A Survey of Reinforcement Learning for Autonomous Air Combat: Current Progresses and Limitations

Alex Pierron
SAMOVAR, Télécom SudParis,
Institut Polytechnique de Paris
91120 Palaiseau, France
alex.pierron@telecom-sudparis.eu

Thibault Lahire
Dassault Aviation
92210 Saint-Cloud, France
thibault.lahire@dassault-aviation.com

ABSTRACT

Autonomous air combat represents one of the most demanding challenges in artificial intelligence, requiring agents to operate under uncertainty, partial observability, and adversarial dynamics. Reinforcement Learning and Multi-Agent Reinforcement Learning have recently emerged as promising approaches for enabling adaptive decision-making and coordination in this domain. This survey provides a structured overview of Reinforcement Learning-powered autonomous air combat, with emphasis on open-source environments, algorithmic frameworks, and hierarchical control architectures. We systematically compare three aspects: (1) single-agent and multi-agent settings, (2) full-control and hierarchical abstractions, and (3) the treatment of sensors and observability. Furthermore, we analyze the reproducibility of recent contributions, highlighting the tension between fidelity and accessibility across open and closed-source platforms. Beyond a methodological review, we identify persistent challenges related to scalability, transfer to real-world platforms, non symmetrical scenarios robustness, and computational requirements. By consolidating these advances and limitations, this survey aims to clarify the current state of the field, highlight open problems, and outline pathways toward more robust, scalable, and operationally relevant autonomous collaboration in future air combat systems.

KEYWORDS

Deep Learning, Reinforcement Learning, Multi-Agent Reinforcement Learning, Survey, Autonomous Air Combat, Artificial Intelligence

ACM Reference Format:
Alex Pierron and Thibault Lahire. 2026. A Survey of Reinforcement Learning for Autonomous Air Combat: Current Progresses and Limitations. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/GEMF4800

1 INTRODUCTION

Air combat, defined as air-to-air engagements between fighter jets or drones of similar scale, has undergone several paradigm shifts over the past decades. Early encounters were fought Within Visual Range (WVR), relying on close-range manoeuvring, pilot expertise, and rapid decision-making under uncertainty. At this early stage, studies already recognized the inherently the inherently adversarial and dynamic nature of aerial engagements. Isaacs' foundational work on pursuit games [35] formalized such interactions as differential games, inspiring decades of subsequent research. In the 1970s, the NASA *Adaptive Manoeuvring Logic* program introduced one of the first computer-based approaches to simulate one-to-one engagements, marking a step toward algorithmic decision-making in manoeuvring flight [10–12]. These efforts underscored both the promise and the difficulty of encoding tactical reasoning into autonomous systems.

The following decades saw the introduction of advanced radar, Beyond Visual Range (BVR) missiles, and network-centric warfare, which progressively moved engagements farther apart and made information dominance as critical as kinematic performance. In parallel, computational approaches to tactical guidance expanded: game-theoretic formulations tailored to air-to-air combat [3] and high-fidelity batch simulation environments [24] enabled more systematic exploration of guidance strategies. Progressively, the focus shifted toward learning-based methods. Approximate dynamic programming was applied to derive viable combat strategies under uncertainty [46], followed by self-organizing neural networks for manoeuvre learning [71].

Today, the battlespace is characterized by distributed sensors, secure datalinks, and electronic warfare capabilities, enabling near-instantaneous sharing of targeting and situational awareness across platforms. In this high dimensional space, Reinforcement Learning (RL) and its multi-agent extension, Multi-Agent Reinforcement Learning (MARL), have emerged as key potential enablers for next-generation autonomous combat systems. RL enables a single agent to learn decision-making policies directly from interaction with simulated environments, while MARL extends this capability to multiple agents, potentially operating under adversarial conditions. These techniques now underpin several high-profile demonstrations of autonomous aerial combat. In DARPA's *AlphaDogfight Trials*, an RL-trained agent decisively outperformed a U.S. Air Force Weapons Instructor Course graduate in simulated WVR engagements [19]. The U.S. Air Force's *Collaborative Combat Aircraft* program [1] aims to deploy over a thousand autonomous drones, leveraging cooperative decision-making principles closely related to MARL, to operate alongside crewed fighters.

In parallel, the academic community has demonstrated a rapidly growing interest in this topic. As shown in Figure 1, the number of Google Scholar entries with titles containing both "Reinforcement Learning" and "Air Combat" (retrieved using the query *intitle:air combat intitle:reinforcement learning*) has risen sharply since

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/GEMF4800

2017, highlighting the accelerated pace of research in this area. A growing body of open-source works has emerged, ranging from high-fidelity flight simulators integrated with RL toolkits [53] to hierarchical MARL architectures capable of decomposing complex mission profiles into manageable sub-tasks [62]. However, the literature remains fragmented, and the integration of MARL research into realistic, autonomous air combat scenarios are far from solved. While Wang et al. [77] provided an early survey on RL for air combat in 2023, the field has evolved rapidly since then, with significant advances in MARL frameworks, simulation platforms, and training methodologies. Other existing surveys [17, 25, 65] adopt a broad machine learning perspective that encompasses supervised, evolutionary, and rule-based approaches in addition to RL, which limits the level of detail devoted to it and to MARL approaches. In this context, our survey focuses exclusively on reinforcement learning–based methods, providing an up-to-date and systematic analysis of the challenges and design choices involved in applying RL and MARL to autonomous air combat.

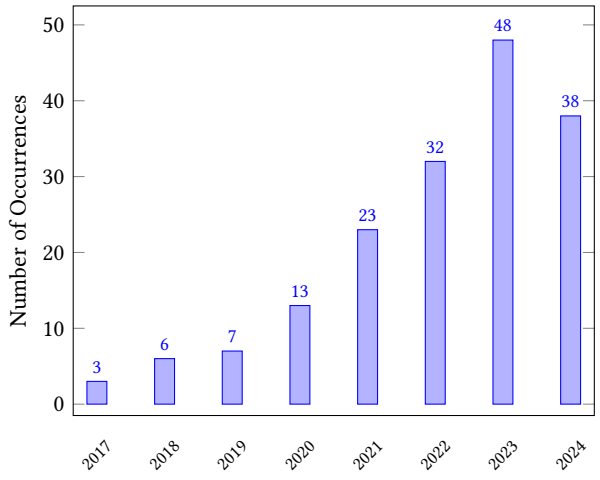

Figure 1: Temporal Evolution of Google Scholar Occurrences matching the query "*intitle:air combat intitle:reinforcement learning*" (2017–2024)

Scope and Contributions. This paper surveys the latest and most promising research in RL and MARL for autonomous collaborative air combat. Inevitably, given the breadth of the field, this survey cannot claim to be perfectly exhaustive. Instead, we focus on representative and influential contributions that illustrate the main lines of progress and highlight open challenges. The paper is organized as follows:

- **Section 2** establishes the theoretical and practical foundations of RL and MARL, including key algorithms used in the surveyed works at the intersection of RL and autonomous air combat.
- **Section 3** reviews existing approaches in autonomous aerial combat, analysing their core ideas and categorizing them notably through Table 1.
- **Section 4** focuses on the limitations and bottlenecks reported in the surveyed articles.

