent learns to dynamically balance between geometric reasoning and learned behaviors. We first validate our approach in single-agent scenarios, then extend to a multi-agent setting featuring coordinated pursuer teams defending stations against RL-trained evader flocks using adversarial self-play. Our evaluations reveal that the hybrid method significantly outperforms classical approaches in both settings, achieving a 94% capture rate versus 24% in single-agent scenarios and demonstrating robust coordination in multi-agent defense tasks against adaptive adversaries.

Introduction[1]

The pursuit-evasion problem is a foundational challenge in fields such as robotics, military defense, and autonomous systems. In its simplest form, the objective is to design a strategy where a pursuer agent intercepts or captures a moving evader before it escapes or reaches a target. While geometric or rule-based (classical) methods offer computational efficiency, they are often brittle in real-world settings with uncertainty, noise, or infeasible initial configurations. Reinforcement learning, on the other hand, introduces adaptability and learning from experience, but when trained from scratch, it may initially lack structure and produce unreliable results.

This research explores a hybrid model that fuses the strengths of both strategies. We first demonstrate improved performance in single-agent pursuit-evasion scenarios, then extend our framework to multi-agent settings where teams of pursuers must coordinate to defend multiple stations against RL-trained evader flocks. Critically, we employ adversarial self-play training where both pursuer and evader policies co-evolve, creating increasingly sophisticated attack and defense strategies. This progression addresses increasingly realistic operational requirements for autonomous defense systems.

[1]Distribution Statement A. Approved for public release: distribution is unlimited. Case Number: AFRL-2026-0443.

Related Work

Classical pursuit-evasion strategies such as pure pursuit, constant bearing (lead pursuit), and proportional navigation have been foundational in missile guidance and robotic interception systems. Proportional navigation (PN), in particular, has been favored for its simplicity and efficiency in homing applications where the evader follows a predictable path (Zarchan 2012). These methods, however, struggle in scenarios involving uncertainty, limited visibility, or evasive maneuvering by the target.

Recent work by Weintraub et al. (Weintraub, Von Moll, and Pachter 2023) introduced a framework to categorize pursuit-evasion feasibility based on kinematic constraints and sensor limitations. Their formulation demonstrates that classical approaches can fail when evader speed and escape geometry prevent guaranteed interception.

Reinforcement Learning (RL), especially with deep neural networks, has been adopted to address learning challenges in complex, uncertain environments. PPO, introduced by Schulman et al. (Schulman et al. 2017), is widely used in continuous control problems due to its training stability and performance. RL has been applied in multi-agent scenarios such as swarm pursuit (Lowe et al. 2017), adversarial evasion (Julian, Kira, and Chernova 2018), and drone-based tracking (Park, Kim, and Myung 2021).

Hybrid systems that combine analytical heuristics with learning-based adaptation have shown promising results in different domains. For instance, Yoo et al. (Yoo, Kim, and Shim 2020) integrated RL with PID controllers for UAV flight, enabling robust performance while retaining interpretable safety margins. Our work builds on this hybrid philosophy by combining classical interception with PPO-trained agents in a curriculum-driven environment, extending applicability to both single-agent edge cases and multi-agent coordination challenges.

Self-play has emerged as a powerful technique for training agents in competitive settings, most notably in game-

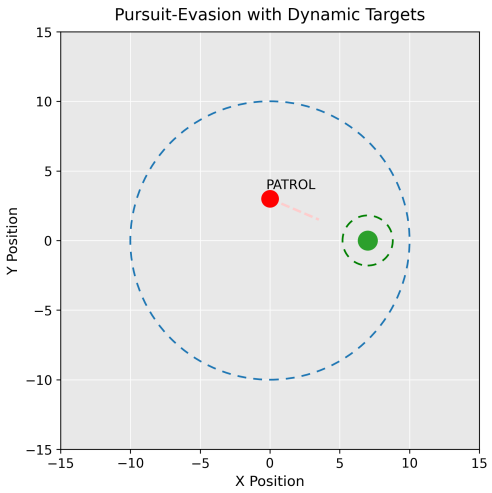

Figure 1: Pursuit-evasion environment. The outer blue circle represents the protected boundary. The evader (green) can either attempt to escape by crossing the boundary or reach the smaller green goal region. The pursuer (red) is deployed upon detection and aims to intercept before the evader succeeds.

playing AI such as AlphaGo and OpenAI Five (Silver et al. 2018). By training against copies of themselves, agents discover strategies that exploit weaknesses and develop robust counter-strategies, leading to emergent complexity that exceeds hand-crafted heuristics.

