A Multi-Aircraft Co-Operative Trajectory Planning Model Under Dynamic Thunderstorm Cells Using Decentralized Deep Reinforcement Learning

Bizhao Pang [a, #], Xinting Hu [a, b, #], Mingcheng Zhang [b], Sameer Alam [a, b, *] and Guglielmo Lulli [c]

a. Air Traffic Management Research Institute, Nanyang Technological University, 637460 Singapore

b. School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798 Singapore

c. Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20126 Milano, Italy

Abstract: Climate change induces an increased frequency of adverse weather, particularly thunderstorms, posing significant safety and efficiency challenges in en-route airspace, especially in oceanic regions where air traffic control services are limited. These conditions require multi-aircraft cooperative trajectory planning to avoid both dynamic thunderstorms and other aircraft. Existing literature has typically relied on centralized approaches and single-agent principles, which lack coordination and robustness when surrounding aircraft or thunderstorms change paths, leading to scalability issues due to heavy trajectory regeneration needs. To address these gaps, this paper introduces a multi-agent cooperative framework for autonomous trajectory planning. The problem is modeled as a Decentralized Markov Decision Process (DEC-MDP) and solved using an Independent Deep Deterministic Policy Gradient (IDDPG) learning framework. A shared actor-critic network is trained using combined experiences from all aircraft to optimize joint behavior. During execution, each aircraft acts independently based on its own observations, with coordination ensured through the shared policy. The model is validated through extensive simulations, including uncertainty analysis, baseline comparisons, and ablation studies. With known thunderstorm paths, the model achieved a 2% loss of separation rate, which increased to 4% under random storm paths. An ETA uncertainty analysis demonstrated the model's robustness, while baseline comparisons with the state-of-the-art Fast Marching Tree and centralized DDPG highlighted its scalability and efficiency. These findings contribute to autonomous aircraft operations, especially in oceanic airspace with limited ATC support.

Keywords: Air traffic management, autonomous trajectory planning, multi-aircraft coordination, deep reinforcement learning, dynamic thunderstorm cells, climate change

[*] Corresponding author: Sameer Alam (sameeralam@ntu.edu.sg).

[#] These two authors contributed equally to this work.

1. Introduction

1.1 Research motivations

Climate change is impacting air transport through adverse weather, particularly thunderstorms, with a rise in frequency and severity [1]. These weather phenomena are highly disruptive, causing aircraft to lose separation and leading to severe turbulence, sometimes resulting in severe injuries [2]. The adverse weather has also, for the first time, become the leading reason for en-route air traffic flow management (ATFM) delays in Europe [1], as shown in **Fig. 1**. These dynamic thunderstorm cells can develop rapidly, obstructing nominal flight paths and requiring immediate and effective trajectory planning [3]. The challenge is further compounded when multiple aircraft are involved, each needing to make real-time adjustments to avoid both other aircraft and the evolving thunderstorm cells [4]. This is especially critical in oceanic airspace, where air traffic control coordination is limited or unavailable. In such conditions, the absence of proper coordination among multiple aircraft can lead to numerous loss-of-separation incidents [5]. Addressing this issue is crucial, given the increasing unpredictability and intensity of weather phenomena [6], highlighting the need for automated tools that provide cooperative and scalable solutions to ensure safety and efficiency in increasingly congested airspace.

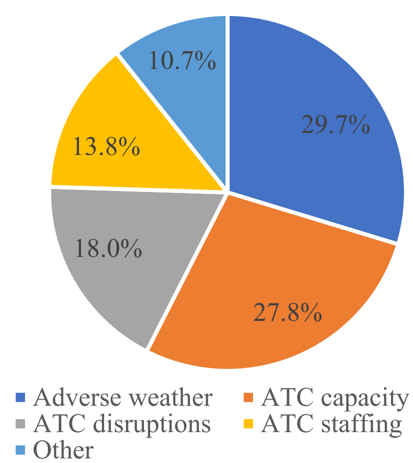


Fig.1. Reasons for en-route ATFM delay in 2023 by Eurocontrol [1].

Current strategies in the Air Traffic Control (ATC) system to address these challenges are implemented in two phases: strategic and tactical. In the strategic phase, Air Navigation Service Providers (ANSPs), such as network managers, commonly employ air traffic flow control strategies [7,8]. These strategies involve regulating the number of flights entering weather-affected airspace hours in advance, thereby reducing traffic flow complexity and the workload for pilots and Air Traffic Control Officers (ATCOs) [9]. However, the effectiveness of these strategies heavily depends on the accurate and timely prediction of thunderstorm weather, which is inherently difficult to achieve with precision. Inaccurate predictions can lead to the formation of airspace hotspots during the tactical flight phase that require immediate resolution [10]. Additionally, minimizing the use of flow control strategies has become a major objective in the ATFM system to reduce flight delays and cancellations [1]. The combination of increased traffic density and adverse weather conditions is likely to exacerbate traffic complexity during the tactical phase.

In the tactical phase, coordination among multiple aircraft is critical. ATCOs are responsible for this coordination, ensuring that aircraft maintain safe separation from each other and hazardous thunderstorm cells. However, this task is particularly challenging for ATCOs due to their limited cognitive capabilities [11,12], especially under rapidly evolving weather conditions that demand constant resolution and traffic flow reorganization. The continuous adaptation required by changing weather conditions significantly increases the complexity and workload for ATCOs. In areas where radar coverage and ATC services are not available (e.g., oceanic airspace), the situation is even more challenging, as the coordination among multiple rerouting aircraft is missing [13]. The lack of coordination among multiple rerouting trajectories in such environments poses significant risks, making it urgently needed to develop automated methods for ensuring safe separation between aircraft and dynamic thunderstorm cells.

To assist pilots and ATCOs in managing complex tasks in these scenarios, several automation tools have been developed for conflict resolution and flight rerouting [14–17]. However, existing research has several limitations in terms of coordination and scalability. For example, one model plans each flight's trajectory by treating the paths of preceding flights as obstacles [15]. This approach requires recalculations if any surrounding aircraft or thunderstorm cells change course, reducing robustness and increasing computational burden, making it challenging for these models to perform effectively in dynamic and unpredictable environments. Additionally, a cooperative decision-making framework is absent. Most current methods rely on the First-Come-First-Served principle [18], generating conflict-free trajectories for individual aircraft without enabling coordination between them. This single-agent approach limits the ability to manage multiple aircraft collaboratively, thereby restricting overall safety, efficiency, and fairness. Furthermore, the fast-changing nature of thunderstorm conditions and the unique combinations of traffic flow and weather patterns require a scalable and computationally efficient framework, which current models lack.