By consolidating recent open-source advances and highlighting open challenges, this survey aims to identify ongoing research efforts and highlight how the development of autonomous capabilities can steer future air combat systems.

2 BACKGROUND

This section reviews the theoretical foundations of reinforcement learning, its extension to multi-agent systems, and hierarchical formulations, in order to establish the methodological basis for the subsequent survey of autonomous air combat approaches. Note that the background work presented here has been carefully selected as it is reused in at least one of the cited articles of Section 3.

2.1 Reinforcement Learning

The foundations of Reinforcement Learning (RL) are closely connected to Control Theory and Dynamic Programming. Early seminal contributions include Bellman's formulation of dynamic programming [7], which introduced the principle of optimality, and Kalman's development of optimal linear quadratic regulators [37]. These methods laid the groundwork for modern control approaches [9, 41]. Reinforcement Learning extends these principles to settings where the agent lacks an explicit model of the environment dynamics and must learn directly from interaction [68]. In this work, we restrict our consideration to model-free reinforcement learning, where policies are optimized solely from sampled trajectories.

The agent–environment interaction is typically formalized as a Markov Decision Process (MDP) [54]. At each discrete time step t, the system occupies a state s_t. Upon applying action a_t, the agent transitions to state s_{t+1}, while receiving reward r_t. The agent's objective is to maximize the expected discounted return, expressed through the state-action value function:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a, a_t \sim \pi(s_t)\right],$$

where π denotes the policy mapping states to distributions over actions and $\gamma \in [0,1]$ is the discount factor.

A central distinction among reinforcement learning methods lies in whether they are *on-policy* or *off-policy*. On-policy algorithms improve a policy by learning directly from the data generated by that same policy, while off-policy algorithms can learn from data collected by a different behaviour policy, allowing them to reuse past experience. This distinction is closely tied to the design of experience replay [43] mechanisms and the efficiency of exploration. Another important axis differentiates *value-based* and *policy-based* methods. Value-based algorithms estimate value functions from which a policy is derived implicitly, while policy-based methods directly parameterize and optimize the policy itself. Actor–critic algorithms combine both paradigms by learning a policy (the actor) alongside a value function (the critic) to stabilize training. The advent of Deep Reinforcement Learning (DRL) [76] overcame the limitations of early model-free methods [57, 78] regarding large or continuous state and action spaces by employing deep neural networks [23, 56] to approximate value functions and policies, thereby enabling reinforcement learning to tackle high-dimensional and complex environments.

In this work, we focus on four influential DRL algorithms: Deep Q-Networks (DQN), Deep Deterministic Policy Gradient (DDPG), Soft-Actor Critic (SAC), and Proximal Policy Optimization (PPO), as they are the ones used in the research articles of this survey. DQN [47] extends Q-learning by approximating the Q-function with a deep neural network. It is an *off-policy, value-based* method tailored to discrete action spaces. Its main contribution is the stabilization of training via the use of a target network and experience replay [43]. This breakthrough allowed RL agents to achieve human-level control in high-dimensional tasks such as Atari [6].

DDPG [42] adapts reinforcement learning to continuous action spaces by combining deterministic policy gradients with an actor–critic framework. It is an *off-policy, policy-based* algorithm, where the actor network outputs deterministic actions and the critic estimates action values. DDPG leverages experience replay to improve sample efficiency, making it suitable for continuous control domains. SAC [27] improves upon this framework by augmenting the optimization objective with an entropy maximization term. By explicitly encouraging stochastic policies, SAC achieves a balance between exploration and exploitation, leading to improved robustness and sample efficiency. SAC is an *off-policy, policy-based* algorithm designed for continuous control tasks and is widely regarded as one of the most effective algorithms in this setting.

Proximal Policy Optimization (PPO) [60], by contrast, represents a family of *on-policy, policy-based* methods. It optimizes policies using a clipped surrogate objective that prevents excessively large updates, thereby improving training stability without the complexity of trust-region approaches such as Trust Region Policy Optimization (TRPO) [59]. PPO is applicable to both discrete and continuous action spaces, and has since become a standard baseline for reinforcement learning research due to its simplicity and empirical reliability.

Beyond the algorithms themselves, several techniques have been introduced to accelerate learning and improve policy quality. Prioritized Experience Replay (PER) [58] enhances the back-propagation [56] by sampling transitions with higher temporal-difference errors more frequently, ensuring that updates are focused on the most informative experiences. Another key mechanism is Self Play (SP) [64]: by continuously training against copies of itself, an agent generates an evolving curriculum of increasingly challenging interactions, fostering the development of complex strategies without requiring handcrafted opponents. Additionally, Curriculum Learning (CL) [8] structures the learning process by starting with simpler tasks and gradually introducing more complex ones, which can significantly enhance learning efficiency and policy quality.

2.2 Multi-Agent Reinforcement Learning

Even if RL provides the foundations for sequential decision-making in single-agent settings, many domains of interest involve multiple agents interacting within the same environment. Multi-Agent Reinforcement Learning (MARL) [2, 13, 26, 29, 84] extends the principles of RL to such settings, where agents may pursue cooperative, competitive, or mixed objectives. The presence of multiple learners adds several challenges. The environment becomes non-stationary because its dynamics depend not only on state transitions but also on the changing policies of other agents. Partial observability is more pronounced, as each agent has access only to local informations. Additionally, credit assignment becomes difficult, since it is often hard to determine how individual agents contribute to team-level outcomes [48, 85].

Formally, MARL is modelled as a Partially Observable Markov Decision Process (POMDP) expressed as:

$$\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^{N}, \mathcal{P}, \{\Omega_i\}_{i=1}^{N}, \mathcal{O}, \{R_i\}_{i=1}^{N}, \gamma \rangle,$$

where $\mathcal{N}$ is the set of agents, $\mathcal{S}$ is the global state space, and $\mathcal{A}_i$ is the action space available to agent i with $a_{t,i}$ the action taken at time step t by agent i. The state transitions are governed by the state transition kernel $\mathcal{P}$, while each agent i receives a private observation $o_i \in \Omega_i$ through the observation function $\mathcal{O}$ that can accept task dependent combination of arguments: observations from one or several agents, previous states... The goal of each agent is to maximize the expected discounted return defined by the possibly equal reward functions $\{R_i\}_{i=1}^{N}$, under the discount factor γ. This generalizes the single-agent MDP formulation by incorporating multiple decision-makers, each of whom must adapt to the simultaneous presence of others.

