POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

ACE - Autonomous Combat Environment: a Modular Reinforcement Learning Framework for Dogfighting

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Emanuele Greco

Advisor: Prof. Nicola Gatti

Co-advisor: Piergiuseppe Pezzoli

Academic year: 2024-2025

1. Introduction

Autonomous decision-making for air combat is becoming increasingly critical, as engagements require fast, high-quality decisions under uncertainty, often involving multiple adversaries and coordination requirements. This push is strengthened by next-generation air systems that rely on AI and teaming with autonomous UAVs [1, 2]. In this landscape, Deep Reinforcement Learning (RL) is a promising approach for learning adaptive behaviors in complex, adversarial settings [3, 4].

Within Visual Range (WVR) combat, colloquially known as dogfighting, is a demanding benchmark: fights unfold at short distances and high angular rates, where small timing and geometry changes can decide the outcome. An agent must control the aircraft continuously, respect flight-envelope and safety limits, and plan against an opponent that actively reacts and counters.

This thesis develops and evaluates RL-based dogfighting agents with an emphasis on reproducible benchmarking and a clear split between prototyping and realism. First, modular Gymnasium [5] toy-model environments are used to prototype observations, actions, and rewards, and to benchmark standard RL algorithms on key sub-tasks (interception, tracking, pursuit). Second, the same pipeline is transferred to a high-fidelity JSBSim [6] setup with realistic 6-DoF dynamics and constraints.

In this thesis, we explore a unified and modular framework for reinforcement learning in autonomous aerial combat, designed to support reproducible experimentation. The objective is not to impose predefined control strategies or expert-driven behaviors, but rather to investigate whether meaningful and effective behaviors can emerge directly from pure reinforcement learning.

2. Methods Overview

This work uses a standard RL formulation based on a Markov Decision Process (MDP), where the agent maps observations to actions to maximize expected return (Figure 1). Neural policies and value functions are represented with actor–critic architectures: MLPs are used by default, while recurrent memory is adopted when temporal context is needed. Implemented methods across the toy-model environments and the JSBSim stage are:

- PPO (on-policy) [7]

- TD3 (off-policy) [8]
- SAC (off-policy, entropy-regularized) [9]
- rPPO with LSTM (on-policy) [10]

Figure 1: Visual representation of an MDP.

3. Toy Model

A first stage uses custom Gymnasium toy-model environments to prototype the full RL pipeline under fast and controllable simulation. Each environment isolates a key capability and increases complexity step by step, so that observation design, action parameterization, reward shaping, and training stability can be validated before moving to high-fidelity flight dynamics.
The toy-model roadmap includes:

- FollowPoint: minimal 3D point reaching with kinematic updates
- FollowTrajectory: tracking a moving target along randomized 3D Bézier paths to stress generalization
- Tag: two-agent pursuit–evasion with warm-start and Frozen Opponent Self-Play
- Fight: a simplified dogfight task with tactical geometry, including standard tracking angles and a Weapon Engagement Zone (WEZ) concept, plus hit/win signals.

Across these tasks, the toy stage enables rapid debugging and ablation studies while keeping the interfaces consistent with the later JSBSim setup.

4. JSBSim Simulator

JSBSim is used as the realistic validation stage of the framework, based on a physics-based 6-DoF rigid-body aircraft model (F-16). The simulator integrates coupled translational and rotational dynamics, expressed through standard reference frames and attitude angles (roll–pitch–yaw), as illustrated in Figure 2. In this setting, control is no longer kinematic: aerodynamic coupling, actuator limits, and delayed responses make the learning problem significantly harder and closer to real flight.

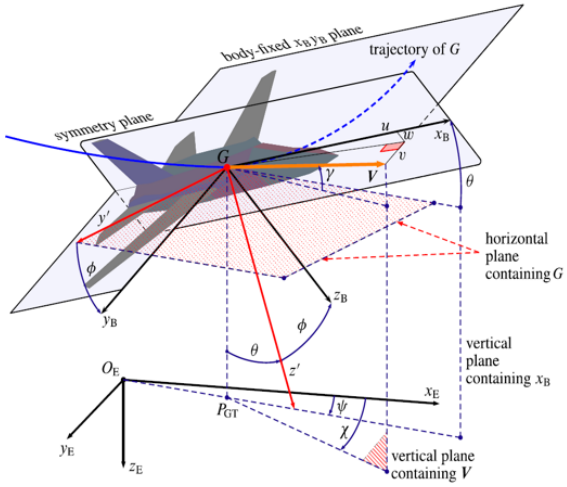


Figure 2: Reference frames and attitude angles used to describe a 6-DoF rigid-body aircraft motion [2].

