Integrating Simulation and Machine Learning for Autonomous Aerial Combat Agents

João Paulo de Andrade Dantas[1], **Marcos R. O. A. Maximo**[1], **Takashi Yoneyama**[1]

[1]Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, SP, Brazil

{jpdantas,mmaximo,takashi}@ita.br

Abstract. *This work presents a research framework that brings together detailed simulation and machine learning to improve decision support and autonomy for aerial vehicles, with beyond-visual-range (BVR) air combat as a specific case study in the air domain. The framework covers five connected areas: a simulation environment and services; learning-based models of engagement zones and kill probability; in-flight tactical decision support for launch timing and situational awareness; tools for aerospace data analysis; and autonomous agents trained through reinforcement, imitation, and generative learning. This study focuses on the last area: agents learn cooperative tactics from reward and reproduce expert maneuvers from a small set of demonstrations. Although developed for air combat, the methodology can be applied to other applications in aviation, such as commercial flight operations and pilot training.*

Keywords: Autonomous agents; Reinforcement learning; Imitation learning; Aerospace simulation; Multi-agent coordination.

Level: Ph.D. **Thesis conclusion:** 27 June 2025.
Advisor: Takashi Yoneyama. **Co-advisor:** Marcos R. O. A. Maximo.

1. Introduction

Autonomy is a central theme in modern robotics, and aerial vehicles are among its most demanding platforms: they must perceive, decide, and act in environments that are fast, dynamic, partially observable, and often adversarial. In this setting, machine learning supports stronger situational awareness, faster decision-making, and agents able to operate alone or alongside human pilots [Dantas et al. 2024a].

Like most learning-based robots, these agents need large amounts of interaction data, which is scarce and costly to gather on real aircraft, and, in the defense context, often classified. Simulation is the standard answer: a safe, flexible, and low-cost environment to generate scenarios and to train, test, and validate models before deployment [Costa et al. 2025]. What makes it usable is fidelity, since the closer the simulated aircraft, sensors, and weapons are to the real ones, the more of what an agent learns survives the transfer.

Despite steady progress, simulation tools, weapon models, tactical decision aids, data-analysis pipelines, and autonomous agents are usually studied in isolation [Costa et al. 2025]. The same fragmentation appears inside the agent literature: reinforcement learning (RL) for beyond-visual-range (BVR) air combat is dominated by

one-versus-one engagements [Hu et al. 2021, Zhang et al. 2022], and work on larger formations still rewards each agent individually, so the reward carries no information about the ally [Liu et al. 2022, Zhou et al. 2024]. On the imitation side, behavior cloning has reproduced piloting skills [Sandström et al. 2022] and generative models such as variational autoencoders can enlarge small datasets [Kingma and Welling 2013], without a systematic account of when that augmentation helps. This motivates the central question of the doctoral thesis summarized in this paper [Dantas 2025]:

Research Question

How can a modern air force integrate simulation and machine learning to support decision-making and autonomy in air combat operations, and which areas are most critical to enabling this integration?

The thesis answers it with a unified framework built on five connected areas, described in Section 2. BVR air combat is the case study throughout: the dominant form of modern aerial engagement [Costa et al. 2025], and a challenging robotics problem, being multi-agent, partially observable, and adversarial under tight time constraints. The main contribution is the framework and the solutions developed within each area, all peer-reviewed and published, as summarized below.

Doctoral Research Outputs

The doctoral research produced **12 journal articles**, all accepted for publication, **6 of them in journals ranked above the 90th percentile** in their fields (Scopus), and **17 peer-reviewed conference papers**[1]. The journals include Scientific Data (Nature Portfolio) [Patrikar et al. 2025], IEEE Robotics and Automation Letters [Navarro et al. 2024], IEEE Access [Costa et al. 2025, Dantas et al. 2025d, Dantas et al. 2025c, Silva et al. 2025], the Journal of Defense Modeling and Simulation [Dantas et al. 2026], and IEEE Latin America Transactions [Dantas et al. 2022b]; the conferences include the ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS) [Dantas et al. 2023b, Dantas et al. 2024b, Dantas et al. 2024a], the IEEE Winter Simulation Conference (WSC) [Dantas et al. 2022a, Dantas 2024], the Latin American Robotics Symposium and Brazilian Symposium on Robotics (LARS/SBR) [Dantas et al. 2021a], and the Aerial Robotics Workshop of the IEEE International Conference on Robotics and Automation (ICRA) [Patrikar et al. 2022], among others. The research also delivered **7 software artifacts**, including **3** registered with the Brazilian National Institute of Industrial Property (INPI), **1** registered at Carnegie Mellon University, and **2** released as open source. Part of the research was carried out during a 14-month research exchange at the Robotics Institute, Carnegie Mellon University (November 2021 to January 2023).