1.2. Contributions

The above-mentioned challenges motivate us to explore a cooperative, robust, and scalable multi-agent framework for multi-aircraft trajectory planning under dynamic traffic and weather conditions. This framework is expected to: (a) effectively handle multi-aircraft trajectory planning in dynamic and complex airspace environments; (b) maintain robustness against trajectory dependency, accommodating emerging disruptions without recalculating all existing trajectories; (c) generalize effectively to unseen scenarios where traffic and weather patterns combine in diverse ways. The key contributions of this research are summarized as follows.

(1) We propose a novel multi-agent cooperative framework for autonomous multi-aircraft trajectory planning under rapidly evolving thunderstorm cells. Our framework addresses research gaps related to trajectory dependency and robustness in dynamic thunderstorm conditions. A key aspect of our contribution is the development of a custom-built simulator that models dynamic thunderstorm cells during the training phase, enabling our model to adapt to real-world weather patterns and traffic complexities.

(2) We develop a Decentralized Markov Decision Process (DEC-MDP) model for cooperative multi-aircraft trajectory planning. To solve the problem, we proposed an Independent Deep Deterministic Policy Gradient (IDDPG) algorithm with the introduction of shared neural networks for consistent training and target networks for enhanced stability.

(3) We validate the robustness, generalization, and scalability of the proposed framework by solving complex multi-aircraft trajectory planning tasks under diverse combinations of traffic density and weather conditions in both real-world and simulated airspace.

(4) We conduct baseline comparisons of the proposed decentralized multi-agent IDDPG with the state-of-the-art Fast Marching Tree and centralized DDPG models. The results demonstrate the strong robustness and scalability of the IDDPG model, particularly in high-density airspace scenarios with dynamic thunderstorms.

The rest of the paper is structured as follows: Section 2 reviews existing studies on flight trajectory planning under convective weather and the applications of reinforcement learning in air traffic management. Section 3 presents the problem formulation in its mathematical form. Section 4 details the formulation of the decentralized Markov Decision Process and the framework of the IDDPG algorithm. Section 5 discusses simulation environment, model training, and performance testing results. Finally, Section 6 concludes this work and suggests future research directions.

2. Related Works

This section reviews the existing literature on flight trajectory planning under thunderstorm conditions, highlighting the strengths and limitations of various methods and algorithms. We also explore recent advancements in reinforcement learning techniques applied to ATM fields. Finally, we identify the research gaps in addressing the challenges of dynamic multi-aircraft rerouting under rapidly evolving thunderstorm cells.

2.1. Aircraft trajectory planning under thunderstorms

Trajectory planning under thunderstorm weather has attracted significant attention from researchers, exploring various problem settings across different phases of flight operations. These studies have considered both static and dynamic thunderstorm cells and employed methodologies such as geometrical methods, optimization models, and heuristic algorithms.

Erzberger et al. [14] pioneered the development of automated conflict resolution algorithms aimed at enhancing safety and airspace capacity for future air traffic control systems. Building on this, Erzberger et al. [16] presented a collective separation assurance tool named Autoresolver, which includes an Arrival Manager algorithm for flight sequencing, a Resolution Generator for solving conflict, and a Weather Avoidance algorithm, as illustrated in **Fig. 2**. The key feature was the generation of conflict-free trajectories through a multi-step iterative process, which provides foundations for subsequent research in trajectory planning. This research also highlighted the need for decentralized and real-time separation assurance approaches to allow input from individual pilots.

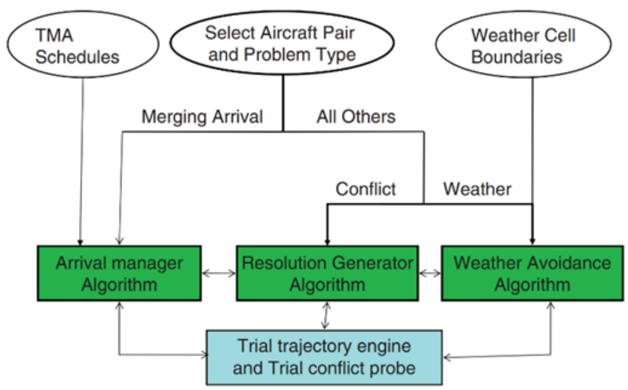


Fig. 2. Functional diagram of Autoresolver by Erzberger et al. [16].

Follow-up works explored the uncertainties in trajectory planning problems. Ng et al. [19] studied the flight rerouting problem under weather uncertainties at the pre-departure phase and developed a dynamic programming model to calculate the probabilities of potential route deviations, aiming to reduce fuel and route deviation costs. Kamgarpour et al. [20] addressed the gap in trajectory planning under dynamic weather uncertainties by proposing a receding horizon control framework. This framework generates rerouting trajectories using a constrained optimization model, solved via an optimization solver. Results confirmed the benefits of considering dynamic weather conditions over static ones. However, the study faced computational efficiency and scalability challenges, indicating the need for a more efficient method. In another work, Zhang et al. [21] proposed a simulation-based approach to quantify the impact of various uncertainties on aircraft rerouting, but these time-consuming methods are primarily useful for rerouting in the pre-tactical phase rather than real-time separation assurance. Hentzen [18] introduced a model for the stochastic development of thunderstorm cells, applying stochastic optimal control to generate safety-optimal trajectories for a single flight. However, this model struggles with real-time computation, reducing its applicability when new weather updates are available. In a follow-up study, Taylor et al. [22] focused on generating diverse reroutes for tactical constraint avoidance using multi-objective optimization with Dijkstra and Genetic Algorithms. While effective at the strategic phase, this approach falls short in real-time computation and coordination among multiple aircraft, particularly under evolving weather conditions.