A first class of approaches, known as *independent learning*, extends single-agent algorithms directly to the multi-agent setting. Each agent learns its own policy independently, treating the other agents as part of the environment. Independent Q Learning (IQL) [70] is the most canonical example, adapting tabular Q-learning to multi-agent contexts. More recent adaptations include Independent Proximal Policy Optimization (IPPO) [82] and Independent Soft-Actor Critic (ISAC), which leverage deep actor–critic structures to scale to continuous and high-dimensional settings. Independent methods are attractive for their simplicity and scalability, as they avoid the complexity of joint state–action representations. However, because the learning signal of each agent evolves as others change their policies, these methods can suffer from instability in tightly coupled environments [21, 30].

To address these limitations, a dominant paradigm has emerged under the label of Centralized Training with Decentralized Execution (CTDE). In CTDE, agents are trained using privileged information such as global states or joint actions, but policies are executed using only local observations. This paradigm strikes a balance between tractability and realism, as it stabilizes training without sacrificing decentralized deployment. Within CTDE, two families of algorithms have proven particularly influential. The first is value decomposition, where the joint action-value function is factorized into local components. Value Decomposition Network (VDN) [67] implement a simple linear decomposition, while QMIX [55] introduces a nonlinear mixing network that enforces a monotonicity constraint, ensuring that maximizing local action-values also maximizes the global value. These methods have become standard in cooperative tasks with discrete action spaces, where the decomposition structure increases both interpretability and stability.

The second influential family within CTDE is the class of actor–critic algorithms, which learn explicit policies alongside centralized critics. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [45] extends DDPG to multi-agent settings by equipping each agent with a decentralized deterministic actor policy, while critics are conditioned on the full joint state and joint actions. This enables learning in continuous action spaces and in mixed

cooperative–competitive environments. Multi-Agent Proximal Policy Optimization (MAPPO) [82] adapts the PPO framework to the CTDE setting, maintaining decentralized policies while leveraging a centralized critic during training. By combining PPO's clipped surrogate objective with CTDE, MAPPO achieves robustness and stable performance across a wide variety of cooperative and mixed-agent domains, and has emerged as one of the most reliable and widely used MARL baselines.

2.3 Hierarchical Reinforcement Learning

Standard Reinforcement Learning methods operate at a flat level of abstraction, directly mapping states or observations to primitive actions. However, many sequential decision-making problems require reasoning across multiple temporal scales. Hierarchical Reinforcement Learning (HRL) [5, 49, 73] extends standard RL by introducing temporal abstraction, thereby enabling agents to plan over extended horizons by decomposing behaviour into sub-policies or *options*. In the options framework [69], each option is characterized by an initiation set, an intra-option policy, and a termination condition, while a high-level policy determines which option to execute at each stage. This architecture facilitates the reuse of sub-policies, improves exploration, and enhances sample efficiency.

The general principles of HRL naturally extend to multi-agent systems, giving rise to Hierarchical Multi-Agent Reinforcement Learning (HMARL). In HMARL, multi-agent behaviour is structured across multiple levels of abstraction: high-level policies allocate tasks, roles, or strategies among agents, while low-level controllers govern fine-grained execution such as manoeuvring or coordination with nearby teammates. This hierarchical decomposition is particularly beneficial in multi-agent contexts, as it enables scalable coordination across both spatial and temporal dimensions. Figure 2 illustrates the difference between standard RL and HRL. At the high level, CTDE algorithms such as QMIX [55] or MAPPO [82] can be employed to coordinate team-level strategies, while low-level execution can leverage continuous-control algorithms such as SAC or discrete policies as in DQN, with multi-agent variants applied as needed. With high-level controllers governing long-term strategic decisions and low-level sub-policies managing short-term execution, agents can potentially operate more effectively across different timescales. As such, HRL and HMARL represent a natural extensions of reinforcement learning principles to the complex, hierarchical nature of real-world environments.

3 EXISTING APPROACHES FOR REINFORCEMENT LEARNING POWERED AUTONOMOUS AIR COMBAT

RL, MARL and HRL have emerged as particularly suitable frameworks for autonomous air combat as they directly address sequential decision-making under uncertainty and adversarial dynamics. In contrast, classical control-theoretic methods such as optimal control or dynamic programming rely on explicit system models and often become intractable in high-dimensional, non-linear, or partially observable environments. Building on the methodological advantages of RL in general, this section surveys existing approaches to RL in autonomous air combat. We structure the discussion along four axes that capture key design choices. First, we distinguish

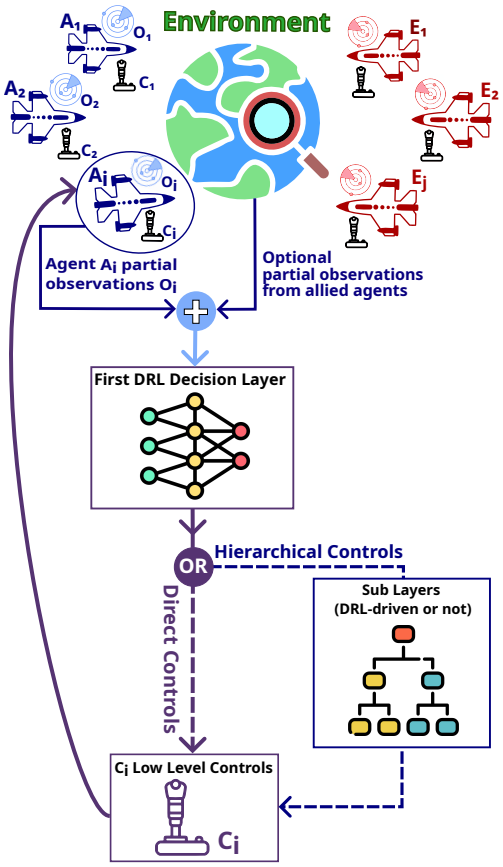

Figure 2: Control architectures for an agent A_i in a possibly multi-agent environment. The top decision layer either directly outputs low-level control commands or selects high-level actions that are executed by lower-level controllers, which generate the concrete controls required to operate the aircraft based on partial observations.

between single-agent and multi-agent formulations (3.1). Second, we contrast full-control with hierarchical control frameworks (3.2). Third, we examine how sensors and observability are modelled (3.3). Finally, we discuss reproducibility by distinguishing between open-source and closed-source environments (3.4). For each axis, we highlight representative studies that exemplify the underlying trade-offs while situating other relevant contributions in the same category. This organization emphasizes conceptual progress and clarifies how modelling assumptions shape algorithmic design, scalability, and reproducibility in autonomous air combat research.

3.1 Single-Agent and Multi-Agent Settings

The single-agent setting has historically served as the foundational case in air combat research, providing the simplest environment possible for a confrontation scenario. These studies often focus on direct manoeuvring or tactical reasoning for a single aircraft and are typically the starting point before extending to coordinated multi-agent systems. In contrast, multi-agent approaches incorporate

coordination, communication, and adversarial reasoning, capturing the collective nature of real-world aerial engagements.