This paper examines the fifth area, aerial autonomous agents, through learning from reward (Section 3) and from demonstrations (Section 4); Section 5 discusses lessons and limitations, and Section 6 concludes.

[1]Further information about the first author is available at `https://www.joaopadantas.com`.

2. Research Framework

The framework organizes the problem into five areas, shown in Figure 1. Simulation, machine learning, and air combat operations sit at the same level and form a cycle in which progress in each one feeds the others. Below them, the areas split into two branches, decision support and autonomy.

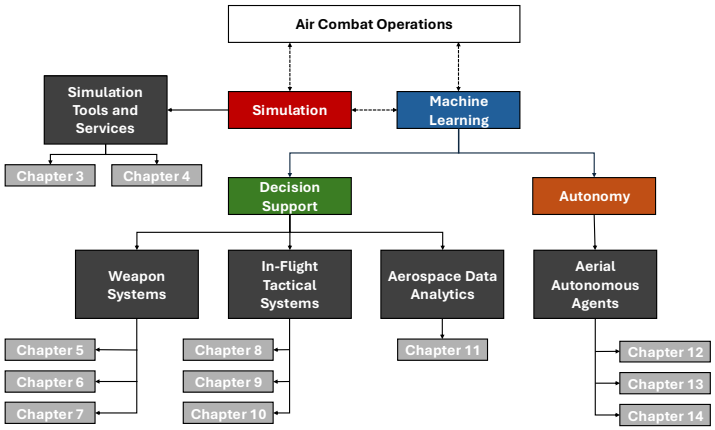


Figure 1. The five areas of development. The numbered boxes refer to the chapters of the thesis [Dantas 2025].

(I) Simulation tools and services. The Aerospace Simulation Environment (ASA), developed by the Institute for Advanced Studies (IEAv) of the Brazilian Air Force, simulates aircraft, radars, radar warning receivers, datalinks, and missiles [Dantas et al. 2025a], and is exposed as a set of services so large batches of scenarios run on demand [Dantas et al. 2025b], while a human-in-the-loop module lets pilots fly in the same scenarios [Silva et al. 2025]. Each aircraft is driven by a Behavior Tree (BT) encoding standard combat behaviors [Colledanchise and Ögren 2020].

(II) Weapon systems. Learned models estimate the Weapon Engagement Zone (WEZ) of air-to-air missiles [Dantas et al. 2021b], predict surface-to-air engagement zones in real time [Dantas et al. 2023a], and estimate probability of kill against stochastic targets [Dantas et al. 2026].

(III) In-flight tactical systems. Supervised models support decisions during the engagement, including missile launch timing and situational awareness from partial sensor information [Dantas et al. 2021a, Dantas et al. 2022a, Dantas et al. 2022b].

(IV) Aerospace data analytics. AsaPy is a Python library for design of experiments, execution control, and analysis of large simulation datasets [Dantas et al. 2024b].

(V) Aerial autonomous agents. Agents learn air combat behavior in two ways: RL inside AsaGym, a library that wraps ASA as a standard learning environment [Dantas et al. 2023b, Dantas et al. 2025d], and imitation learning from recorded pilot flights, with generative models synthesizing further trajectories [Dantas 2024, Dantas et al. 2025c].

3. Learning Tactical Behavior with Deep Reinforcement Learning

The first approach asks an agent to discover tactical behavior on its own, from interaction and reward [Dantas et al. 2023b, Dantas et al. 2025d].

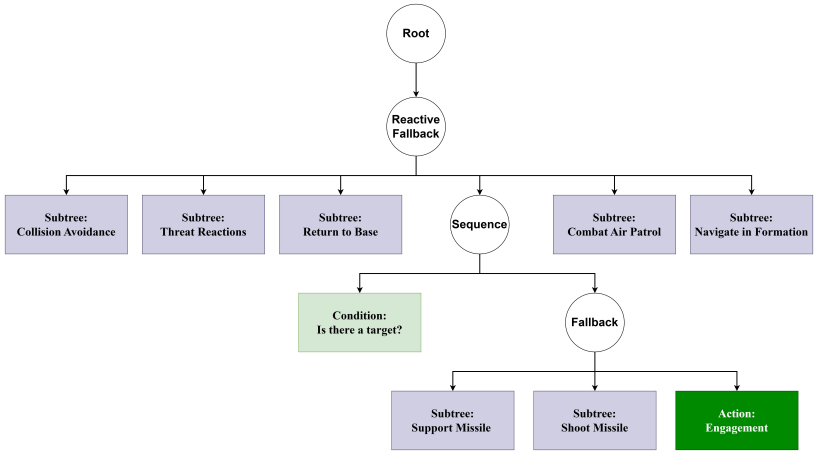


Figure 2. Simplified Behavior Tree of the fighter agent. Only the engagement node, at the lower right, is replaced by the learned policy [Dantas 2025].