Another group of studies focused on tactical decision making under thunderstorm weather conditions. Pannequin et al. [23] developed a nonlinear model predictive control (NMPC) method for multi-aircraft motion planning, assuming static weather conditions. Although their simulations produced locally optimal trajectories to minimize time or fuel costs, the scalability and assumption of static weather are major limitations. Summers et al. [24] explored reach-avoid navigation problems in stochastic environments with time-varying obstacles. Results demonstrated effectiveness in aircraft motion planning under uncertain weather but oversimplified the thunderstorm cells' movement and shape. To better understand the evolvement of thunderstorm cells, González-Arribas et al. [25] estimated fast-developing thunderstorms and used optimal control methods to generate robust flight trajectories against uncertain convective weather, yet scalability and conflict resolution remain problematic. Seenivasan et al. [26] proposed a feedback mechanism using optimal control for dynamic planning during the arrival phase under uncertain thunderstorm cells. Despite showing promise, their study faced challenges in scalability and coordination.

Heuristic and sampling-based methods have been employed to quickly generate rerouting trajectories. Liu et al. [15] studied multiple aircraft conflict resolution using a probabilistic conflict risk map, employing the A* algorithm to search for conflict-free trajectories. However, the method's scalability and robustness are limited as all other aircraft are treated as intruders during path search. Andres et al. [27] proposed a scenario-based RRT* for near real-time trajectory planning under ensemble thunderstorm forecasts, focusing on single-flight safety and efficiency. Their follow-up work [3] employed an Augmented Random Search algorithm to handle thunderstorm evolution uncertainties but faced issues in scalability and coordination among multiple flights. To fill these gaps, Guitart et al. [28] proposed a sample-based path planning algorithm for on-board conflict-free flight trajectory generation. Although effective, trajectory dependency and scalability issues remain unsolved.

In summary, current methods for trajectory planning are predominantly centralized, posing challenges in coordination, robustness, and scalability, especially in future high-density traffic environments. Incorporating dynamic weather updates remains difficult as many models assume static weather, leading to unsafe and inefficient trajectories. Furthermore, these methods are often computationally inefficient and lack the robustness needed for effective decision-making in dynamic and complex traffic and weather conditions.

2.2. Reinforcement learning applications in Air Traffic Management

One of the main applications of RL in ATM is for flight conflict detection and resolution (CD&R). Pioneer work has contributed to applying RL in ATM and Urban Air Mobility (UAM) fields, with a particular focus on aircraft separation assurance [29,30]. For example, Brittain and Wei [31] addressed tactical multi-agent CD&R in the en-route phase using a deep multi-agent RL (MARL) model with Proximal Policy Optimization (PPO) to control discrete speed, demonstrating scalability and efficiency in simulation environments. Pham et al. [32] proposed a two-layer Deep Deterministic Policy Gradient (DDPG) algorithm for CD&R to optimize the vectoring angles and timing for aircraft course changes. They provide a solution framework (as shown in **Fig. 3**) for aircraft CD&R under uncertainties by using deep learning techniques, which showed promise in handling high-density traffic under uncertainties. Zhao and Liu [33] integrated physics-based knowledge into RL for conflict resolution, creating human-explainable results with a solution space diagram but encountered challenges in practical ATC applications due to limited consideration of real-world constraints. Chen et al. [34] explored generalization in RL models for CD&R by using adaptive maneuver strategies for action selection. Their findings revealed that as traffic density increases, the flight distance in CD&R tasks also increases, highlighting a trade-off between maintaining safe separation and optimizing flight efficiency. Papadopoulos et al. [35] validated a deep multi-agent reinforcement learning model in simulated real-world environments, proving effective but requiring alignment with standard ATCO procedures. To fill this gap, a follow-up study, by Guleria et al. [36] developed a machine learning model to predict ATCOs' preference for conflict resolution, addressing acceptance issues of AI decisions by human controllers, and demonstrating effectiveness in predicting resolution preferences based on ATCOs' behavior.

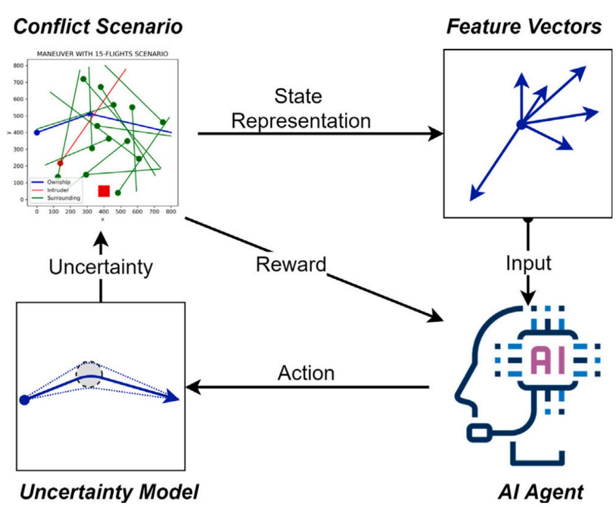


Fig. 3. The concept diagram for the interaction between the learning environment and the agent by Pham et al. [32].

RL methods have also been applied to ATFM challenges, focusing on optimizing traffic flow and managing disruptions. Kravari et al. [37] proposed a multi-agent RL approach for solving airspace congestion during the pre-tactical phase, showing superiority over conventional centralized methods by balancing ground delay decisions across flight agents. However, the focus on ground delay strategies before operations limits its effectiveness for tactical rerouting under convective weather. Pham et al. [38] introduced a novel DRL model for real-time departure slotting in mixed-mode runway operations, enhancing computational efficiency and handling stochastic runway capacity issues. Ali et al. [39] addressed the departure metering problem, proposing a model-free DRL method to manage pushback timings and reduce fuel consumption and emissions. Lee et al. [40] focused on airline disruption recovery, modeling the problem as an MDP and solving it with a Double Q-learning method to minimize total delays. Wang et al. [41] integrated RL with prescriptive analytics, which achieves significant improvements in computation time while maintaining optimality. Spatharis et al. [42] introduced a hierarchical MDP for the demand capacity balancing (DCB) problem at the pre-tactical level, enhancing coordination performance by utilizing multiple levels of abstraction in action and state spaces. In a follow-up work, Chen et al. [43] proposed a general MARL framework with an LSTM network to improve generalization in solving DCB problems, with a heuristic-based delay priority strategy to enhance learning efficiency. Ding et al. [44] employed DRL to improve the efficiency and scalability of a heuristic Variable Neighborhood Search algorithm for instant airline disruption recovery.