A paradigmatic single-agent contribution is the hierarchical RL system employed by the Lockheed Martin team, which placed second behind Heron Systems (now Shield AI [63]), during the DARPA's AlphaDogfight Trials (ADT) challenge [52, 53] in which agents competed in symmetric 1v1 WVR engagements limited to cannons. The airframe used is a F-16 [22] for both side and the environment relies on the flight dynamic model JSBSim [36] as the Flight Dynamics Model (FDM) to model a realistic flight physic. Their framework employed a set of SAC-trained sub-policies, each corresponding to a tactical mode: *aggressive*, *conservative*, or *control zone*. A high-level selector, also trained with SAC, determined the active sub-policy at each timestep. The neural network used is a Multi Layer Perceptron (MLP) [32]. Training efficiency was enhanced by Curriculum Learning, progressing from scripted opponents to Self-Play (SP), while PER [58] and dense reward shaping addressed sparsity in the reward signal. The resulting agent demonstrated operational credibility by decisively defeating a USAF Weapons Instructor 5–0. This study underscores how even in single-agent contexts, hierarchical structuring and staged training can yield competitive and interpretable behaviour.

At the other end of the spectrum, Kong et al. [38] propose a fully multi-agent HMARL framework for WVR team combat using JSBSim [36]. Their system accommodates variable team sizes and symmetric matchups, combining SAC-trained manoeuvring sub-strategies (*offensive*, *defensive*, and *aggressive shooting*) with a cooperative high-level controller trained via a Multi-Head Attention (MHA) [72] augmented QMIX. The architecture explicitly incorporates intra-team communication and limited radar sensing, enabling decentralized decision-making under uncertainty. Training was staged: manoeuvring sub-strategies were first trained independently with dual Gated Recurrent Units (GRUs) [15] for temporal reasoning, before integration into the cooperative controller. This approach highlights the additional algorithmic and architectural complexity required when scaling from a single agent to coordinated multi-agent teams.

Other single-agent works provide alternative baselines. Bae et al. [4] adopt a POMDP formulation where a forward-facing radar and distance-dependent noise model observations, and employ SAC with either MLPs or Long Short-Term Memorys (LSTMs) [31]. In the BVR domain, Piao et al. [50] exploit the WUKONG simulator [50] to model radar-guided missile combat with macro-level actions, while Zhang et al. [83] design a tailored PPO variant for missile duels. On the multi-agent side, the Light Aircraft Game (LAG) environment [44] extends its single-agent baseline to symmetric 2v2 WVR engagements using MAPPO, while Graph Neural Network (GNN) [79] based approaches such as Han et al. [28], Huo et al. [34], leverage graph representations to capture relational structure between team members and opponents and explore the possible air manoeuvring tactics. Taken together, these contributions illustrate the field's progression from isolated single-agent experiments toward scalable multi-agent coordination.

3.2 Full-Control and Hierarchical Control

Another methodological split concerns the granularity of control as presented in Figure 2. Full-control approaches require agents to directly generate actuator-level commands (e.g., aileron, rudder, elevator, throttle), tightly coupling policy learning with flight dynamics. Hierarchical approaches, by contrast, separate low-level manoeuvring from higher-level tactical reasoning, often by reusing pre-trained controllers or rule-based modules to stabilize training and promote scalability.

A paradigmatic full-control contribution is De Marco et al. [18], which presents a deep reinforcement learning control system for high-performance aircraft also using JSBSim [36] and F-16 [22] airframe. Their study employs a DDPG controller to navigate randomly generated waypoint sequences at varying altitudes and Mach numbers. The trained agent is capable of executing tightly coupled manoeuvres, including rapid turn sequences, while maintaining robustness to atmospheric disturbances, noisy sensors, and different initial conditions. Other representative full-control works includes Bae et al. [4], which train agents with SAC to directly output actuator-level commands in JSBSim under partial observability, relying on LSTMs to integrate noisy radar observations. These studies highlight the realism gained from low-level control, but also its training instabilities and sensitivity to observation fidelity.

In contrast, the LAG environment [44] illustrates hierarchical control. Its architecture is explicitly two-layered: a low-level controller trained with PPO for basic manoeuvres, and a high-level tactical layer issuing macro-actions such as heading changes and missile firing. This separation stabilizes learning and enables direct reuse in multi-agent contexts. Other works exploit hierarchical coordination at different levels. The Lockheed Martin system [53] blurs boundaries between levels by combining actuator-level subpolicies with high-level selectors, while Kong et al. [38] employ hierarchical multi-agent controllers with SAC-trained manoeuvring strategies and a modified QMIX high-level policy using MHA to deal with fluctuating number of agents during an episode.

A distinct hierarchical paradigm is introduced in Piao et al. [51], who propose the ENACTIVE algorithm for BVR 1v1 combat in the WUKONG simulator [50]. Unlike fixed decision-interval methods, their ENHANCE module adaptively adjusts decision timing via a neuroscience-inspired excitatory/inhibitory balance mechanism, learning not only *what* actions to execute but also *when* to act. Their FACTIVE module factorizes manoeuvres into offensive/defensive and energy semantics, optimizing them jointly to expand the tactic space. Macro-actions such as heading, vertical manoeuvres, speed, and g-force are transmitted to a dedicated controller, combining high-level DRL tactical reasoning with reliable low-level execution. This yields both tactical diversity and interpretability, claiming expanding tactic coverage compared to classic Basic Fighter Manoeuvress (BFMs) [51]. Beyond performance, this study underscores the importance of explicitly modelling the temporal structure of decision-making in air combat, an aspect often neglected in hierarchical approaches.

More recent work by Kuroswiski et al. [40] introduces Knowledge-informed Tasks for Attention-driven Behaviour (KTAB), a task-based decision layer that bridges perception and actuation.Candidate manoeuvres are generated by a rule-based module and evaluated