3.1. Environment and Setup

We built AsaGym, a library that wraps ASA as a Gymnasium environment [Towers et al. 2024], so standard RL implementations run without modification [Raffin et al. 2021]. As shown in Figure 2, only the engagement node of the fighter BT is redirected to an external process, which exchanges observations and actions over a ZeroMQ socket using Protobuf messages. The other subtrees, covering collision avoidance, threat reactions, return to base, combat air patrol, and formation flight [Costa et al. 2022], keep running unchanged.

The task is a two-versus-two BVR engagement between F-16 models equipped with radar, radar warning receiver, datalink, and six active missiles [Dantas et al. 2025a]. Two setups are evaluated, shown in Figure 3. In **Setup 1**, only the blue leader is controlled by a learned policy, while its wingman and both opponents follow expert-written engagement rules. In **Setup 2**, both blue aircraft share a single policy, each acting on its own state but receiving a shared observation of the whole scene. Episodes start head-on at 25,000 feet, with separation drawn uniformly from 20 to 60 nautical miles, and end when any aircraft is eliminated or a higher-priority BT node takes over. Altitude is held constant, so the engagement takes place in the horizontal plane.

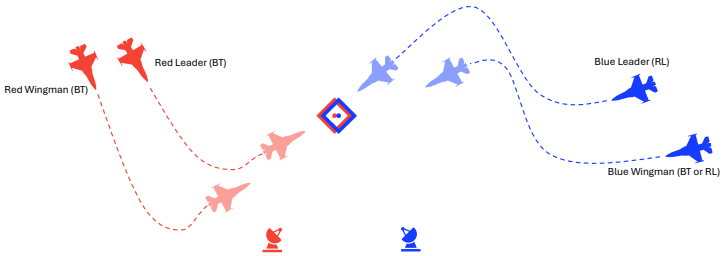


Figure 3. Two-versus-two engagement. In Setup 1 only the blue leader is controlled by a learned policy; in Setup 2 both blue aircraft share a single policy. Each team defends a common point of interest and is supported by ground radar [Dantas 2025].

3.2. Observation, Action, and Reward

The observation encodes the collective situational awareness of the engagement: for each of the four blue-red pairs and for the blue pair itself, it holds the magnitude difference of relative azimuths, the slant distance, and the airspeed difference, all normalized. This makes the policy invariant to where the engagement happens and to the absolute heading of the formation. The action is a continuous desired heading and airspeed, tracked by the underlying flight model, held for roughly five seconds by a frame-skipping wrapper so the agent commits to a maneuver instead of oscillating at the simulation rate.

The reward encodes what pilots call collective situational awareness. The *relative azimuth* term rewards the blue team for pointing at the red team while the red team points away, summed over all pairs:

$$\text{reward}_{ra} = \sum_{i=1}^{n_B} \sum_{j=1}^{n_R} \frac{|ra(R_j, B_i)| - |ra(B_i, R_j)|}{n_B \, n_R \, ra_{\max}}, \tag{1}$$

where n_B and n_R are the number of blue and red aircraft, B_i and R_j are the i-th blue and j-th red agent, $ra(A, B)$ is the relative azimuth of A with respect to B, and $ra_{\max} = 180°$. A *velocity* term repeats this angular advantage for the nearest opponent alone and scales it by the airspeed difference between the two aircraft, so an agent holding a favorable angle without the speed to exploit it earns less. It applies only when the angular advantage is positive, and is zero otherwise. The total reward is the sum of the two normalized terms.

The key design point is that the reward is summed over the team rather than computed per agent. In preliminary qualitative tests, not a controlled ablation, individualistic rewards such as minimizing the distance to an opponent or maximizing one's own angular advantage produced agents that pursued conflicting goals and left the formation uncovered, which motivated a team-level formulation that aligns each agent's objective with that of the formation.