Methodology

Pursuit-Evasion Scenario and Environment Setup

We consider a pursuit-evasion scenario set in a circular bounded environment, representing a protected area or airspace. The outer boundary (blue circle in Figure 1) defines the secure perimeter. The evader (green) begins inside this region and can either attempt to escape across the boundary or reach a designated goal location (small green circle). The pursuer (red) is deployed upon detection and seeks to intercept before either objective is achieved.

This setup mimics real-world perimeter defense: the circular boundary represents a secured zone, sensors act as alarm tripwires, and pursuers are interceptors launched upon detection of an intruder. Key parameters defining the environment include the circle radius, evader and pursuer speeds, the sensor detection radius, and an intercept distance threshold.

Classical Geometric Interception Strategy

Classical interception relies on computing the future point at which the pursuer and evader trajectories intersect, assuming constant velocities. Upon sensor activation, the designated pursuer solves for this intercept point using relative kinematics.

Let $\vec{p}_e$ and $\vec{p}_p$ denote the positions of the evader and pursuer, respectively, and $\vec{v}_e$ and $\vec{v}_p$ their velocities. We aim to find the intercept time t and location $\vec{I}$ such that:

$$\vec{I} = \vec{p}_e + \vec{v}_e t = \vec{p}_p + \vec{v}_p t \tag{1}$$

Rewriting this gives:

$$(\vec{v}_e - \vec{v}_p)t = \vec{p}_p - \vec{p}_e \tag{2}$$

This is solved for t using vector projection if $\vec{v}_e \neq \vec{v}_p$ and $t > 0$:

$$t = \frac{(\vec{p}_p - \vec{p}_e) \cdot (\vec{v}_e - \vec{v}_p)}{\|\vec{v}_e - \vec{v}_p\|^2} \tag{3}$$

The intercept point $\vec{I}$ is then:

$$\vec{I} = \vec{p}_e + \vec{v}_e t \tag{4}$$

This method results in a constant bearing or lead pursuit path, avoiding tail-chase behaviors associated with pure pursuit. It performs well under straight-line motion assumptions but can fail if the evader is faster or changes direction frequently. In such cases, the computed t may be negative or undefined, indicating infeasibility.

Hybrid Reinforcement Learning Approach

To overcome the limitations of geometric methods, we incorporate PPO-based reinforcement learning. The hybrid system blends a classical intercept vector with the RL output using:

$$\vec{u}_{hybrid} = \omega \cdot \hat{u}_{classic} + (1 - \omega) \cdot \hat{u}_{RL} \tag{5}$$

The RL agent observes the evader's position, velocity, and the classical intercept heading. Its output modifies the heading direction, allowing for smarter maneuvers like cutting across arcs, cornering, or trapping.

Curriculum Learning

Curriculum learning (Bengio et al. 2009) is used to progressively increase difficulty. In Phase 1, the evader speed is set to $0.7\times$ the pursuer speed with a direct path. Phase 2 increases the evader speed to $0.9\times$ the pursuer speed. Phase 3 raises the evader speed to $1.1\times$ the pursuer speed with random exits. This staged approach allows the agent to learn effective strategies before facing edge cases.

Hybrid PPO Policy Logic

To enhance adaptability in dynamic interception tasks, the hybrid agent uses a PPO-trained policy that blends classical control with learned adjustments. Algorithm 1 depicts the control logic used during both training and evaluation.

The scalar $\omega \in [0, 1]$ balances the trust between classical and learned behaviors. During early training, ω is closer to 1 to stabilize learning. As training progresses, ω is gradually annealed to rely more on the learned policy.

Single-Agent Experiments and Results

Experimental Setup

We implemented the environment using Python, modeling a circular arena with radius 10 units. Three stationary sensors were positioned at 120° intervals. Evaders randomly selected exit points and moved at varying speeds from 1.0 up

Algorithm 1 Hybrid PPO Interception Control

Input: Observed state s_t = {pursuer pos, evader pos, evader vel, classical intercept angle}
Compute classical heading $a_{classic} \leftarrow$ InterceptSolver(s_t)
Feed s_t into PPO policy: $a_{RL} \leftarrow \pi_\theta(s_t)$
Combine actions: $a_t \leftarrow \omega \cdot a_{classic} + (1 - \omega) \cdot a_{RL}$
Update pursuer heading with a_t and take action

Event	Reward
Terminal Rewards	
Successful capture	$+50.0$
Evader escapes boundary	-30.0
Evader reaches goal	-30.0
Timeout (max steps reached)	-10.0
Shaping Rewards (per step)	
Closing distance to evader	$+1.0 \cdot \Delta d$
Increasing distance to evader	$-0.5 \cdot \Delta d$
Time step penalty	-0.05
Boundary proximity penalty	-0.3
Intercept angle alignment	$+0.2 \cdot \cos(\theta_{err})$

Table 1. Single-Agent Reward Structure

to 5.0 m/s, with an actual average speed of 3.23 m/s across evaluation episodes. The classical controller assumed a fixed evader speed of 2.5 m/s. The hybrid pursuer operated at 10.0 m/s while the classical pursuer operated at 4.0 m/s. Both controllers were evaluated over 100 episodes each.