Table 1: Summary of the surveyed works

Study	WVR or BVR	Mission Task	Single/Multi Agent	Main Algorithm & Technics	High-Level / Low-Level actions	Physics Simulator	Radar Usage	Open Source Code
Yang et al. [80] (2019)	WVR	Combat (Canons)	Single Agent	DQN + SP	High-Level	Custom	None	No
Piao et al. [50] (2020)	BVR	Combat (Missiles)	Single Agent	PPO with advanced reward shaping	High-Level	WUKONG (Custom)	- RWR - Lock-on and missile guidance	No
Sun et al. [66] (2021)	BVR	Combat (Missiles)	Multi Agent	Multi-Agent Hierarchical Policy Gradient algorithm (MAHPG)	Both (Hierarchical)	WUKONG (Custom)	- RWR - Lock-on and missile guidance	No
Liu et al. [44] (2022)	WVR	- Navigation - Evading Missiles - Combat (Canon + Missiles)	Both	PPO / MAPPO	Both (Hierarchical)	JSBSim	None	Yes
Han et al. [28] (2022)	BVR	Combat (Missiles)	Multi Agent	Deep Relationship Graph Reinforcement Learning (DRGRL)	High-Level	WUKONG (Custom)	- RWR - Lock-on and missile guidance	No
Pope et al. [53] (2022)	WVR	Combat (Canon)	Single Agent	SAC + CL + PER + SP	Both (Hierarchical)	JSBSim	None	No
Zhang et al. [83] (2022)	BVR	Combat (Missile)	Single Agent	Final Reward Estimation PPO + SP	Both	Custom	None	No
Yoo et al. [81] (2022)	WVR	Target Interception (Cannon)	Single Agent	SAC / PPO	Low-Level	DCS	Radar Usage for the target aircraft	No
Kong et al. [38] (2023)	WVR	Combat (Canon)	Both	- SAC with GRUs - attention based QMIX + CL + SP	Both (Hierarchical)	JSBSim	Partial Observations (Simplified)	No
Chai et al. [14] (2023)	WVR	Combat (Missiles)	Single Agent	PPO + SP	Both (Hierarchical)	Custom	None	No
Selmonaj et al. [62] (2023)	BVR	Combat (Canon + rockets)	Multi Agent	IPPO	High-Level	Custom (2d grid)	None	Yes
Bae et al. [4] (2023)	WVR	Combat (Canon)	Single Agent	SAC with LSTM + CL	Low-level	JSBSim	Partial Observations (Simplified)	No
De Marco et al. [18] (2023)	WVR	Navigation	Single Agent	DDPG	Low-Level	JSBSim	None	No
Wang et al. [77] (2023)	WVR	Combat (Cannons)	Single Agent	DQN	High-Level	Custom	None	Yes
Scukins et al. [61] (2024)	BVR	- Evading Missile - Combat (Missiles)	Single Agent	PPO	Low-Level	JSBSim	None	Yes
Wang and Wang [75] (2024)	WVR	Combat (Missiles)	Multi Agent	Modified QMIX + basic SP	Low-Level (Hierarchical)	JSBSim	Partial Observations (Simplified)	No
Zhu et al. [86] (2024)	BVR	- Combat (Missiles) - Strike (Avoiding Missiles while reaching a specific point)	Single Agent	Custom PPO + CL + SP	High-Level	Custom (Based on Unity)	- Detection & RWR - Lock-on & missile guidance	No
Kuroswiski et al. [39] (2024)	BVR	- Interception (Missiles) - Strike (Avoiding Missiles while reaching a specific point)	Multi Agent	MAPPO / MADDPG	High-Level	Godot Engine	- Detection - Lock-on & missile guidance	Yes
Wang et al. [74] (2024)	WVR	Combat (Missiles)	Single Agent	PPO	Both	JSBSim	None	No
Piao et al. [51] (2024)	BVR	Combat (missiles)	Both	Custom	High-Level	WUKONG	- Detection - Lock-on & missile guidance	No
Kuroswiski et al. [40] (2025)	BVR	Combat (Missiles)	Multi Agent	Knowledge-informed Tasks for Attention-driven Behaviour (KTAB) (Custom, attention based)	High-Level	Godot Engine	- Detection - Lock-on & missile guidance	Yes
Huo et al. [34] (2025)	WVR	Combat (Missiles)	Multi Agent	GraphZero-PPO + SP	Both	Not Specified (Custom)	None	No

by a DQN-enhanced policy network with MHA [72] mechanisms, demonstrating an alternative hybrid between classical control and learned decision-making.

All the mentioned variations collectively illustrate how the choice of control granularity interacts with sample efficiency, stability, and interpretability of learned behaviour.

3.3 Simulated Sensors and Observability

The treatment of observability constitutes a critical methodological axis in autonomous air combat research. When agents have access to the complete global state, the problem is formalized as a MDP. When agents must act based on local and potentially incomplete information, the problem becomes a POMDP. Full observability

simplifies learning and accelerates convergence, but fails to capture the uncertainty and noise of real-world aerial engagements. Partial observability, while more realistic, introduces additional challenges, as agents must integrate imperfect information over time and reason under uncertainty. In both settings, the information available to agents depends on the sensors modelled within the simulation. Radar systems play a central role: in WVR scenarios, radar enables detection, tracking, and targeting of nearby adversaries, while in BVR engagements, it also determines engagement ranges, lock-on capabilities, and missile guidance. Other sensor modalities such as Radar Warning Receiver (RWR), Missile Approach Warning (MAW) systems, and Infrared (IR) sensors contribute to situational awareness. The fidelity with which these systems are modelled directly shapes the realism of the learning environment and the operational relevance of resulting policies.

Several surveyed works illustrate different approaches. Bae et al. [4] introduce a POMDP-based environment employing a forward-facing radar with distance-proportional noise. Opponents outside the radar cone can still be detected under simplified visual rules and policies are trained using SAC with LSTMs to integrate noisy observations over time. Kong et al. [38] impose radar and communication constraints: each aircraft operates with a conceptual 20-nautical-mile radar range and restricted detection cone, with intra-team communication limited to a 6000-foot radius.

In contrast, many earlier works assume full observability. Pope et al. [53] trained agents with full state access, including opponent information, enabling efficient convergence but sacrificing sensor realism. WUKONG-based works [28, 50, 66] similarly assume perfect opponent visibility, using radar only for lock-on and missile guidance.

Intermediate solutions also exist. BVR Gym [61] assumes adversaries are always visible but omits post-launch missile tracking, requiring agents to infer threats from onboard sensors. B-ACE [39] models radar volumetrically, providing full observability only within a defined cone while prohibiting enemy data sharing across teammates.

The key insight is that more precise and physically relevant modelling of radar and auxiliary systems yields more meaningful policies for real-world deployment. Accurate sensor models enable agents to develop tactics accounting for detection ranges, blind spots, and possibly countermeasures. However, achieving such fidelity is hindered by a fundamental obstacle: the performance characteristics of military radar and sensor systems are typically classified. This secrecy limits the extent to which open research can faithfully reproduce operational conditions. This leads us to introduce another axis in the following subsection: the distinction between open-source and closed-source simulation platforms.

3.4 Open-Source and Closed-Source Environments

The simulation environment is a decisive factor in autonomous air combat research, shaping not only the realism of the problem but also the reproducibility and comparability of results. Closed-source platforms typically emphasize detailed proprietary modelling of radar, missile guidance, and aircraft dynamics, but at the expense of transparency and external validation. Open-source environments, by contrast, facilitate community-driven development and benchmarking, although they often rely on simplifying assumptions to maintain tractability or rely on publicly disclosed data to model the multiple critical components (airframe, missiles, radar). The tension between fidelity and reproducibility is therefore central to understanding the strengths and limitations of existing approaches.