3.3. Results

Four algorithms were compared using Stable-Baselines3 defaults [Raffin et al. 2021]: Proximal Policy Optimization (PPO) [Schulman et al. 2017], Advantage Actor-Critic (A2C) [Mnih et al. 2016], Soft Actor-Critic (SAC) [Haarnoja et al. 2018], and Twin Delayed Deep Deterministic Policy Gradient (TD3) [Fujimoto et al. 2018]. The only changes were the rollout parameter (`n_steps` for PPO and A2C, `train_freq` for SAC and TD3) and, where applicable, the batch size, both set to 128. Each was trained five times with different seeds for 500,000 steps across 16 parallel environments.

The top row of Figure 4 shows the reward curves for both setups. PPO converges highest in both, and it stabilizes even faster in Setup 2, the harder task, which suggests that sharing one policy across two agents helped rather than hurt. SAC learns stably but settles lower. TD3 rises early and then degrades, consistent with favorable random choices rather than a learned policy, and A2C makes little progress.

Episode length, in the bottom row, separates the algorithms in the same order, with PPO and SAC stabilizing at the shortest episodes and A2C staying high throughout. Length alone is ambiguous, since an episode ends when any aircraft is eliminated, but

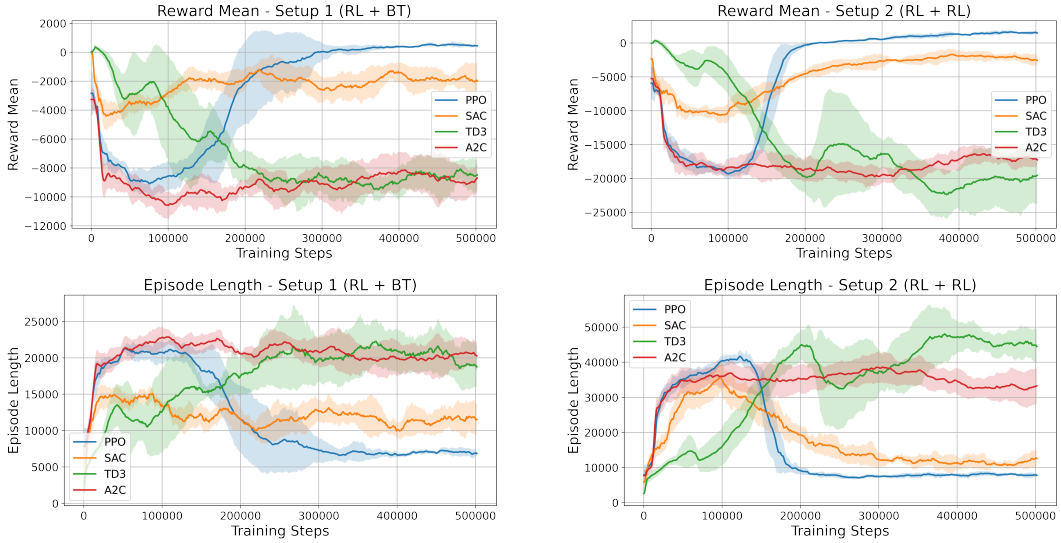


Figure 4. Average episode reward (top) and mean episode length (bottom) over five seeds, in Setup 1 (left) and Setup 2 (right). The solid line is the mean and the shaded band the standard deviation [Dantas 2025].

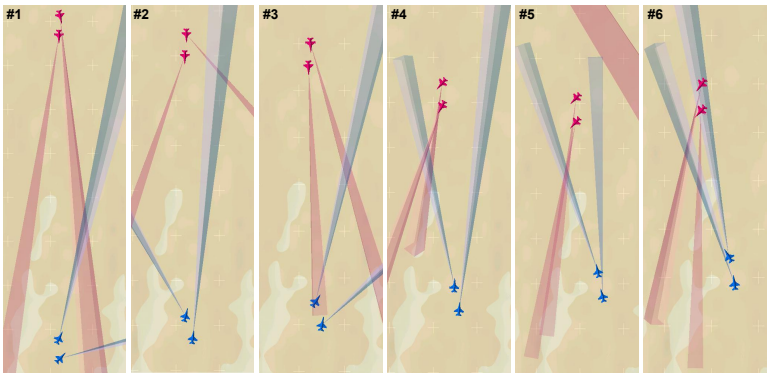


Figure 5. Six frames from a typical engagement in Setup 2, showing how the two policy-controlled aircraft maneuver together to gain positional advantage [Dantas 2025].

paired with high reward, the short episodes of PPO and SAC mark decisive engagements rather than early losses.

Table 1. Training time in hours, mean $\pm$ standard deviation over five seeds.