To connect RL to the aircraft, the agent outputs continuous commands that are mapped to primary control surfaces and throttle. In particular, the action vector is:

$$a = [\text{aileron}_{cmd}, \text{elevator}_{cmd}, \text{rudder}_{cmd}, \text{throttle}_{cmd}]$$

with normalized ranges for surfaces and a constrained throttle range to avoid extreme regimes. The role and deflection logic of the surfaces is summarized in the Figure 3 (ailerons modulate roll, elevator changes pitch, rudder modifies yaw).

Figure 3: Main aircraft control surfaces and their typical deflection directions.

The agent observes a fixed, shared state representation across tasks to keep comparisons consistent. The observation is a 20-dimensional vector including position, attitude, velocities in both NED and body frames, calibrated airspeed,

accelerations, aerodynamic angles (Angle of Attack α, Sideslip Angle β), and angular rates (p, q, r). This design supports both single-agent flight tasks and multi-agent dogfighting without introducing task-specific shortcuts.
For dogfighting, the simulator also relies on standard combat-geometry angles (shown in Figure 4), such as ATA, AA, and HCA, which compactly describe line-of-sight alignment, tail aspect, and heading crossing. These variables are used both as evaluation metrics and (in the fully observable setting) as explicit inputs to reduce ambiguity in credit assignment.

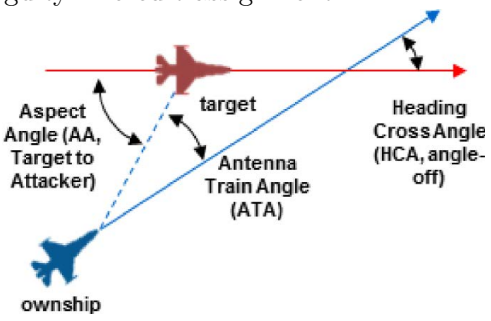


Figure 4: Visualization of tactical tracking angles in air-combat geometry.

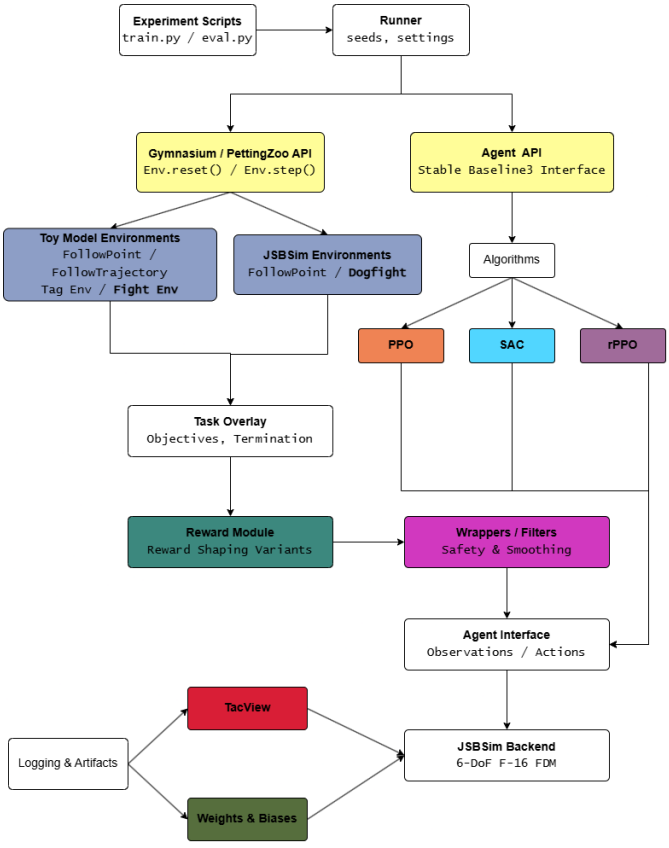


Figure 5: Block diagram showing the full modular framework. Colored blocks represent modular components.