The WUKONG simulator exemplifies the closed-source category and has been used across a wide spectrum of studies. In the single-agent case, Piao et al. [50] employ it for 1v1 BVR combat, where agents act at the macro Basic Fighter Manoeuvres (BFMs) level under a customized PPO algorithm with targeted reward shaping and Self-Play(SP). Similarly Zhang et al. [83] apply WUKONG to missile duels, disclosing some missile parameters such as guidance logic and overload limits, yet withholding most environment details. Multi-agent extensions also rely on WUKONG: Sun et al. [66] introduce a hierarchical actor–critic method , while [28] design a GNN-based controller to capture tactical relations between agents. Piao et al. [51] also rely on WUKONG for training and testing its custom ENACTIVE algorithm. While algorithmically innovative, this contribution illustrates how reliance on a closed-source platform limits reproducibility and restricts independent validation of the environment's underlying assumptions. These works highlight WUKONG's capacity for modelling radar lock-on and missile guidance at different ranges, while also underscoring the barriers imposed by its proprietary nature. In a different direction, Yoo et al. [81] construct their closed-source solution on top of a publicly available third-party platform, namely Digital Combat Simulator (DCS) [20], thereby leveraging its commercial-grade complexity partially while still restricting access to the research community. Other closed-source approaches nonetheless build on open FDM backends: for instance [4, 38, 53, 75] rely on the JSBSim [36] flight dynamics engine, which is open-source but embedded within otherwise proprietary simulation pipelines.

Open-source initiatives offer an important counterbalance by prioritizing accessibility and standardization. BVR Gym [61], developed on JSBSim [36], provides modular scenarios, proportional navigation missile models, and direct compatibility with major RL toolkits. A distinctive feature is the omission of post-launch missile tracking, forcing agents to reason about threats from onboard sensors and missile metadata, thereby balancing simplified radar modelling with non-trivial tactical challenges. Similarly, B-ACE [39] provides an open-source environment built on the open-source Godot Engine [16], designed for high-level decision-making rather than fine-grained flight control. Its simplified altitude–speed coupling and volumetric radar modelling make it particularly suited to multi-agent BVR scenarios and coordination studies. Both platforms respond directly to reproducibility concerns raised by Costa et al. [17], by offering standardized, extensible testbeds for the broader MARL community.

Additional open-source contributions include the LAG environment [44], integrated into MARLlib [33] and also relying on JSBSim, which provides hierarchical task decomposition for both single-agent and team-based WVR engagements. More recent task-based extensions (e.g [40]), further highlight the versatility of open-source infrastructures for integrating hybrid decision-making architectures. On the closed-source side, elaborated implementations (e.g

[34]) remain inaccessible to the community, despite advancing hierarchical multi-agent controllers.

3.5 Computational Demands and Resource Requirements

Beyond methodological and reproducibility considerations, reinforcement learning for autonomous air combat is also defined by its substantial computational footprint. Reported experiments reveal a wide spectrum of hardware requirements, reflecting both the complexity of flight dynamics modelling and the diversity of simulation platforms. JSBSim-based studies are notably demanding on the CPU. Pope et al. [53] report that training on an Amazon EC2 P3.16xlarge instance equipped with 64 CPUs, 8 V100 GPUs, and 488 GB RAM was ultimately CPU-limited: policy selector networks required up to one week to train, while low-level manoeuvring policies required nearly a month. Bae et al. [4] similarly describe training durations of 18-21 days per agent type when using Xeon-class CPUs and RTX 2080Ti GPUs. By contrast, De Marco et al. [18] achieved significantly shorter training times using an Intel i7-9750H CPU and RTX 2060 GPU, illustrating the sensitivity of performance to task scope and fidelity. Kong et al. [38], Wang and Wang [75] employed consumer-grade hardware (respectively Intel i9-12900K + RTX 3060 Ti and Intel Xeon Silver 4210R CPU, RTX 3080 + 64 GB RAM), suggesting that scalable experimentation remains possible outside of data centre infrastructures.Environments built upon the WUKONG simulator show similarly varied requirements. Piao et al. [51] used parallel sampling across five Intel i7-9700K CPUs combined with two RTX 2080 SUPER GPUs for training, whereas earlier WUKONG-based works [28, 66] do not disclose hardware specifications. Custom frameworks also span a wide range: Huo et al. [34] report training on a Tesla P100-equipped Dell PowerEdge server, while Kuroswiski et al. [40] uses an Intel i9-19000KF CPU, 64 GB RAM, and an RTX 3070 GPU.

4 REPORTED LIMITATIONS AND BOTTLENECKS

While the surveyed literature highlights methodological progress, it also points to constraints that limit the generality and fidelity of current research. Authors frequently emphasize such challenges in their own discussions, identifying them as obstacles that remain unresolved. In what follows, we synthesize these recurring themes, organizing them along four main dimensions: sensing realism, scenario diversity, communication fidelity, and computational throughput.

A first recurring theme concerns sensing and observability. Several works [38–40, 61] acknowledge that their environments simplify radar or electromagnetic modelling, and that detection, lock-on procedures, and missile type diversity remain simplified or absent. As noted in their own discussions, this abstraction limits the fidelity and exploration of tactics central to Beyond Visual Range engagements.

The scope of simulated scenarios emerges as a second limitation. Reported studies frequently rely on symmetric engagements with homogeneous airframes (e.g [38, 44, 53]), and many restrict their focus to Within Visual Range manoeuvring (e.g [4, 14, 75, 77, 80]). This restriction narrows the range of tactical behaviours that can be meaningfully assessed, particularly in contexts involving heterogeneous platforms or asymmetric force compositions.

Communication modelling is identified as a third bottleneck. While many studies assume perfect or near-perfect intra-team communication, the authors themselves highlight that real-world operations are subject to intermittent connectivity, interference, or deliberate silence. As a result, the corresponding algorithms are not yet evaluated under degraded or contested communication conditions, which several papers [40, 62, 66, 75] suggest as a necessary direction for future research.

Finally, computational aspects are highlighted by [38, 53]. Environments based on physics engines such as JSBSim are repeatedly described as CPU-bound, which slows training despite the use of GPU acceleration for policy optimization. This bottleneck, typical of DRL approaches and explicitly mentioned in several studies, limits scalability when increasing the number of agents, sensors, or when incorporating more detailed physical models. Reported training times, sometimes extending over weeks [53], underline the practical challenges noted in the literature.

5 CONCLUSION

This survey reviewed reinforcement learning and multi-agent reinforcement learning in autonomous collaborative air combat, highlighting both notable progress and enduring challenges. In particular, we summarized the key characteristics of the works compared in Table 1. Advances include the adoption of hierarchical control, specialized algorithms for partial observability, and open-source simulation platforms. DRL methods such as SAC, PPO or QMIX, alongside hierarchical or graph-based approaches, have enabled agents to learn complex manoeuvres and coordinated strategies in simulation.