Despite the progress, existing RL applications in air traffic management often address single tasks such as conflict resolution or flow management in isolation. However, the complex and dynamic nature of multi-aircraft trajectory planning under rapidly evolving thunderstorms requires integrated solutions that can simultaneously handle multiple tasks, such as separation assurance, thunderstorm cell avoidance, and adherence to exit waypoints.

3. Problem Formulation

In this study, we address the research problem of multi-aircraft co-operative trajectory planning under rapidly evolving thunderstorm cells. The objective is to ensure separation assurance among multiple aircraft and thunderstorm cells while minimizing overall flight distance. To clarify the scope of the problem, we made several notes and assumptions as follows.

(a) The study assumes fully autonomous operations, with no intervention from human agents such as pilots or Air Traffic Control Officers (ATCOs). This allows for the exploration of DRL-based solutions without the variability introduced by human decision-making.

(b) The focus is on en route airspace, particularly in procedure airspace with limited or unavailable ATC services, such as oceanic regions, where autonomous and distributed decision-making is essential.

(c) Thunderstorm cells are considered cumulonimbus clouds, which are significant hazards that aircraft must avoid. The primary strategy for avoiding these cells is heading changes, as adjustments in altitude or speed are less effective and less commonly employed in practice.

(d) Thunderstorm cell information is assumed to be available through meteorological forecasting tools and aircraft onboard weather radar, which provides real-time data on the location, size, and movement trajectories of thunderstorm cells. For simulation purposes, we employ reasonable parameters to model the dynamic behavior of evolving thunderstorm cells to reflect realistic and challenging operational conditions.

(e) It is assumed that flight information among multiple aircraft within a certain range is shared via onboard ADS-B systems, including data on position, heading, speed, and pre-planned waypoints.

3.1. Problem statement

Thunderstorms or rapidly developing storms pose significant challenges in busy airspace due to their unpredictability and swift onset. As depicted in **Fig. 4**, emerging thunderstorm cells can obstruct key air routes such as L101, L102, L103, and L104, forcing multiple aircraft to deviate from their planned paths and quickly decide on new trajectories. Managing these rerouted trajectories in real-time, within such a complex and dynamic environment, becomes especially challenging when dealing with high-density air traffic interactions [45] and unstable thunderstorm conditions [4].

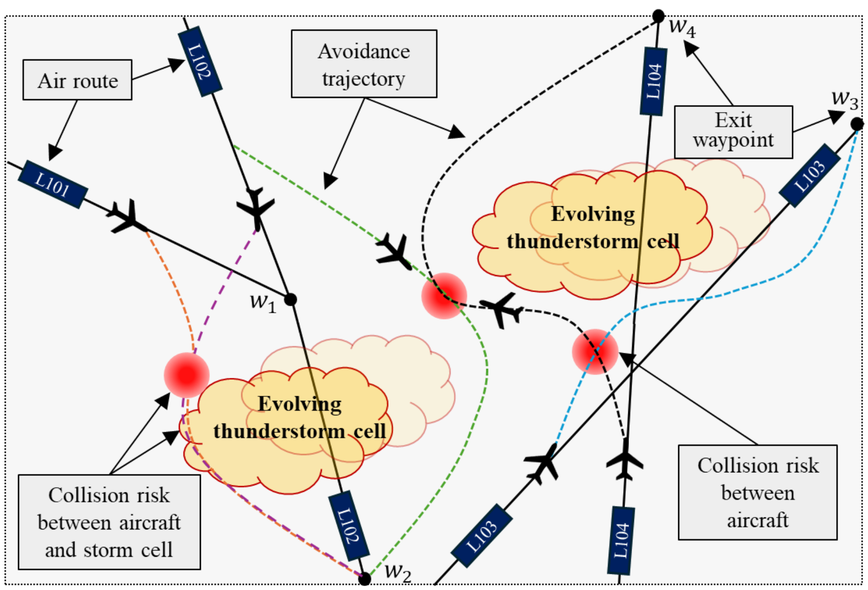


Fig. 4. Illustration of multi-aircraft trajectory planning under dynamic thunderstorm cells.

A primary concern in this scenario is maintaining safe separation between aircraft as they navigate these new trajectories, while also minimizing the additional distance flown. For example, aircraft rerouted from air route L102 may face conflicts with those deviating from L104, potentially creating collision risk hotspots. Similarly, flights avoiding thunderstorm cells on L103 may intersect others on alternate routes, further complicating traffic management and increasing the likelihood of mid-air collisions.

Additionally, aircraft rerouted from L101 and L102 must carefully avoid not only each other but also the evolving thunderstorm cells that are impacting these routes. The risk of both aircraft-to-aircraft and aircraft-to-storm cell collisions adds further complexity to the rerouting process. Despite these deviations during rerouting, flights must ultimately rejoin their nominal air routes at designated exit waypoints (w_1, w_2, w_3, w_4) to continue toward their destinations, with the assurance that no aircraft will be approaching from the opposite direction at the same flight level on these routes. These potential conflicts highlight the need for a robust system capable of dynamically adapting to rapidly changing weather conditions and ensuring safe separation.

3.2. Mathematical model

We develop a mathematical formulation with defined parameters, objectives, and constraints, which provides a clear scope and definition of the research problem.

Notations

- Aircraft: let $AC = \{ac_1, ac_2, \dots, ac_n\}$ be the set of aircraft in the weather-affected airspace.
- Exit waypoints: let $W = \{w_1, w_2, \dots, w_l\}$ be the set of positions of exit waypoints.
- Time: let $T = \{t_0, t_1, \dots, t_{\text{final}}\}$ denote discrete time steps, where t_0 is the rerouting start time, and t_{Final} is when an aircraft reaches its exit waypoint w_l and leaves the affected airspace.
- Weather thunderstorm cells: let $O^{\text{W}}(t) = \{o_1^{\text{W}}(t), o_2^{\text{W}}(t), \dots, o_j^{\text{W}}(t)\}$ be the set of thunderstorm cells' range at time t. Each cell acts as a time-dependent obstacle.
- Centroid of the thunderstorm cell: $p_{O^{\text{W}}}(t)$ is the centroid position of the thunderstorm cell O^{W} at t.
- Aircraft position: the two-dimensional position of aircraft ac_i at time t is denoted by $p_i(t) \in \mathbb{R}^2$.
- Safe separation: let $d_{\min}^{\text{flight}}$ be the separation minima between aircraft, and $d_{\min}^{\text{W}}$ be the minimum distance to thunderstorm cells.