To support reproducibility and fair algorithmic comparison, we developed the framework shown in Fig. 5, which emphasizes modularity and plug-and-play integration.

5. Environments

All ToyModel environments were analyzed, as well as several environments from the JSBSim model. However, for brevity, this document focuses only on the two most relevant ones.

5.1. FollowPoint

FollowPoint is a single-agent guidance task used to validate goal-reaching under realistic 6-DoF dynamics. The aircraft must reach a randomly generated fixed 3D target while staying within altitude bounds; success is defined by a distance threshold, and failures are triggered by leaving the safe altitude band or by exceeding the step budget.
The reward is mainly an error-penalization shaping that combines altitude error, heading error, attitude leveling, and normalized distance, with sparse terminal events for success/failure. The final reward is:

$$r_{\text{dense}} = -w_h \frac{|e_h|}{H_s} - w_\psi \frac{|e_\psi|}{\pi} - w_{\text{lvl}}\big(|e_\phi| + |e_\theta|\big) - k_d \frac{d}{d_0}$$

$$r = r_{\text{dense}} + \begin{cases} -500 & \text{if } h \le 1000 \text{ or } h \ge 10000, \\ +500 & \text{if } d \le 200, \\ 0 & \text{otherwise.} \end{cases}$$

This dense reward penalizes altitude, heading, roll/pitch, and distance errors, each as a weighted term.

5.2. Dogfight

Dogfight is the main benchmark of the thesis: two agents fight in a realistic 3D WVR setting with JSBSim F-16 dynamics. The environment uses multiple initial combat scenarios to reduce overfitting to a single opening geometry, explicitly studying reward coupling (Zerosum vs Non-Zerosum). No curriculum or imitation learning is used: behaviors must emerge from reward-driven Frozen Opponent Self-Play [11] under full aircraft constraints.
Two reward formulations are considered for the JSBSim dogfight task. A Zerosum version,

which enforces strict competition by construction:

$$\mathrm{adv}(t) = \tanh\big(\alpha\,(s_0(t) - s_1(t))\big).$$

$$(r_0(t),\, r_1(t)) = \begin{cases} (-R_{\text{crash}},\, +R_{\text{crash}}) & \text{if } \mathrm{crash}_0(t) \wedge \neg\mathrm{crash}_1(t), \\ (+R_{\text{crash}},\, -R_{\text{crash}}) & \text{if } \neg\mathrm{crash}_0(t) \wedge \mathrm{crash}_1(t), \\ (0,\, 0) & \text{if } \mathrm{crash}_0(t) \wedge \mathrm{crash}_1(t), \\ (0,\, 0) & \text{if } \mathrm{kill}_0(t) \wedge \mathrm{kill}_1(t), \\ (\mathrm{adv}(t) + R_{\text{kill}},\, -\mathrm{adv}(t) - R_{\text{kill}}) & \text{if } \mathrm{kill}_0(t) \wedge \neg\mathrm{kill}_1(t), \\ (\mathrm{adv}(t) - R_{\text{kill}},\, -\mathrm{adv}(t) + R_{\text{kill}}) & \text{if } \neg\mathrm{kill}_0(t) \wedge \mathrm{kill}_1(t), \\ (\mathrm{adv}(t),\, -\mathrm{adv}(t)) & \text{otherwise,} \end{cases}$$

with $s_i(t), i \in \{0, 1\}$ being the per-agent dense score, and a Non-Zerosum version, which combines engagement, safety, and outcome terms to improve training stability:

$$r_i^{\text{eng}}(t) = r_i^{\text{base}}(t) + r_i^{\text{close}}(t) + r_i^{\text{WEZ}}(t) + r_i^{\text{kill}}(t)$$