However, key limitations remain. Most studies assume simplified sensing, communication, and engagement scenarios, limiting realism and robustness. High-fidelity simulators offer accuracy but impose heavy computational costs and restrictions, while simplified environments scale more easily but can lack operational relevance. Moreover, current research predominantly focuses on homogeneous platforms and fully autonomous operations, leaving the coordination of heterogeneous assets largely unexplored despite their operational significance.

Future progress will depend on balancing fidelity with scalability, advancing sensor and communication models, and developing standardized benchmarks. Investment in GPU-accelerated simulation, adaptive fidelity methods, and interdisciplinary collaboration will be essential. While DRL shows strong potential, bridging the gap between simulation and operationally viable systems requires sustained methodological and infrastructural innovation.

ACKNOWLEDGMENTS

This publication was co-funded by the European Union under the Grant Agreement 101103669-EICACS (European Initiative on Collaborative Air Combat Standardization). Its contents are the sole responsibility of the author and do not necessarily reflect the views of the European Union or the European Commission. Neither the European Union nor the granting authority can be responsible for them.

REFERENCES

[1] 2025. Collaborative Combat Aircraft Program. https://en.wikipedia.org/wiki/Autonomous_aircraft.

[2] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches.* MIT Press.

[3] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis. 1990. Game theory for automated maneuvering during air-to-air combat. *Journal of Guidance, Control, and Dynamics* 13, 6 (1990), 1143–1149.

[4] Jung Ho Bae, Hoseong Jung, Seogbong Kim, Sungho Kim, and Yong-Duk Kim. 2023. Deep reinforcement learning-based air-to-air combat maneuver generation in a realistic environment. *IEEE Access* 11 (2023), 26427–26440.

[5] Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems* 13, 4 (2003), 341–379.

[6] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research* 47 (2013), 253–279.

[7] Richard Bellman. 1957. *Dynamic Programming.* Princeton University Press, Princeton, NJ.

[8] Yoshua Bengio, Jérémie Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning.* ACM, 41–48.

[9] Dimitri P. Bertsekas. 1995. *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, MA.

[10] G. H. Burgin. 1976. *Improvements to the Adaptive Maneuvering Logic Program.* NASA Contractor Report. NASA.

[11] G. H. Burgin and D. M. Eggleston. 1976. *Design of an all-attitude flight control system to execute commanded bank angles and angles of attack.* NASA Contractor Report. NASA.

[12] G. H. Burgin, L. J. Fogel, and J. P. Phelps. 1975. *An Adaptive Maneuvering Logic Computer Program for the Simulation of One-On-One Air-To-Air Combat.* NASA Contractor Report. NASA.

[13] L. Busoniu, R. Babuska, and B. De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (2008), 156 – 172. doi:10.1109/TSMCC.2007.913919

[14] Jiajun Chai, Wenzhang Chen, Yuanheng Zhu, Zong-Xin Yao, and Dongbin Zhao. 2023. A hierarchical deep reinforcement learning framework for 6-DOF UCAV air-to-air combat. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 9 (2023), 5417–5429.

[15] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014.*

[16] Godot Engine Community. 2024. *Godot Engine.* https://godotengine.org Open Source Game Engine.

[17] Andre N Costa, Joao PA Dantas, Edvards Scukins, Felipe LL Medeiros, and Petter Ã-gren. 2025. Simulation and Machine Learning in Beyond Visual Range Air Combat: A Survey. *IEEE Access* (2025).

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

[19] Christopher R DeMay, Edward L White, William D Dunham, and Johnathan A Pino. 2022. Alphadogfight trials: Bringing autonomy to air combat. *Johns Hopkins APL Technical Digest* 36, 2 (2022), 154–163.

[20] Eagle Dynamics. 2008. Digital Combat Simulator (DCS). https://www.digitalcombatsimulator.com/. Accessed: 2025-09-13.

[21] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning.* PMLR, 1146–1155.

[22] General Dynamics. [n. d.]. *F-16 Dash-1.* Basic drag and weight data only.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT Press.

[24] K. H. Goodrich. 1993. *A High-Fidelity, Six-Degree-Of-Freedom Batch Simulation Environment for Tactical Guidance Research and Evaluation.* Technical Report. NASA, Scientific and Technical Information Program, Washington, DC.

[25] Patrick Ribu Gorton, Andreas Strand, and Karsten Brathen. 2024. A survey of air combat behavior modeling using machine learning. *arXiv preprint arXiv:2404.13954* (2024).

[26] Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* 55, 2 (2022), 895–943.

[27] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on Machine Learning.* PMLR, 1861–1870.

[28] Yue Han, Haiyin Piao, Yaqing Hou, Yang Sun, Zhixiao Sun, Deyun Zhou, Shengqi Yang, Xuanqi Peng, and Songyuan Fan. 2022. Deep relationship graph reinforcement learning for multi-aircraft air combat. In *2022 International Joint Conference on Neural Networks (IJCNN).* IEEE, 1–8.

[29] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 33, 6 (2019), 750–797.

[30] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 33 (2019), 750–797.

[31] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[32] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366.

[33] Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Xiaodan Liang, Zhihui Li, Xiaojun Chang, and Yaodong Yang. 2022. MARLlib: A Scalable and Efficient Multi-agent Reinforcement Learning Library. *arXiv preprint arXiv:2210.13708* (2022). https://arxiv.org/abs/2210.13708.

[34] Lin Huo, Chudi Wang, and Yue Han. 2025. Autonomous air combat decision making via graph neural networks and reinforcement learning. *Scientific Reports* 15, 1 (2025), 16169.

[35] R. Isaacs. 1951. *Games of pursuit.* Technical Report P-257. RAND Corporation, Santa Monica, CA, USA.

[36] JSBSim Team. 1996-Present. *JSBSim.* http://jsbsim.sourceforge.net/ Open Source Flight Dynamics Model.

[37] Rudolf Emil Kalman. 1960. Contributions to the theory of optimal control. *Boletín de la Sociedad Matemática Mexicana* 5, 2 (1960), 102–119.

[38] Wei-ren Kong, De-yun Zhou, Yong-jie Du, Ying Zhou, and Yi-yang Zhao. 2023. Hierarchical multi-agent reinforcement learning for multi-aircraft close-range air combat. *IET Control Theory & Applications* 17, 13 (Sept. 2023), 1840–1862. doi:10.1049/cth2.12413

[39] Andre R. Kuroswiski, Annie S. Wu, and Angelo Passaro. 2024. B-ACE: An Open Lightweight Beyond Visual Range Air Combat Simulation Environment for Multi-Agent Reinforcement Learning. In *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2024.* doi:10.13140/RG.2.2.11999.57762

[40] Andre R. Kuroswiski, Annie S. Wu, and Angelo Passaro. 2025. Enhancing MARL BVR Air Combat using Domain Expert Knowledge at the Action Level. *IEEE Access* 13 (2025), 70446–70463. doi:10.1109/ACCESS.2025.3561250

[41] Frank L. Lewis, Draguna L. Vrabie, and Vassilis L. Syrmos. 2012. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control.* Wiley, New York, NY.