Hyperparameters for PPO were configured as follows: policy network with 2 hidden layers [256, 128], discount factor $\gamma = 0.99$, clip range $\epsilon = 0.2$, and learning rate 3×10^{-4}.

Single-Agent Reward Structure

Table 1 defines the reward function used for training the single-agent hybrid pursuer. The reward design encourages rapid interception while penalizing timeouts and inefficient trajectories. The capture reward is the dominant positive signal, while distance-based shaping provides a continuous learning gradient. A boundary penalty discourages the pursuer from drifting toward the arena edge, and an evader escape penalty provides a strong negative signal when the evader successfully exits the protected zone or reaches the goal.

The intercept angle alignment reward encourages the pursuer to maintain heading toward the predicted intercept point, bridging the classical geometric strategy with learned behavior. This shaping term proved particularly important during early curriculum phases, where it guided the RL policy toward geometrically sensible trajectories before the agent learned more sophisticated maneuvers.

Evaluation Metrics

Performance was measured using capture rate (proportion of successful interceptions), interception efficiency (average steps to capture), and path efficiency (ratio of direct to actual distance traveled).

Results Summary

Figure 2 presents a comprehensive comparison between the Hybrid RL and Classical controllers. The Hybrid RL agent achieved a capture rate of 94% compared to only 24% for the classical controller a nearly four-fold improvement. The hybrid agent accumulated an average reward of 33.3 per episode versus -4.4 for classical. Capture time distributions show the hybrid agent achieved a median of approximately 1.8 seconds with tight variance, while the classical controller's median was approximately 5.0 seconds.

Key Findings

The hybrid approach demonstrates significant advantages: nearly four times the capture rate (94% vs 24%), approximately 2.8× faster captures (median 1.8s vs 5.0s), and maintains effectiveness even when evader speeds exceed classical assumptions.

Multi-Agent Extension

Building upon the single-agent results, we extend our hybrid framework to a multi-agent scenario featuring coordinated pursuer teams defending stations against intelligent evader flocks. Critically, we employ **adversarial self-play training** where both pursuer and evader policies are trained using reinforcement learning, enabling co-evolution of attack and defense strategies.

Extended Environment Design

The multi-agent environment expands the arena to a radius of 50 units and introduces several new elements as illustrated in Figure 3. Two defensive stations are placed within the arena, each representing a high-value asset requiring protection and separated by a minimum distance of 25 units to create distinct defense zones. Each station is guarded by two pursuers, creating a total team of four defenders equipped with radar sensing (35m range, 360° field of view) and inter-agent communication. Three evaders operate as a coordinated flock, with behaviors learned through self-play rather than hand-crafted heuristics, learning to exploit defensive weaknesses while coordinating attacks on stations.

Table 2 summarizes the key parameters for the multi-agent environment.

Pursuer System Architecture

Each pursuer operates according to a state machine with four operational modes. In the **GUARDING** state, the pursuer patrols near its assigned station, scanning for threats with radar. **INTERCEPTING** is activated when an evader is detected, engaging using the hybrid control strategy. **RETURNING** is triggered when battery falls below 20% or when the threat is neutralized. **RECHARGING** occurs