Decision variables

- $\Delta h_i(t)$: represents heading change value for aircraft i at t. The heading change is taken as a continuous variable for autonomous aircraft operations.
- $\alpha_i(t)$: an artificial variable representing the heading of aircraft ac_i at t.
- $p_i(t)$: an artificial variable representing the position of aircraft ac_i at t.

Objective functions:

One objective is to minimize the total distance each aircraft travels from its current position $p_i(t)$ at each time step to its respective exit waypoint $p_i(t_{\text{final}})$, simply minimizing the distance between consecutive positions. This approach encourages direct paths for aircraft, which may lead to deviations from their original routes if those routes are not the most efficient. Consequently, this model is particularly suited for airspace operations where avoidance of dynamic thunderstorm cells is required, but it may need modifications for non-weather airspace operations where more structured route adherence is necessary.

$$\min \sum_{i=1}^{n} \sum_{t=t_0}^{t_{\text{final}}} \|p_i(t) - p_i(t_{\text{final}})\| \tag{1}$$

Another objective is to minimize heading change maneuvers, as they increase aircraft interactions and airspace complexity, potentially leading to passenger injuries. We define Δh as a constant threshold for each heading change, making $\left|\frac{\Delta h_i(t)}{\Delta h}\right|$ in the range of [0, 1].

$$\min \sum_{i=1}^{n} \sum_{t=t_0}^{t_{\text{final}}} \left|\frac{\Delta h_i(t)}{\Delta h}\right| \tag{2}$$

Constraints:

Aircraft separation constraint. The distance between any two aircraft a_i and a_j at any given time t must be greater than or equal to the minimum separation distance $d_{\min}^{\text{flight}}$.

$$\left\|p_i(t) - p_j(t)\right\| \geq d_{\min}^{\text{flight}}, \forall i, j \in \{1, \dots, n\}, i \neq j, \forall t \in T \tag{3}$$

Thunderstorm cell separation constraint. The distance of an aircraft (position $p_i(t)$) from the centroid of the thunderstorm cells ($p_{O^{\text{W}}}(t)$) at any given time t is greater than the minimum separation distance $d_{\min}^{\text{W}}$ plus the radius of the major axis of the thunderstorm cell ellipse $\text{R}_{\text{major\_axis}}^{\text{W}}$.

$$\|p_i(t) - p_{O^{\text{W}}}(t)\| \geq d_{\min}^{\text{W}} + \text{R}_{\text{major\_axis}}^{\text{W}}, \forall i \in \{1, \dots, n\}, \forall t \in T \tag{4}$$

Exit waypoint reachability constraint. To ensure rerouted flights resume their nominal air routes, each aircraft must reach its assigned exit waypoint by the final time step.

$$p_i(t_{\text{final}}) = w_{a_i}, \forall i \in \{1, \dots, n\} \tag{5}$$

Aircraft Dynamics. The position of each aircraft at each time step is updated based on its current position, velocity, and heading change. Let $p_i(t) = (x_i(t), y_i(t))$ represent the position of aircraft i at t, $v_i(t)$ be the velocity, and $\Delta h_i(t)$ the heading change value. The position update model is as follows:

$$p_i(t+1) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} + v_i(t)\Delta t \begin{pmatrix} \cos(\alpha_i(t+1)) \\ \sin(\alpha_i(t+1)) \end{pmatrix} \tag{6}$$

$$\alpha_i(t+1) = \alpha_i(t) + \Delta h_i(t) \quad (7)$$

where $\alpha_i(t+1)$ is the new heading angle after applying the heading change, $p_i(t+1)$ is the new position, and Δt is the time step duration.

The problem formulation is a multi-objective nonlinear and non-convex programming problem, which poses severe computational challenges. Therefore, it is impractical for time-sensitive applications like those encountered in dynamic air traffic management that require a solution almost in real-time. Additionally, certain adaptive and experience-based decision-making features, such as learning from past trajectory adjustments in similar weather conditions, cannot be effectively captured through mathematical formulations alone. These features rely on an agent's ability to accumulate experience over time, improving its decision-making through reinforcement learning in dynamic and uncertain environments. These need for handling multiple objectives, ensuring fast computation, and modeling implicit features render conventional model-based optimization and meta-heuristic algorithms insufficient, motivating us to explore learning-based methods to solve this problem.

4. DEC-MDP Model and IDDPG Solution

4.1. Decentralized Markov Decision Process (DEC-MDP)

We reformulate the optimization problem as a Markov Decision Process, as it involves a sequence of decisions in a dynamic and uncertain environment, which aligns well with the capabilities of MDPs. The complexity of this problem arises from the need for each aircraft to continuously adapt to evolving conditions, such as weather changes and the movements of other aircraft. In this setting, each aircraft functions as an independent agent with its own observations and state transitions, while all agents collectively aim to maximize the expected joint reward for the entire system. Therefore, we model the problem as a decentralized Markov Decision Process (DEC-MDP) defined for multiple agents.

For a system of n agents, the DEC-MDP is defined by the tuple (S, A, R, P), where:

- $S = S_1 \times S_2 \times, \dots, \times S_n$: represents the joint state space of all agents, with S_i the state space of agent i.
- $A = A_1 \times A_2 \times, \dots, \times A_n$: represents the joint action space of all agents, with A_i the action space of agent i.
- $R(s_1, s_2, \dots, s_n, a_1, a_2, \dots, a_n)$: represents the joint reward function that maps the state-action pair of all agents to a real-valued reward.
- $P(P_1, P_2, \dots, P_n)$: represents the state transition probabilities for each agent, where each P_i is a function $S_i \times S_i \times A_i \to \mathbb{R}$.

The transition probabilities for joint states $(s_1, s_2, \dots, s_n)$ to $(s'_1, s'_2, \dots, s'_n)$ given a joint action $(a_1, a_2, \dots, a_n)$ are computed as:

$$P(s'_1, s'_2, \dots, s'_n | s_1, s_2, \dots, s_n, a_1, a_2, \dots, a_n) = \prod_{i=1}^{n} P_i\left(s'_i | s_i, a_i\right) \tag{8}$$

This formulation is scalable to an increased number of agents [46], making it well-suited for the multi-aircraft trajectory planning problem, where each aircraft operates independently but with a shared objective. The DEC-MDP framework effectively models the dynamic interactions between aircraft, the stochastic nature of weather, and the need for real-time decision-making. We provide details for each element of the DEC-MDP model with parameters defined in this work.