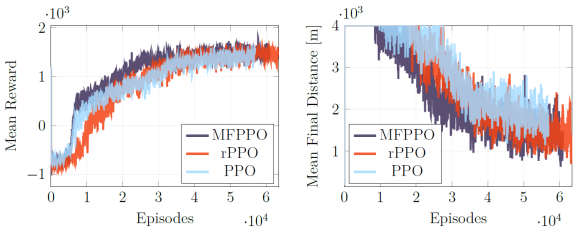
$$\begin{aligned} r_i(t) = \lambda_i(t)\, r_i^{\text{eng}}(t) + r_i^{\text{alt}}(t) + r_i^{\text{dmg}}(t) \\ + r_i^{\text{death}}(t) + r_i^{\text{far}}(t) + r_i^{\text{time}}(t) + r_i^{\text{inv}}(t) \end{aligned}$$

which combines an engagement part and several safety/regularization terms. The engagement reward includes r_i^{base} (geometry shaping: preferred range, LOS positioning and nose alignment), r_i^{close} (reward for reducing distance), r_i^{WEZ} (bonus if you are in WEZ and penalty if the opponent is), and r_i^{kill} (sparse hit/destroy event). On top of that, it adds r_i^{alt} (penalize unsafe altitudes and sinking while low), r_i^{far} (discourage disengagement), r_i^{inv} (penalize excessive roll/inversion), r_i^{time} (small per-step time cost), plus r_i^{dmg} and r_i^{death} to penalize taking damage and being destroyed.

6. Remarkable Results

6.1. FollowPoint

In this experiment, three PPO-based variants are compared: standard PPO, a recurrent PPO (rPPO), and a Multi-Frame PPO (MFPPO) baseline that stacks observations over time. Training trends show a consistent learning pattern across methods: as experience accumulates, mean episode reward increases while the mean final distance to the target decreases, indicating that the agent progressively learns a viable closed-loop guidance strategy (Fig. 6). Among the compared approaches, MFPPO reaches higher rewards earlier and reduces final distance more decisively during the early-to-mid training phase, suggesting that explicit temporal context through frame stacking improves short-horizon prediction of target-relative dynamics.

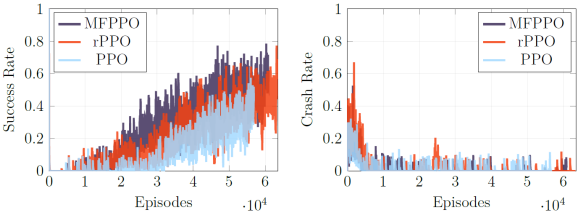


(a) Mean episode reward (b) Mean final distance

Figure 6: FollowPoint training reward and final distance from target.

rPPO follows a similar trajectory, with slightly slower early gains but a steadier progression, consistent with the role of recurrent memory in filtering short-term observation noise and stabilizing control. Standard PPO remains systematically behind, with slower and noisier improvement and a weaker reduction in final distance, indicating that single-step observations can be insufficient to consistently infer turning dynamics in this flight-control setting.

Success rate increases for all methods, but it grows earlier and becomes higher for MFPPO and rPPO, while PPO lags and shows larger variability (Fig. 7). At the same time, crash rate rapidly collapses toward zero after the initial exploration phase for every method, implying that catastrophic failures are mostly confined to the beginning of training.



(a) Success rate (b) Crash rate

Figure 7: Success and crash rates in FollowPoint.

After basic stability is learned, the timeout rate rises sharply early on (the agent remains controlled but does not reach the target fast enough), and then decreases more for MFPPO and rPPO than for PPO. Both policy and value

losses decay rapidly from initial peaks and then settle into a lower-variance regime, with rPPO showing especially regular loss behavior in practice.

Evaluation results (Table 1) confirm the advantage of temporal context. Over 100 deterministic evaluation episodes, rPPO and MFPPO achieve higher average reward and better accuracy-oriented metrics than PPO.

The number of steps per episode is similar across methods in this setup, so the main improvement is expressed as higher success and better final positioning rather than shorter episodes. Frame stacking (MFPPO) improves early learning and target approach quality, while recurrent memory (rPPO) achieves comparable or better final performance with smoother, more realistic control behavior. This distinction is relevant for downstream use in flight tasks, where achieving the objective with fewer oscillations and less control "chatter" is an important practical requirement.

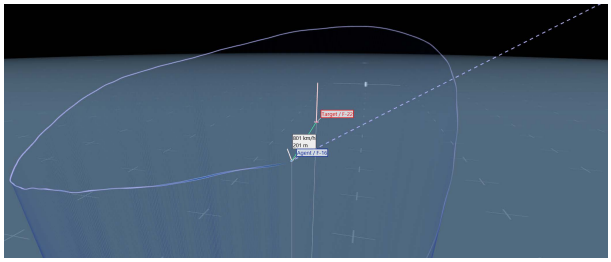


Figure 8: FollowPoint evaluation episode.