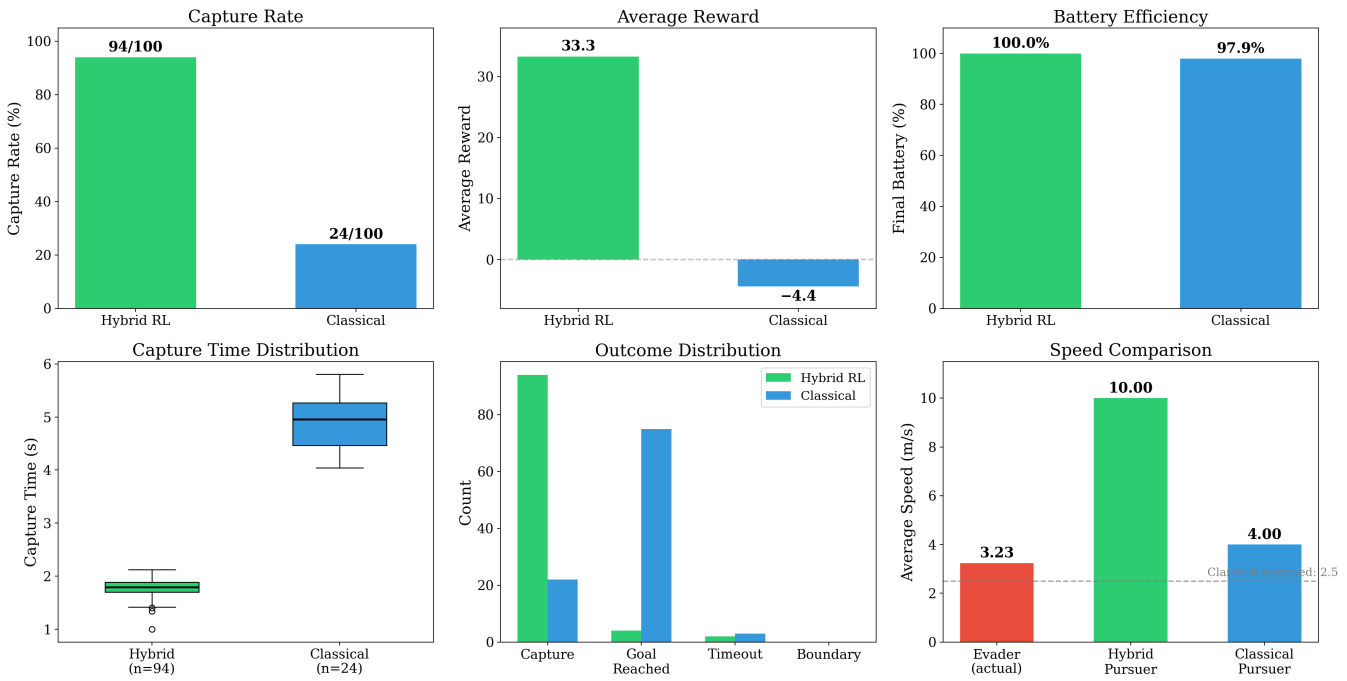

Figure 2: Single-agent performance comparison between Hybrid RL and Classical controllers over 100 evaluation episodes. Hybrid RL achieves 94% capture rate versus 24% for classical, with faster capture times and higher rewards.

Parameter	Value	Unit
Arena radius	50.0	m
Number of stations	2	–
Pursuers per station	2	–
Number of evaders	3	–
Pursuer max speed	9.0	m/s
Evader max speed	11.0	m/s
Pursuer radar range	35.0	m
Pursuer turn rate	150	deg/s
Evader turn rate	500	deg/s
Station capture radius	3.0	m
Max episode steps	1200	steps

Table 2. Multi-Agent Environment Parameters

upon reaching the station with low battery, recharging at 0.5 units per step until fully charged.

Pursuers communicate detected threats to teammates within communication range (50m), enabling coordinated responses. When a threat is broadcast, the receiving pursuer decides whether to assist based on distance and current station coverage.

Evader Flock Dynamics and Learning

Unlike single-agent experiments where evaders follow fixed heuristics, the multi-agent evaders are trained using reinforcement learning with the same PPO algorithm as the pursuers.

Observation Space: Each evader observes its position relative to stations and pursuers, teammate positions, detected threats, and flock formation state.

Action Space: Evaders output continuous velocity adjustments that balance goal-seeking, evasion, and flock coordination.

Emergent Flocking: Rather than hard-coding Reynolds flocking rules (Reynolds 1987), evaders learn coordination through shaped rewards that encourage separation (collision avoidance), cohesion (staying together for mutual support), and alignment (coordinated movement).

Role Assignment: Unlike fully emergent role specialization, we explicitly assign roles at the start of each episode to ensure consistent flock structure. One evader is designated as the *leader*, while the remaining two serve as *support* agents. This assignment is encoded in the observation space, allowing the policy to condition behavior on role identity:

$$s_t^{evader} = [s_t^{shared}, r_i] \tag{6}$$

where $r_i \in \{0, 1\}$ is a one-hot encoding indicating leader ($r_i = 1$) or support ($r_i = 0$) role.