4.1.1. State space

State space captures all relevant information about the environment at a given time step. For our problem, the state s_t^i at time t for aircraft i includes:

- Aircraft positions: $p_i(t) = [x_i(t), y_i(t)\ ]$ represent their coordinates.
- Aircraft velocities: $v_i(t) = \left[v_{x_i}(t), v_{y_i}(t)\right]$ represent their speed components along each axis.
- Thunderstorm cells' information: $O^{\mathrm{W}}(t)$, which includes the centroid positions of the cells, representing dynamic obstacles.
- Remaining distance to exit waypoint: $d_t^{i\text{-Exit}}$ for each aircraft defined as $d_t^{i\text{-Exit}} = \|p_i(t) - w_i\|$

Thus, the state space for each aircraft i at time t is defined as:

$$s_t^i = \left(p_i(t), v_i(t), d_t^{i\text{-Exit}}, O^{\mathrm{W}}(t)\right) \tag{9}$$

4.1.2. Action space

The action of each aircraft is heading change, which is a continuous variable representing course adjustments. The action space for each aircraft i at time t is defined as: $\Delta h_i(t)$, which is within the range [-30, 30] degrees, denoted as:

$$a_t^i = \Delta h_i(t) \tag{10}$$

4.1.3. Reward function

The reward function is a critical component that guides the learning process by evaluating the immediate benefit of an action taken in a given state. In the context of multi-aircraft trajectory planning, the objective is to minimize total rerouting distance and heading change maneuvers while ensuring safety through several constraints. These objectives and constraints are incorporated into the reward function, which is defined for each agent i at time t as follows:

Separation assurance: $r_{i(t)}^{\mathrm{sepa}}$ is given as a negative reward if the distance between aircraft i and any other aircraft j falls below the minimum reparation distance $d_{\mathrm{min}}^{\mathrm{flight}}$. This penalizes unsafe proximity between aircraft.

$$r_{i(t)}^{\mathrm{sepa}} = \begin{cases} -1, \text{if } \ \|p_i(t) - p_j(t)\| < d_{\mathrm{min}}^{\mathrm{flight}}, \forall j \neq i \\ \qquad 0, \text{otherwise} \end{cases} \tag{11}$$

Dynamic thunderstorm cell avoidance: $r_{i(t)}^{\mathrm{W}}$ is given a negative reward if aircraft i has a loss of separation with a thunderstorm cell O^{W} at time t. This encourages the aircraft to avoid hazardous thunderstorm cells.

$$r_{i(t)}^{\mathrm{W}} = \begin{cases} -1, \text{if } \|p_i(t) - p_{O^{\mathrm{W}}}(t)\| < d_{\mathrm{min}}^{\mathrm{W}} + \mathrm{R}_{\mathrm{major\_axis}}^{\mathrm{W}}, \forall i \in \{1, \dots, n\}, \forall t \in T \\ \qquad 0, \text{otherwise} \end{cases} \tag{12}$$

Exit waypoint reachability: $r_{i(t)}^{\mathrm{exit}}$ is assigned a positive reward if the aircraft reaches its designated exit waypoint, encouraging the aircraft to rejoin its original air route.

$$r_{i(t)}^{\mathrm{exit}} = \begin{cases} 0, \text{if } p_i(t) = w_i \\ -1, \text{otherwise} \end{cases} \tag{13}$$

Heading change discouragement: $r_{i(t)}^{\mathrm{heading}}$ is a negative reward to minimize course change actions that subsequently reduce possible interactions and discomfort of passengers. Here h^{interval} is the interval of heading change, which acts as a normalization term making the reward in a set {-1, 0}.

$$r_{i(t)}^{\mathrm{heading}} = -\frac{\Delta h_i(t)}{h^{\mathrm{interval}}} \tag{14}$$

Rerouting distance minimization: $r_{i(t)}^{\mathrm{dist}}$ is a negative reward proportional to the ratio of the actual rerouting distance to the shortest distance.

$$r_{i(t)}^{dist} = -\frac{\|p_i(t) - w_i\|}{\|p_i(t_0) - w_i\|} \tag{15}$$

where $p_i(t)$ is the current position of aircraft i, $p_i(t_0)$ is the position of aircraft i at the start of rerouting (when it enters the affected airspace), and w_i is the position of its corresponding exit waypoint. The reward $r_{i(t)}^{\mathrm{dist}}$ falls in the range of [-1, 0].

Each of the five rewards now is normalized into [-1, 0], and their significance is adjusted by weights ω_{sepa}, ω_{W}, ω_{exit}, $\omega_{\mathrm{heading}}$, and ω_{dist}. Based on that, the total system reward r_t^{total} for all agents at time t is computed by summing the weighted individual rewards:

$$r_t^{\mathrm{total}} = \sum_{i=1}^{n} r_{i(t)} = \sum_{i=1}^{n} \left(\omega_{\mathrm{sepa}} r_{i(t)}^{\mathrm{sepa}} + \omega_{\mathrm{W}} r_{i(t)}^{\mathrm{W}} + \omega_{\mathrm{exit}} r_{i(t)}^{\mathrm{exit}} + \omega_{\mathrm{heading}} r_{i(t)}^{\mathrm{heading}} + \omega_{\mathrm{dist}} r_{i(t)}^{\mathrm{dist}} \right) \tag{16}$$

With the normalized rewards, fine-tuned weights are assigned for each reward based on the priority of tasks. Separation assurance task with other aircraft is the top priority and the ω_{sepa} is set to 10.

Avoidance of thunderstorm cells is given the second significance with weight $\omega_{\mathrm{W}} = 8$. Note that the weights ω_{sepa} and ω_{W} are given much larger values than the other to impose feasibility requirements. As frequent heading changes may increase airspace complexity, the weight $\omega_{\mathrm{heading}}$ is set to 1, while the efficiency cost of rerouting distance is given a weight of 0.5. Lastly, the weight ω_{exit} is set to 10 as the only positive goal reach reward to ensure that the rerouted aircraft can join remaining flight routes via their designated exit waypoints. These normalized rewards, along with their respective weights, guide the learning process in optimizing aircraft trajectories while adhering to safety and operational constraints.