[42] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning.*

[43] Long-Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning* 8, 3 (1992), 293–321.

[44] Qihan Liu, Yuhua Jiang, and Xiaoteng Ma. 2022. Light Aircraft Game: A lightweight, scalable, gym-wrapped aircraft competitive environment with baseline reinforcement learning algorithms. https://github.com/liuqh16/CloseAirCombat.

[45] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[46] J. S. McGrew, J. P. How, and B. Williams. 2010. Air-combat strategy using approximate dynamic programming. *Journal of Guidance, Control, and Dynamics* 33, 5 (2010), 1641–1654.

[47] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[48] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. 2019. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737* (2019).

[49] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. 2021. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–35.

[50] Haiyin Piao, Zhixiao Sun, Guanglei Meng, Hechang Chen, Bohao Qu, Kuijun Lang, Yang Sun, Shengqi Yang, and Xuanqi Peng. 2020. Beyond-visual-range air combat tactics auto-generation by reinforcement learning. In *2020 international joint conference on neural networks (IJCNN).* IEEE, 1–8.

[51] Hai Yin Piao, Shengqi Yang, Hechang Chen, Junnan Li, Jin Yu, Xuanqi Peng, Xin Yang, Zhen Yang, Zhixiao Sun, and Yi Chang. 2024. Discovering expert-level air combat knowledge via deep excitatory-inhibitory factorized reinforcement learning. *ACM Transactions on Intelligent Systems and Technology* 15, 4 (2024), 1–28.

[52] Adrian P Pope, Jaime S Ide, Daria Mićović, Henry Diaz, David Rosenbluth, Lee Ritholtz, Jason C Twedt, Thayne T Walker, Kevin Alcedo, and Daniel Javorsek. 2021. Hierarchical reinforcement learning for air-to-air combat. In *2021 international conference on unmanned aircraft systems (ICUAS).* IEEE, 275–284.

[53] 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. 2022.

Hierarchical reinforcement learning for air combat at DARPA's AlphaDogfight trials. *IEEE Transactions on Artificial Intelligence* 4, 6 (2022), 1371–1385.
[54] Martin L. Puterman. 2014. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
[55] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research* 21, 178 (2020), 1–51.
[56] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.
[57] Gavin A. Rummery and Mahesan Niranjan. 1994. On-line Q-learning using connectionist systems. *Technical Report* (1994).
[58] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. In *International Conference on Learning Representations*.
[59] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.
[60] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. In *Proceedings of the 34th International Conference on Machine Learning (ICML) Workshop*. arXiv:1707.06347
[61] Edvards Scukins et al. 2024. BVR Gym: A Reinforcement Learning Environment for Beyond-Visual-Range Air Combat. *arXiv preprint arXiv:2403.17533* (2024). https://arxiv.org/abs/2403.17533
[62] Ardian Selmonaj, Oleg Szehr, Giacomo Del Rio, Alessandro Antonucci, Adrian Schneider, and Michael Rüegsegger. 2023. Hierarchical multi-agent reinforcement learning for air combat maneuvering. In *2023 International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 1031–1038.
[63] Shield AI. 2021. Shield AI Acquires Heron Systems. https://shield.ai/shield-ai-acquires-heron-systems/. Accessed: 2025-07-13.
[64] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of Go without human knowledge. *Nature* 550, 7676 (2017), 354–359.
[65] Andreas Strand, Patrick Ribu Gorton, and Karsten Brathen. 2025. Modeling air combat behavior for simulation-based pilot training: A survey of machine learning approaches. *IEEE Access* (2025).
[66] Zhixiao Sun, Haiyin Piao, Zhen Yang, Yiyang Zhao, Guang Zhan, Deyun Zhou, Guanglei Meng, Hechang Chen, Xing Chen, Bohao Qu, et al. 2021. Multi-agent hierarchical policy gradient for air combat tactics emergence via self-play. *Engineering Applications of Artificial Intelligence* 98 (2021), 104112.
[67] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2085–2087.
[68] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning: an introduction*. MIT Press, Cambridge, Mass.
[69] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. In *Artificial intelligence*, Vol. 112. Elsevier, 181–211.
[70] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*. 330–337.
[71] T.-H. Teng, A.-H. Tan, Y.-S. Tan, and A. Yeo. 2012. Self-organizing neural networks for learning air combat maneuvers. In *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[73] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. FeUdal networks for hierarchical reinforcement learning. *International conference on machine learning* (2017), 3540–3549.
[74] Dinghan Wang, Jiandong Zhang, Qiming Yang, Jieling Liu, Guoqing Shi, and Yaozhong Zhang. 2024. An Autonomous Attack Decision-Making Method Based on Hierarchical Virtual Bayesian Reinforcement Learning. *IEEE Trans. Aerospace Electron. Systems* 60, 5 (2024), 7075–7088.
[75] Huan Wang and Jintao Wang. 2024. Enhancing multi-UAV air combat decision making via hierarchical reinforcement learning. *Scientific Reports* 14, 1 (2024), 4458.
[76] Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. 2022. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 35, 4 (2022), 5064–5078.
[77] Xinwei Wang, Yihui Wang, Xichao Su, Lei Wang, Chen Lu, Haijun Peng, and Jie Liu. 2024. Deep reinforcement learning-based air combat maneuver decision-making: literature review, implementation tutorial and future direction. *Artificial Intelligence Review* 57, 1 (2024), 1.
[78] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* 8, 3-4 (1992), 279–292.
[79] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
[80] Qiming Yang, Jiandong Zhang, Guoqing Shi, Jinwen Hu, and Yong Wu. 2019. Maneuver decision of UAV in short-range air combat based on deep reinforcement learning. *IEEE Access* 8 (2019), 363–378.
[81] Jaewoong Yoo, Hyunki Seong, David Hyunchul Shim, Jung Ho Bae, and Yong-Duk Kim. 2022. Deep reinforcement learning-based intelligent agent for autonomous air combat. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*. IEEE, 1–9.
[82] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems* 35 (2022), 24611–24624.
[83] Hongpeng Zhang, Yujie Wei, Huan Zhou, and Changqiang Huang. 2022. Maneuver decision-making for autonomous air combat based on FRE-PPO. *Applied sciences* 12, 20 (2022), 10230.
[84] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control* (2021), 321–384.
[85] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems* 33 (2020), 11853–11864.
[86] Jingyu Zhu, Minchi Kuang, Wenqing Zhou, Heng Shi, Jihong Zhu, and Xu Han. 2024. Mastering air combat game with deep reinforcement learning. *Defence Technology* 34 (2024), 295–312.