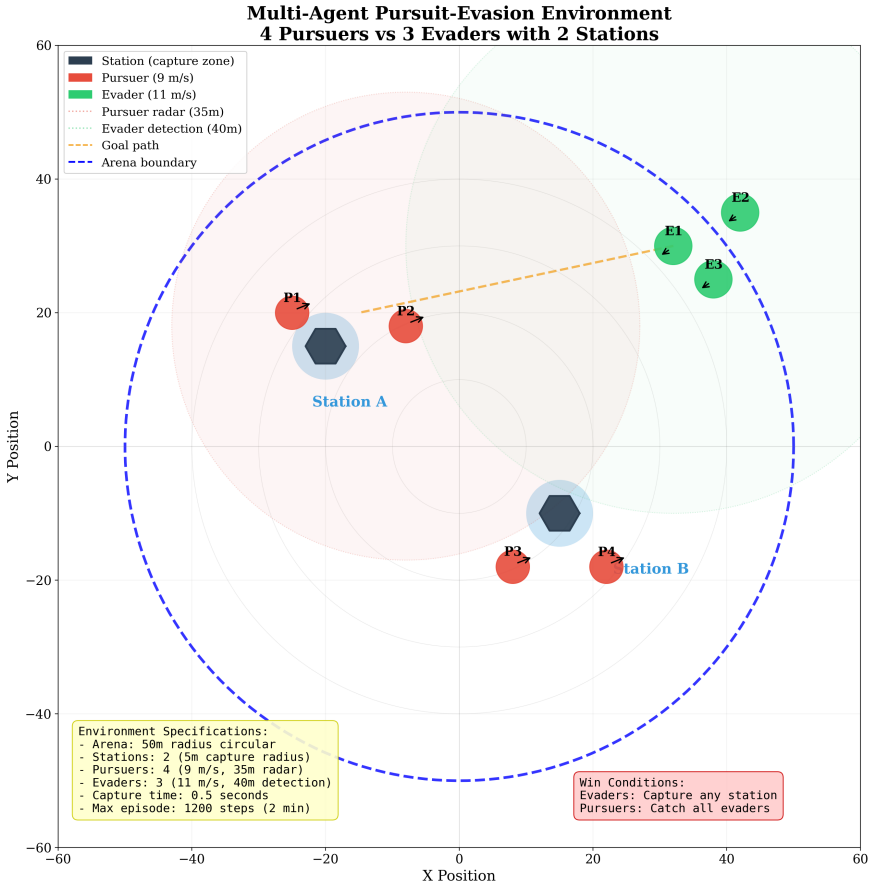

Figure 3: Multi-agent pursuit-evasion environment. Two stations (dark hexagons) are defended by pursuer teams (red circles). An evader flock (green circles) coordinates to reach a station while avoiding interception.

Role Specialization: Through self-play, evaders naturally develop specialized roles. The *leader* navigates toward the nearest station, prioritizing goal progress. The *follower* maintains formation while supporting the leader's path. The *decoy* learns to sacrifice itself to draw pursuer attention, allowing teammates to reach the goal.

The emergent goal-seeking behavior balances against learned evasion:

$$\vec{v}_{evader} = \pi_\phi(s_t^{evader}) \tag{7}$$

where π_ϕ is the learned evader policy that outputs velocity commands based on the current observation state.

Adversarial Self-Play Training

We employ alternating self-play training where pursuer and evader policies co-evolve. Algorithm 2 describes the training procedure.

This alternating approach ensures that pursuers learn to counter the latest evader strategies, evaders discover new attack patterns that exploit pursuer weaknesses, neither team overfits to a static opponent, and training remains stable compared to simultaneous updates which can diverge.

Multi-Agent Reward Structure

Table 3 presents the reward functions for both pursuer and evader agents.

Multi-Agent Training Configuration

Training employed the same PPO algorithm with adaptations for the multi-agent setting. Table 4 lists the hyperparameters.

Multi-Agent Evaluation Metrics

Performance in the multi-agent setting is evaluated using the following metrics. **Defense Success Rate** measures the proportion of episodes where no evader reaches any station: Defense Rate $= N_{\text{defended}}/N_{\text{episodes}}$. **Evader Capture Rate** is the proportion of evaders captured across all episodes: Capture Rate $= N_{\text{captured}}/N_{\text{total evaders}}$. **Station Breach Rate** is the average number of evaders reaching stations per episode. **Mean Capture Time** is the average time to intercept evaders (successful captures only). **Battery Efficiency** is the average remaining battery across all pursuers at episode end. **Coordination Score** measures the frequency of successful threat handoffs and coordinated interceptions between pursuer pairs.