The DEC-MDP framework effectively models the decentralized nature of multi-aircraft trajectory planning, allowing each agent to independently manage its flight path while contributing to a shared objective. The complexity of dynamic interactions, stochastic weather effects, and the need for real-time decision-making make deep reinforcement learning an appropriate solution.

4.2. Independent Deep Deterministic Policy Gradient (IDDPG) algorithm

In deep reinforcement learning, various algorithms cater to different challenges, each with distinct strengths. Among these, Deep Deterministic Policy Gradient (DDPG) excels in handling problems with large and continuous action spaces [47], making it particularly suitable for complex decision-making tasks like air traffic management (ATM), where continuous control is essential. It has demonstrated its effectiveness in single-agent scenarios within the ATM fields [32,48]. However, the decentralized Markov Decision Process (DEC-MDP) framework, which we model this problem, demands a multi-agent cooperative approach that standard DDPG, designed for single-agent environments, cannot fulfill. This challenge motivates us to adopt the Independent Deep Deterministic Policy Gradient (IDDPG) framework, tailored for multi-agent cooperation [49].

The goal of the IDDPG algorithm is to enable each aircraft agent to learn an optimal policy that maximizes its expected cumulative reward with an actor-critic architecture. Each agent i possesses an actor network $\mu_i(s_t^i|\theta^{\mu_i})$ that maps its current state s_t^i to a continuous action a_t^i (e.g., heading change). The policy network is parameterized by θ^{μ_i}, which is adjusted to improve decision-making based on observed states and received rewards. Concurrently, each agent has a critic network $Q_i(s_t^i, a_t^i|\theta^{Q_i})$ that evaluates the quality of the chosen action a_t^i by estimating the expected return (Q-value). The critic network is parameterized by θ^{Q_i}, which is updated to minimize the loss function defined by the difference between predicted Q-values and target Q-values derived from the environment's feedback.

Mathematically, the goal can be expressed as optimizing each agent's policy μ_i and value function Q_i to maximize the expected cumulative reward $\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \, r_{i(t)}\right]$, where γ is the discount factor. This optimization is performed through iterative updates of the policy and critic network parameters using gradient ascent and descend, and the policy gradient $\nabla_{\theta^{\mu_i}}\mathcal{J}(\mu_i)$ and loss function $\mathcal{L}(\theta^{Q_i})$ are calculated by:

$$\nabla_{\theta^{\mu_i}}\mathcal{J}(\mu_i)=\mathbb{E}_{\mathcal{D}}\left[\nabla_{a_t^i}Q_i(s_t^i,a_t^i|\theta^{Q_i})\nabla_{\theta^{\mu_i}}\mu_i(s_t^i|\theta^{\mu_i})\right] \tag{17}$$

$$\mathcal{L}(\theta^{Q_i})=\mathbb{E}_{\mathcal{D}}\left[\left(r_{i(t)}+\gamma Q_i'\left(s_{t+1}^i,\mu_i'(s_{t+1}^i|\theta^{\mu_i}{}')|\theta^{Q_i}{}'\right)-Q_i(s_t^i,a_t^i|\theta^{Q_i})\right)^2\right] \tag{18}$$

where $\mathcal{D}$ is the experience replay, μ_i' and Q_i' denote target actor and critic neural networks, which are parameterized by $\theta^{\mu_i}{}'$ and $\theta^{Q_i}{}'$.

Fig. 5 illustrates the architecture and training process of the IDDPG algorithm, with an emphasis on the integration of shared and individual network components. The training begins with the Experience Replay component, where observations from the simulated airspace environment, specifically state transitions (s_t, a_t, r_t, s_{t+1}), are stored in the replay buffer $\mathcal{D}$. This buffer plays a crucial role in stabilizing learning by allowing the algorithm to reuse past experiences during training. Mini-batches of these experiences $\mathcal{B}_{\text{size}}$ are sampled from the buffer to update the networks.

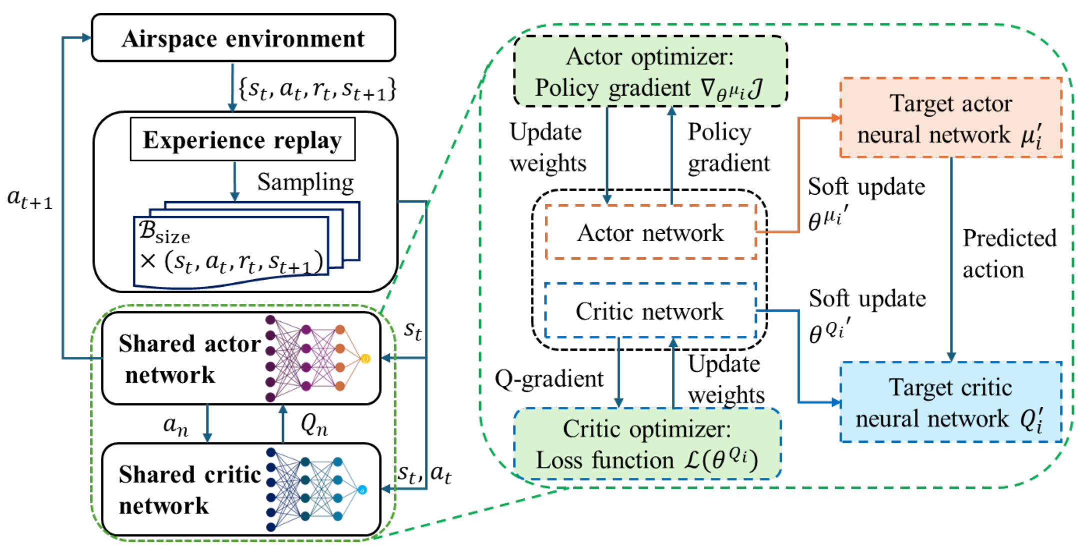


Fig. 5. Training process of the proposed IDDPG framework.

Central to the IDDPG framework are the shared actor and critic networks, which process the observation of individual agents that are sampled from the state inside experience replay. The shared actor network parameterized by θ^{μ} is responsible for generating actions a_n based on the state observations. These actions are then evaluated by the shared critic network, parameterized by θ^{Q}, which calculates the Q-value to measure the expected future reward given the current state and action.