6.2. Dogfight

The Dogfight setting considers a fully observable air-combat scenario in JSBSim, where each agent receives a complete description of the engagement state. Training is carried out with the Non-Zerosum reward formulation, adopted because prior experiments indicated that it provides a more effective learning signal than strict Zerosum shaping in this environment.

The base reward over training (Fig. 9) remains noisy, which is expected in a dogfight setting where small timing and geometry differences can lead to abrupt regime changes. The Weapon Engagement Zone (WEZ) reward and the kill reward show increasingly frequent and persistent activation (Fig. 10), which indicates that, with sufficient training time, the agent is not only surviving or maneuvering, but repeatedly reaching geometries associated with weapon-effective positioning and, in some cases, converting them into terminal outcomes.

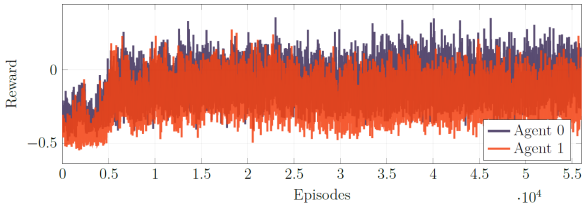
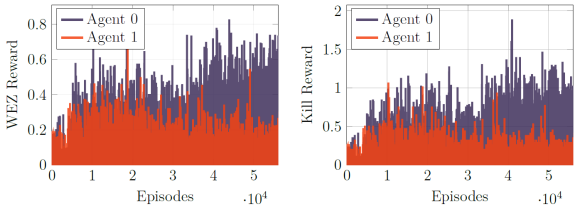


Figure 9: Dogfight base reward.



(a) WEZ reward (b) Kill reward

Figure 10: Dogfight engagement rewards.

The HP trends show an increasing asymmetry between the two agents over training. This suggests that the self-play process can converge toward a regime where one policy gains a systematic advantage that the counterpart does not fully counterbalance. In practical terms, this is consistent with the emergence of a "dominant" behavior under the current training alternation and reward structure. At the same time, tactical geometry indicators show wide fluctuations, reflecting the sensitivity of dogfight dynamics to maneuver timing, closure rate, and relative turning performance.

Over 10 deterministic evaluation episodes (Table 2), Agent 0 achieves substantially higher shot

Algorithm	Avg Reward	Steps/episode ↓	Avg distance ↓	Success ratio [%] ↑
PPO	942.3 ± 710.6	791.9 ± 287.5	2457.8 ± 1920.1	28.0 ± 45.1
MFPPO	1377.2 ± 533.6	791.3 ± 275.7	1849.1 ± 2258.6	44.0 ± 49.9
rPPO	1485.8 ± 330.8	791.1 ± 287.7	1435.6 ± 1376.5	44.0 ± 49.9

Table 1: FollowPoint results.

counts than Agent 1, with the reward values also reflecting this asymmetry, consistent with a learned dominance pattern. These quantitative results indicate that the training leads to policies that not only reach weapon-effective states but can repeatedly capitalize on them to produce damage events. This is a central outcome for the thesis, because it represents a transition from unstable or indecisive short-horizon behaviors to episodes where engagements are closed more decisively and more repeatably.

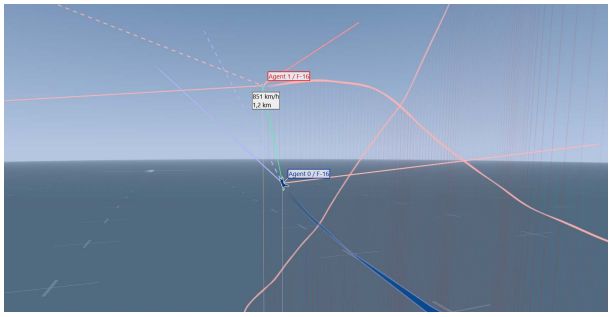


Figure 11: Dogfight evaluation episode.