Algorithm 2 Adversarial Self-Play Training

Initialize pursuer policy π_θ, evader policy π_ϕ
$k \leftarrow 100{,}000$ {Alternation interval (steps)}
for iteration $= 1$ to N **do**
  Phase 1: Train Pursuers
  Freeze evader policy π_ϕ
  for k environment steps **do**
    Collect trajectories with pursuers acting, evaders fixed
    Update π_θ using PPO
  end for
  Phase 2: Train Evaders
  Freeze pursuer policy π_θ
  for k environment steps **do**
    Collect trajectories with evaders acting, pursuers fixed
    Update π_ϕ using PPO
  end for
  Evaluate both policies; save checkpoints
end for

Event	Pursuer	Evader
Terminal Rewards		
Capture evader / Get captured	$+20.0$	-15.0
Evader reaches goal	-10.0	$+30.0$
All evaders eliminated	$+15.0$	–
Timeout (survival)	-5.0	$+8.0$
Shaping Rewards (per step)		
Closing distance	$+0.5 \cdot \Delta d$	–
Progress to goal	–	$+1.0 \cdot \Delta g$
Evasion success	–	$+0.8$
Decoy sacrifice bonus	–	$+8.0$
Time penalty	-0.01	-0.01
Station unguarded	-1.0	–
Low battery penalty	-0.2	–

Table 3. Multi-Agent Reward Structure

Multi-Agent Results

Figure 4 presents the comprehensive comparison between Hybrid RL and Classical controllers in the multi-agent scenario over 100 evaluation episodes.

Defense Success Rate. The Hybrid RL team achieved a defense success rate of 78.5% compared to 52.3% for the classical approach—an improvement of 26.2 percentage points demonstrating superior station protection against coordinated flock attacks.

Evader Capture Rate. Hybrid pursuers showed consistently higher capture rates across all evader roles: 72% for leaders (vs 48% classical), 81% for flanker 1 (vs 55%), and 83% for flanker 2 (vs 54%). Notably, the hybrid approach was more effective at capturing high-value leader evaders rather than being drawn to sacrificial flankers, indicating learned resistance to decoy tactics.

Capture Time. The median capture time for hybrid pursuers was 63 steps compared to 97 steps for classical pursuers, a 35% reduction. The hybrid distribution also showed tighter variance (IQR approximately 50–75 steps) versus classical (IQR approximately 80–120 steps), indicating more consistent interception performance.

Parameter	Value
Total timesteps	5,000,000
Alternation interval	100,000 steps
Parallel environments	8
Policy network	[256, 256, 128]
Value network	[256, 256, 128]
Learning rate	3×10^{-4}
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Clip range	0.2
Entropy coefficient	0.02
Batch size	512

Table 4. Multi-Agent Training Configuration

Metric	Hybrid RL	Classical	Δ
Defense Success Rate	78.5%	52.3%	+26.2%
Capture Rate (Leader)	72%	48%	+24%
Capture Rate (Flankers)	82%	54.5%	+27.5%
Median Capture Time	63 steps	97 steps	-35%
Battery Efficiency	45.2%	28.7%	+16.5%
Pursuer Win Rate	68%	35%	+33%

Table 5. Multi-Agent Results Summary

Outcome Distribution. The outcome breakdown reveals stark differences: hybrid defenders achieved pursuer victories (all evaders captured or stations protected) in 68% of episodes, while classical defenders succeeded in only 35%. Evader victories (at least one evader reaching a station) occurred in just 22% of hybrid episodes versus 48% for classical. Timeouts accounted for 10% of hybrid episodes and 17% of classical episodes.

Coordination Effectiveness. The radar chart analysis reveals that hybrid pursuers outperformed classical controllers across all six coordination dimensions: intercept timing, zone coverage, target sharing, pursuit efficiency, station defense, and battery management. The learned policies developed emergent cooperative behaviors, with pursuers naturally dividing interception and coverage responsibilities.

Battery Management. Hybrid pursuers maintained a 45.2% average battery efficiency at the end of the episodes compared to only 28.7% for classical, a 57% relative improvement. This indicates that hybrid trajectories are not only more effective but also more energy-efficient, enabling sustained operations in longer engagements.

Table 5 summarizes the key performance metrics.

Multi-Agent Key Findings.