Setup	PPO	SAC	TD3	A2C
Setup 1 (policy + BT)	6.77 ± 0.57	5.72 ± 0.26	4.74 ± 0.24	4.10 ± 0.25
Setup 2 (shared policy)	6.30 ± 0.16	5.02 ± 0.31	3.45 ± 0.10	3.10 ± 0.14

PPO was the most expensive to train (Table 1) but produced the best policies; Setup 2 trained faster because two learning agents double the experience per update.

Brazilian Air Force BVR fighter pilots, acting as subject matter experts, reviewed recorded PPO episodes through visual inspection. Figure 5 shows six frames from a representative Setup 2 episode. The experts identified behaviors they recognized from real training: mutual support, coordinated maneuvers, and adaptive targeting. The agents were not told to cooperate and the behavior emerged from a reward defined over the team.

4. Learning Maneuvers from Human Pilots

Reward-driven learning works when the objective can be written down. Many maneuvers are instead defined by how an expert executes them, and for those, learning from demonstration fits better [Dantas 2024, Dantas et al. 2025c].

4.1. Task, Data, and Models

We study the pop-up attack, an air-to-ground maneuver in which an aircraft climbs from low altitude, rolls, and dives onto a ground target while minimizing exposure. Holding the planned climb and dive angles while correcting for wind is what makes it hard to reproduce autonomously. The dataset contains 30 recordings flown by a Brazilian Air Force pilot in a six-degree-of-freedom F-16 simulator, all from the same starting point and trimmed to 39 frames.

Two state vectors are compared. The *baseline* holds the seven quantities an instrument reports directly: altitude, pitch, roll, and yaw angles, radial angle, distance to target, and altitude difference. The *full* configuration adds nine derived ones, the linear velocities, angular velocities, and accelerations, which must be estimated or computed rather than measured directly. The action vector is the pilot's own input: pitch, roll, and throttle. The baseline matters because it is what the model sees whenever those derived quantities are missing or their estimates are unreliable.

Behavior cloning used a multilayer perceptron (MLP), a long short-term memory (LSTM) network [Hochreiter and Schmidhuber 1997], and a gated recurrent unit (GRU) network [Cho et al. 2014], each with 128, 64, and 32 hidden units and a three-unit linear output. The recurrent models use a five-step sliding window with stride one. The 30 recordings were split into 70% for training/validation and 30% for testing, with five-fold cross-validation at the recording level, so windows from one recording stay in the same fold. Training used Adam with a learning rate of 10^{-4}, batch size 64, and early stopping with a patience of 20 epochs. Final models were trained for the average number of epochs across folds, with results averaged over five seeds.

4.2. Results

Table 2 reports the coefficient of determination (R^2) and the root mean squared error (RMSE), computed on the denormalized control inputs and averaged across the three commands; since these have different scales, RMSE serves to compare models rather than as a physical error. Two results stand out. First, the recurrent models retain far more explained variance when derived variables are removed: in R^2 the MLP falls from 0.726 to 0.323, while the GRU falls only from 0.971 to 0.670. RMSE is similar across all three models in the baseline, so the gain lies in tracking the shape of the maneuver rather than in absolute error. Temporal structure substitutes for explicit velocity and acceleration inputs, which matters whenever a robot acts on incomplete state. Second, the GRU matches the LSTM in accuracy with a simpler cell, since its two gates replace the three of the LSTM; in the full configuration it was also slightly faster, although within one standard deviation, which makes it a natural candidate for onboard execution.

Figure 6 compares the predicted and actual commands. Roll is tracked most accurately, including during sharp transitions, while pitch slightly underestimates the amplitude in the most variable segments. Throttle follows the sustained phases well, with larger

Table 2. Behavior cloning results, mean ± standard deviation over five seeds. Inference time is the time to predict the entire test set.

Model	Configuration	R^2	RMSE	Inference (ms)
MLP	Baseline	0.323 ± 0.009	8.850 ± 0.015	267.52 ± 19.19
MLP	Full	0.726 ± 0.070	1.977 ± 0.054	$\mathbf{266.92 \pm 12.25}$
LSTM	Baseline	0.672 ± 0.003	8.708 ± 0.059	1432.22 ± 122.63
LSTM	Full	0.970 ± 0.001	1.743 ± 0.088	1283.86 ± 119.91
GRU	Baseline	0.670 ± 0.003	8.698 ± 0.055	1716.36 ± 304.40
GRU	Full	$\mathbf{0.971 \pm 0.002}$	$\mathbf{1.738 \pm 0.064}$	1206.61 ± 65.94

deviations occurring mainly during abrupt changes. Overall, prediction errors stay within the variability across the 30 pilot recordings, so the model captures the pilot's range of control behavior rather than any single flight.