These results show that selecting a Non-Zerosum shaping objective enables actionable dogfight behavior, and extending training time consolidates this behavior into measurable offensive effectiveness under self-play, particularly when the observation is fully informative. The main limitation visible in the results is the emergence of a strong asymmetry between the two agents, suggesting that long-horizon self-play may converge to a dominant policy pair rather than a balanced equilibrium. Nevertheless, this experiment provides the clearest evidence that the proposed JSBSim-based pipeline can produce long-horizon engagement behaviors with consistent damage outcomes, moving beyond mere survivability toward fully emergent combat interactions.

7. Conclusions

This thesis investigated Deep Reinforcement Learning (DRL) for autonomous aerial combat through a two-stage pipeline that moves from controlled toy-model environments to high-fidelity JSBSim simulation. The main result is that much of the ambiguity commonly found in prior air-combat RL work can be reduced by using our proposed modular and reproducible benchmarking setup, where tasks, observation variants, reward functions, training configurations, and evaluation protocols are explicitly separated and interchangeable. This structure makes comparisons more inspectable and repeatable, and it supports systematic ablations rather than relying on isolated "win-rate" style outcomes.

The toy-model experiments showed that with well-shaped rewards and episode design, pursuit and tracking can be learned reliably under fixed seeds and standardized metrics. Moving to JSBSim stressed the same pipeline under realistic 6-DOF dynamics: meaningful behaviors still emerged, but robustness and generalization became the main bottlenecks, with strong sensitivity to initialization, scenarios, rewards, and opponent non-stationarity. These issues mirror known challenges in the literature, such as reward brittleness, partial observability, and unstable self-play.

A key contribution of the work is the supporting infrastructure itself. Reproducibility is treated as a primary requirement via clean baselines, containerized execution, structured experiment tracking, systematic checkpointing, and detailed logging that enables analysis of trajectories and emergent behavior rather than only aggregate scores. As a result, the thesis provides a concrete reference framework to evaluate methods under consistent interfaces and comparable metrics, and to make remaining challenges explicit and measurable.

Future work should prioritize stabilizing adversarial training and improving tactical performance in JSBSim. Promising directions include league-based training with PFSP to reduce non-stationarity, and a hierarchical policy that separates high-level intent from low-level continuous control.

Algorithm	Reward (0) ↑	Reward (1) ↑	Shots Landed (0) ↑	Shots Landed (1) ↑
rPPO	0.082 ± 0.549	-0.850 ± 0.791	15.0 ± 4.1	1.7 ± 1.6

Table 2: Dogfight results.

References

[1] Adrian P. Pope, Jaime S. Ide, Daria Mićović, Henry Diaz, Jason C. Twedt, Kevin Alcedo, Thayne T. Walker, David Rosenbluth, Lee Ritholtz, and Daniel Javorsek. Hierarchical Reinforcement Learning for Air Combat at DARPA's AlphaDogfight Trials. *IEEE Transactions on Artificial Intelligence*, 4(6):1371–1385, December 2023. Conference Name: IEEE Transactions on Artificial Intelligence.

[2] Agostino De Marco, Paolo Maria D'Onza, and Sabato Manfredi. A deep reinforcement learning control approach for high-performance aircraft. *Nonlinear Dynamics*, 111(18):17037–17077, September 2023.

[3] Y. Han et al. Interpretable DRL-Based Maneuver Decision of UCAV Dogfight. 2024.

[4] Y. Qian et al. H3E: Learning air combat with a three-level hierarchical framework embedding expert knowledge. 2024.

[5] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[6] Jon Berndt. Jsbsim: An open source flight dynamics model. *JSBSim Journal*, 08 2004.

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. arXiv:1707.06347 [cs].

[8] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods, October 2018. arXiv:1802.09477 [cs].

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, August 2018. arXiv:1801.01290 [cs].

[10] Marco Pleines, Matthias Pallasch, Frank Zimmer, and Mike Preuss. Generalization, Mayhems and Limits in Recurrent Proximal Policy Optimization, May 2022. arXiv:2205.11104 [cs].

[11] Julien Hansen, Arthur Louette, Pascal Leroy, and Damien Ernst. Autonomous drone combat: A multi-agent reinforcement learning approach. ORBi (University of Liège Institutional Repository), September 2025. Eprint first made available on ORBi (University of Liège). Publication date: 10 September 2025.