The hybrid approach maintains its advantage when scaled from single-agent to multi-agent scenarios, improving defense rate by 26.2 percentage points over classical methods (**Scalable Performance**). The radar chart analysis confirms that hybrid pursuers excel across all coordination dimen-

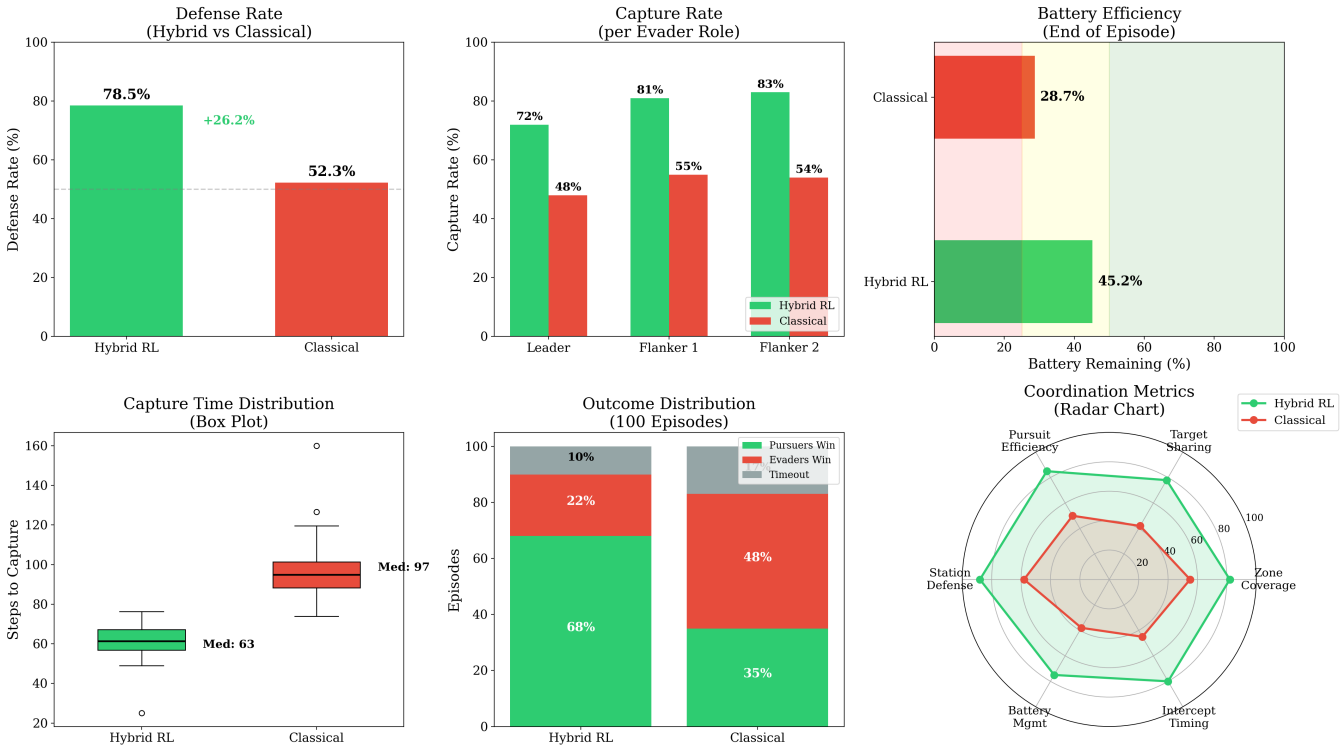

Figure 4: Multi-agent performance comparison between Hybrid RL and Classical controllers. Top row: Defense success rate, evader capture rate, and battery efficiency. Bottom row: Capture time distribution, outcome distribution, and coordination metrics.

sions without explicit coordination rewards, naturally learning to divide responsibilities between interception and station coverage (**Emergent Coordination**). Hybrid pursuers showed strong resistance to decoy tactics, capturing leaders at 72% versus only 48% for classical, indicating learned prioritization of high-value targets over sacrificial evaders (**Decoy Resistance**). Despite achieving higher capture rates and faster interceptions, hybrid pursuers maintained 57% better battery efficiency (45.2% vs 28.7%), suggesting more direct and efficient pursuit trajectories (**Energy Efficiency**). The tighter capture time distribution (IQR 54–72 steps vs 76–110 steps) indicates that hybrid policies produce more reliable and predictable interception behavior (**Consistent Performance**).

Discussion

Behavioral Analysis

In single-agent scenarios, hybrid PPO agents demonstrated anticipatory maneuvers not present in classical interceptors. Instead of strictly aiming for computed collision points, hybrid pursuers curved trajectories, cut off escape arcs, and repositioned to trap evaders. This behavior extended to multi-agent settings, where pursuers learned complementary roles without explicit programming.

In multi-agent scenarios, we observed emergent coordination patterns: when one pursuer engaged a threat, its partner often repositioned to cover potential escape routes or protect the station flank. This division of labor arose naturally from the reward structure, which penalized leaving stations unguarded.