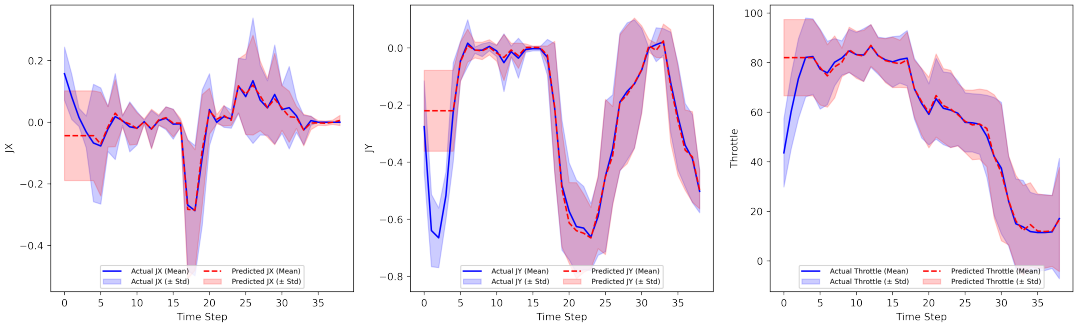


Figure 6. Actual versus predicted pilot commands for pitch, roll, and throttle during the pop-up attack, using the GRU in the full configuration. Shaded areas show the standard deviation across recorded maneuvers and model predictions [Dantas 2025].

4.3. Generative Augmentation

With only 30 demonstrations, a variational autoencoder [Kingma and Welling 2013] synthesized additional trajectories, smoothed with a moving average and reviewed by subject matter experts. Synthetic samples were added in amounts from 7 to 150.

The effect depended on how much real information the model already had, as Table 3 shows. In the baseline configuration the MLP gained most in R^2, rising to 0.455 at 60 samples although its RMSE rose slightly, while the recurrent models changed little because they already extract the structure the extra samples would supply. In the full configuration synthetic data made every model worse, monotonically. Where real information is not missing, augmentation only introduces distribution mismatch.

Table 3. Effect of synthetic data on R^2, mean ± standard deviation over five runs. Best mean per model and configuration in bold.

		Synthetic samples added					
Config.	**Model**	**0**	**7**	**15**	**30**	**60**	**150**
Baseline	MLP	0.323 ± 0.009	0.277 ± 0.069	0.312 ± 0.066	0.414 ± 0.074	$\mathbf{0.455 \pm 0.053}$	0.439 ± 0.043
	LSTM	0.672 ± 0.003	0.680 ± 0.003	0.679 ± 0.005	0.680 ± 0.004	0.680 ± 0.005	$\mathbf{0.681 \pm 0.004}$
	GRU	0.670 ± 0.003	$\mathbf{0.683 \pm 0.002}$	$\mathbf{0.683 \pm 0.005}$	0.681 ± 0.004	0.680 ± 0.007	0.681 ± 0.006
Full	MLP	$\mathbf{0.726 \pm 0.070}$	0.698 ± 0.085	0.616 ± 0.094	0.598 ± 0.078	0.472 ± 0.140	0.397 ± 0.213
	LSTM	$\mathbf{0.970 \pm 0.001}$	0.957 ± 0.002	0.951 ± 0.002	0.950 ± 0.003	0.944 ± 0.004	0.939 ± 0.003
	GRU	$\mathbf{0.971 \pm 0.002}$	0.958 ± 0.002	0.954 ± 0.004	0.948 ± 0.005	0.943 ± 0.005	0.937 ± 0.007

5. Discussion

Three lessons generalize beyond air combat. First, replacing one node of a behavior tree with a learned policy, rather than learning the whole controller, kept the system inspectable; this hybrid pattern applies to any robot that already has a working symbolic controller. Second, defining the reward over the team rather than the individual led to cooperative behavior without scripting it, relevant to multi-robot coordination wherever several vehicles share a task. Third, when demonstrations are few, temporal models tolerate missing state far better than feedforward ones, and synthetic data helps only when that state is genuinely incomplete. As for limitations, everything runs in high-fidelity simulation: AsaGym links the policy to ASA through a socket, which is not software- or hardware-in-the-loop testing, and behaviors were validated qualitatively by experts rather than against real flight data. The RL engagement is planar, so it does not yet support claims of three-dimensional autonomy, and the non-adaptive opponents risk overfitting the policy to a predictable adversary. Finally, comparisons are among algorithms and models within the framework; prior BVR agents run on different, mostly closed simulators, which prevented a direct quantitative comparison.