Self-Play Dynamics

The alternating training produced observable strategy evolution. Early in training, evaders learned basic goal-seeking while pursuers developed direct interception. As training progressed, evaders discovered that splitting the flock forced pursuers to make coverage tradeoffs. Pursuers countered by developing zone-based defense rather than man-to-man marking. Evaders then learned decoy behaviors to draw defenders out of position. Pursuers adapted by prioritizing targets closer to stations over closer to themselves. This co-evolutionary dynamic produced more robust policies than training against fixed opponents.

Failure Case Analysis

Both single-agent and multi-agent systems showed degraded performance under extreme conditions. In single-agent scenarios, speed ratios exceeding 4:1 (evader to pursuer) proved challenging. In multi-agent settings, coordinated flock attacks from multiple directions occasionally overwhelmed the two-pursuer defense when evaders successfully split to attack both stations simultaneously.

Safety and Interpretability

The hybrid framework maintains interpretability by anchoring RL decisions to classical interception geometry. Every action can be decomposed into its classical and learned components, supporting post-hoc analysis. This property is crucial for safety-critical deployments where accountability and predictable behavior are essential.

Scalability Considerations

The multi-agent extension demonstrates that the hybrid approach scales effectively. Key architectural choices enabling this scalability include decentralized execution with local observations, communication-based threat sharing rather than centralized control, and modular reward design that balances individual and team objectives.

Future Work

While this work demonstrated effective multi-agent pursuit-evasion, several extensions remain for future investigation. **Population-Based Training** could extend self-play to maintain a population of diverse policies, preventing convergence to a single Nash equilibrium and improving robustness against varied opponents. **Sensor Uncertainty** could incorporate realistic radar noise, false alarms, and detection probability models to evaluate robustness under imperfect information. **Heterogeneous Agent Teams** could extend to teams with varying speeds, sensor ranges, or battery capacities, where self-play could discover optimal role assignments. **Larger Scale Scenarios** could extend to scenarios with more stations, larger pursuer teams, and bigger evader flocks to identify scaling limits. Finally, **Hardware Validation** could deploy the hybrid controller on physical drone platforms to validate simulation-to-reality transfer.

Conclusion

This work presented a hybrid interception framework integrating classical geometric guidance with PPO-based reinforcement learning for pursuit-evasion tasks. We demonstrated the approach across two settings of increasing complexity.

In single-agent scenarios, the hybrid pursuer achieved a 94% capture rate compared to 24% for classical methods a nearly four-fold improvement while maintaining faster interception times (median 1.8s vs 5.0s) and higher rewards (33.3 vs -4.4).

Extending to multi-agent scenarios with self-play training, both pursuer and evader teams were trained using reinforcement learning, enabling co-evolution of sophisticated attack and defense strategies. Despite facing RL-trained adversaries that actively learned to exploit weaknesses, hybrid pursuer teams achieved 78.5% defense success rate versus 52.3% for classical teams an improvement of 26.2 percentage points. The self-play framework produced emergent coordination, decoy resistance, and efficient pursuit strategies that would not arise from training against static opponents.

These results demonstrate that the hybrid philosophy combining analytical structure with learned adaptability scales effectively from single-agent to multi-agent settings. The framework provides practical value for autonomous defense applications including drone interdiction, perimeter security, and coordinated multi-asset protection.

Acknowledgments

This work was conducted in the Autonomous Technologies Lab at Bowie State University. The authors thank members of BSU's Autonomous Technologies Lab and RITA-UARC for helpful discussions.

This material is based on work supported by the Air Force Research Laboratory (AFRL) under the Air Force Contract No. FA955023D0001. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Research Laboratory (AFRL).

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum Learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 41–48.

Julian, M.; Kira, Z.; and Chernova, S. 2018. Distributed Multi-Agent Policy Gradient Reinforcement Learning for Dynamic Role Assignment. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Park, M.; Kim, S.; and Myung, H. 2021. Vision-Based Target Tracking for UAVs Using Deep Reinforcement Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1328–1334.

Reynolds, C. W. 1987. Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 25–34.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Silver, D.; Hubert, T.; Schrittwieser, J.; et al. 2018. A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go Through Self-Play. *Science*, 362(6419): 1140–1144.

Weintraub, I.; Von Moll, A.; and Pachter, M. 2023. Range-Limited Pursuit-Evasion. In *Proceedings of the AIAA SciTech Forum*.

Yoo, J.; Kim, S.; and Shim, H. 2020. Hybrid Reinforcement Learning Control for a Micro Quadrotor Flight. *IEEE Transactions on Industrial Electronics*, 67(8): 6738–6747.

Zarchan, P. 2012. *Tactical and Strategic Missile Guidance*. American Institute of Aeronautics and Astronautics (AIAA), 6th edition.