6. Conclusion

This paper summarized a doctoral thesis on simulation and machine learning for air combat. With a team-level reward, RL agents developed cooperative tactics that fighter pilots recognized from their own training; with 30 expert flights, temporal models reproduced a demanding maneuver even with incomplete state; and the study of generative augmentation clarified under which conditions synthetic data improves or degrades performance.

Future work on the RL side includes three-dimensional engagements, decentralized policies for larger formations, self-play against adaptive opponents, and a probability-of-kill term in the reward; on the imitation side, flights from more pilots and maneuvers, adversarial imitation, transformer-based models, and validation against real flight data. On both fronts, software- and hardware-in-the-loop testing is the next step toward deployment.

Acknowledgments

This research was partially funded by the Funding Authority for Studies and Projects (FINEP), Finance Code 01.20.0196.00/2824/20.

Source Code

Code is publicly available: AsaGym at `https://github.com/ASA-Simulation/asa-gym` and the imitation and generative learning models at `https://github.com/jpadantas/pop-up_attack_generative`. ASA and the recorded flight data are restricted due to their defense applications.

References

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Colledanchise, M. and Ögren, P. (2020). *Behavior Trees in Robotics and AI: An Introduction*. Chapman & Hall / CRC Artificial Intelligence and Robotics Series. CRC Press, Boca Raton, FL.

Costa, A. N., Dantas, J. P. A., Scukins, E., Medeiros, F. L. L., and Ögren, P. (2025). Simulation and Machine Learning in Beyond Visual Range Air Combat: A Survey. *IEEE Access*, 13:76755–76774.

Costa, A. N., Medeiros, F. L. L., Dantas, J. P. A., Geraldo, D., and Soma, N. Y. (2022). Formation control method based on artificial potential fields for aircraft flight simulation. *Simulation: Transactions of the Society for Modeling and Simulation International*, 98(7):575–595.

Dantas, J. P. A. (2024). Autonomous Pop-Up Attack Maneuver Using Imitation Learning. In *Proceedings of the Winter Simulation Conference, PhD Colloquium*, Orlando, FL, USA. IEEE.

Dantas, J. P. A. (2025). *Simulation and Machine Learning for Decision Support and Autonomy in Air Combat Operations*. PhD thesis, Instituto Tecnológico de Aeronáutica.

Dantas, J. P. A., Costa, A. N., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2021a). Engagement Decision Support for Beyond Visual Range Air Combat. In *Proceedings of the 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, pages 96–101, Natal, RN, Brazil. IEEE.

Dantas, J. P. A., Costa, A. N., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2021b). Weapon Engagement Zone Maximum Launch Range Estimation Using a Deep Neural Network. In *Intelligent Systems: 10th Brazilian Conference, BRACIS 2021*, pages 193–207, Cham. Springer.

Dantas, J. P. A., Costa, A. N., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2026). PoKER: a probability of kill estimation rate model for air-to-air missiles using machine learning on stochastic targets. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 23(4):579–599.

Dantas, J. P. A., Costa, A. N., Gomes, V. C. F., Kuroswiski, A. R., Medeiros, F. L. L., Geraldo, D., Samersla, A. R., Silva, S. R., Caetano, A. F. M., Ferreira, Y. D., Arantes, A. O., Santos, D. S., and Aquino, M. R. C. (2025a). ASA: A Simulation Environment for Evaluating Military Operational Scenarios. In *Computational Science and Computational Intelligence: 11th International Conference, CSCI 2024*, Communications in Computer and Information Science, pages 203–215, Cham, Switzerland. Springer Nature Switzerland.

Dantas, J. P. A., Costa, A. N., Medeiros, F. L. L., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2022a). Supervised Machine Learning for Effective Missile Launch Based on Beyond Visual Range Air Combat Simulations. In *Proceedings of the 2022 Winter Simulation Conference (WSC)*, pages 1990–2001, Singapore. IEEE.

Dantas, J. P. A., Geraldo, D., Costa, A. N., Maximo, M. R. O. A., and Yoneyama, T. (2025b). ASA-SimaaS: Advancing Digital Transformation through Simulation Services in the Brazilian Air Force. *Spectrum: The Journal of Operational Applications in Defense Areas*, 26(1):1–7.

Dantas, J. P. A., Geraldo, D., Medeiros, F. L. L., Maximo, M. R. O. A., and Yoneyama, T. (2023a). Real-Time Surface-to-Air Missile Engagement Zone Prediction Using Simulation and Machine Learning. In *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, USA. National Training and Simulation Association (NTSA).

Dantas, J. P. A., Maximo, M. R. O. A., Costa, A. N., Geraldo, D., and Yoneyama, T. (2022b). Machine Learning to Improve Situational Awareness in Beyond Visual Range Air Combat. *IEEE Latin America Transactions*, 20(8):2039–2045.

Dantas, J. P. A., Maximo, M. R. O. A., and Yoneyama, T. (2023b). Autonomous Agent for Beyond Visual Range Air Combat: A Deep Reinforcement Learning Approach. In *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '23, pages 48–49. Association for Computing Machinery.

Dantas, J. P. A., Maximo, M. R. O. A., and Yoneyama, T. (2024a). Loyal Wingman Assessment: Social Navigation for Human-Autonomous Collaboration in Simulated Air Combat. In *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '24, pages 61–62. Association for Computing Machinery.

Dantas, J. P. A., Maximo, M. R. O. A., and Yoneyama, T. (2025c). Autonomous Aircraft Tactical Pop-Up Attack Using Imitation and Generative Learning. *IEEE Access*, 13:81204–81217.

Dantas, J. P. A., Medeiros, F. L. L., Samersla, A. R., Botelho, P. L. R., Gomes, V. C. F., Silva, S. R., Ferreira, Y. D., Arantes, A. O., Aquino, M. R. C., and Maximo, M. R. O. A. (2025d). Deep Reinforcement Learning Agents With Collective Situational Awareness for Beyond Visual Range Air Combat. *IEEE Access*, 13:143052–143069.

Dantas, J. P. A., Silva, S. R., Gomes, V. C. F., Costa, A. N., Samersla, A. R., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2024b). AsaPy: A Python Library for Aerospace Simulation Analysis. In *Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '24, pages 15–24. Association for Computing Machinery.

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596, Stockholm, Sweden. PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870, Stockholm, Sweden. PMLR.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hu, D., Yang, R., Zuo, J., Zhang, Z., Wu, J., and Wang, Y. (2021). Application of Deep Reinforcement Learning in Maneuver Planning of Beyond-Visual-Range Air Combat. *IEEE Access*, 9:32282–32297.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

Liu, X., Yin, Y., Su, Y., and Ming, R. (2022). A Multi-UCAV Cooperative Decision-Making Method Based on an MAPPO Algorithm for Beyond-Visual-Range Air Combat. *Aerospace*, 9(10):563.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, NY, USA. PMLR.

Navarro, I., Patrikar, J., Dantas, J. P. A., Baijal, R., Higgins, I., Scherer, S., and Oh, J. (2024). SoRTS: Learned Tree Search for Long Horizon Social Robot Navigation. *IEEE Robotics and Automation Letters*, 9(4):3759–3766.

Patrikar, J., Dantas, J. P. A., Ghosh, S., Kapoor, P., Higgins, I., Aloor, J. J., Navarro, I., Sun, J., Stoler, B., Hamidi, M., Baijal, R., Moon, B., Oh, J., and Scherer, S. (2022). Challenges in Close-Proximity Safe and Seamless Operation of Manned and Unmanned Aircraft in Shared Airspace. In *2022 IEEE International Conference on Robotics and Automation (ICRA), Aerial Robotics Workshop*, Philadelphia, PA, USA.

Patrikar, J., Dantas, J. P. A., Moon, B., Hamidi, M., Ghosh, S., Keetha, N., Higgins, I., Chandak, A., Yoneyama, T., and Scherer, S. (2025). Image, speech, and ADS-B trajectory datasets for terminal airspace operations. *Scientific Data*, 12(1):468.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Sandström, V., Luotsinen, L., and Oskarsson, D. (2022). Fighter Pilot Behavior Cloning. In *Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 686–695, Dubrovnik, Croatia. IEEE.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silva, S. R., Gomes, V. C. F., Arantes, A. O., Caetano, A. F. M., Costa, V. L. D. B., Samersla, A. R., Medeiros, F. L. L., Ferreira, Y. D., Aquino, M. R. C., and Dantas, J. P. A. (2025). AsaFG: A Human-in-the-Loop Integration Module for Air Combat Simulations. *IEEE Access*, 13:155821–155834.

Towers, M., Kwiatkowski, A., Terry, J., et al. (2024). Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032*.

Zhang, H., Wei, Y., Zhou, H., and Huang, C. (2022). Maneuver Decision-Making for Autonomous Air Combat Based on FRE-PPO. *Applied Sciences*, 12(20):10230.

Zhou, Y., Yang, F., Zhang, C., Li, S., and Wang, Y. (2024). Cooperative decision-making algorithm with efficient convergence for UCAV formation in beyond-visual-range air combat based on multi-agent reinforcement learning. *Chinese Journal of Aeronautics*, 37(8):311–328.