This shared architecture ensures that all agents are learning from a consistent set of experiences, which helps mitigate the non-stationarity issue of the IDDPG algorithm [49]. The problem arises when each agent independently updates its own policy in response to the actions of others, this dynamic can lead to a constantly changing environment from the perspective of any single agent, causing difficulties in learning stable and optimal policies. This eventually leads to instability in the learning process and difficulty in value estimation, preventing convergence to a stable solution. However, by utilizing shared neural networks and experience replay, all agents essentially learn from a shared experience, ensuring consistency during the learning process. The shared network is updated based on observations from all agents, making the learning process more stable and less susceptible to the fluctuations caused by nonstationary.

To refine the networks, actor and critic optimizers are employed. The actor optimizer updates the parameters of the actor network using the policy gradient $\nabla_{\theta^{\mu_i}} \mathcal{J}$, which guides the network toward actions that maximize expected rewards. Simultaneously, the critic optimizer uses the loss function $\mathcal{L}(\theta^{Q_i})$ to adjust the critic network, improving its accuracy in estimating Q-values.

Additionally, target networks are incorporated to enhance the stability of the training process [50], as shown in **Fig. 5**. The target actor μ_i' and target critic Q_i' networks, which are slowly updated to track the main networks, serve as stable references during training. Both target networks are updated using a soft update mechanism where their parameters $\theta^{\mu_i}{}'$ and $\theta^{Q_i}{}'$ are gradually adjusted towards the current parameters of the main actor and critic networks. This slow update ensures a stable reference for calculating future values, which is essential for accurate and stable updates to the actor network, ultimately leading to a more robust training process.

The parameters in all four neural networks are updated as the training iteration progresses, and the actor neural network will ultimately provide a near-optimal action for the input observation. By integrating shared networks for consistency and target networks for stability, the IDDPG algorithm effectively addresses the challenges of multi-agent cooperative environments, enabling robust learning and decision-making in complex and dynamic airspace scenarios.

5. Simulation and Results

In this section, we conduct a comprehensive evaluation of our algorithm's performance. We start by describing the configurations of the simulation environment and algorithm hyperparameters used. Next, we analyze the training process of the proposed IDDPG learning framework, focusing on its convergence across different observation settings to determine the optimal configuration. The effectiveness of the algorithm is then validated in a real-world scenario selected from the Singapore Flight Information Region [13], followed by an assessment of its generalization capabilities using air route structures generated based on the common guidelines in airspace planning [51]. Finally, we evaluate the robustness of our model under diverse thunderstorm conditions.

5.1. Environment settings

The training scenarios were designed to simulate the complex interactions between airspace structures, aircraft, and thunderstorm cells within a 200 × 200 square nautical miles (nm^2) en route airspace. Each training episode involved resetting the airspace structure, with random generation of air routes defined by randomly selecting entry and exit waypoints with a minimum length of 100 nm. A total of five aircraft were randomly assigned to these routes, with a separation interval of 25 time steps (equivalent to 5 minutes) on the same route. All aircraft operate at the same flight level, maintaining a constant cruising speed of 400 knots. Aircraft may adjust their heading within a range of [-30, 30] degrees per time step for rerouting around thunderstorm cells. The minimum separation between aircraft is set at 5 nm. Two dynamic thunderstorm cells were introduced in this training setting, each with varying speed, direction, size, and shape. Their speed ranged from 50 to 90 knots, with direction varying depending on the scenario. For the thunderstorm cells, we consider ellipsoid shapes with a semi-major axis in the range from 15 to 30 nm, including the 5 nm for the separation buffer between the cell and the aircraft. The shape of these cells was updated every 5 time steps, with each time step lasting 12 seconds [17], and each simulation episode allowed for a maximum of 150 time steps (equivalent to 30 minutes).

Table 1. Simulation environment configurations.

Category	**Parameter**	**Value/description**
Airspace	Phase	En route
	Size of environment	200 nm × 200 nm
Aircraft	Speed	400 knots
	Altitude	Constant
	Heading change range	[-30, 30] degrees
	Separation minima	5 nm
Storm cell	Speed	[50, 90] knots
	Movement direction	[0, 359] degrees
	Size (radius)	[10, 25] nm
	Separation with aircraft	5 nm
	Update frequency	5 time steps
Duration	Time step	12 seconds
	maximum steps	150
Hardware	CPU	Intel i9-11900
	GPU	NVIDIA GeForce RTX 3080
	Memory	32GB
Software	Custom-built simulator in Python 3.9	

All training and testing simulations were executed on a hardware setup consisting of an Intel i9-11900 CPU, NVIDIA GeForce RTX 3080 GPU, and 32GB of memory, using a custom-built simulator developed in Python 3.9. **Table 1** provides a comprehensive summary of the environment configurations and **Table 2** presents the hyperparameter setting for the IDDPG algorithm used in this study.

Table 3 presents the performance metrics used to evaluate the safety and efficiency of the proposed method. Safety metrics include the aircraft loss of separation (LOS), defined as the ratio of LOS incidents between aircraft ($n_{\text{LOS\_aircraft}}$) to the total number of aircraft ($n_{\text{total\_aircraft}}$). The LOS rate with thunderstorm cells is defined as the ratio of LOS events with thunderstorm cells ($n_{\text{LOS\_storm}}$) to the total number of aircraft. Efficiency metrics include the goal reach rate, which represents the proportion of aircraft that successfully reach their intended destinations ($n_{\text{all\_reach}}$), and the flight distance ratio, which compares the actual flight distance to the planned route distance, indicating the efficiency of the planning process.

Table 2. Hyperparameter settings for IDDPG algorithm.

Parameter	**Value**
Minibatch size	512
Reply buffer size	100000
Actor learning rate	0.0001
Critic learning rate	0.0001
Discount factor	0.95
Number of training episodes	20000
Soft update rate	0.01
Target network update frequency	1
Maximum play per episode	200
Noise level	1->0.03

Table 3. Performance Metrics.

Metric	**Description**
Aircraft loss of separation rate	$n_{\text{LOS\_aircraft}}/n_{\text{total\_aircraft}}$
Thunderstorm loss of separation rate	$n_{\text{LOS\_storm}}/n_{\text{total\_aircraft}}$
Goal reach rate	$n_{\text{all\_reach}}/n_{\text{total\_aircraft}}$
Flight distance ratio	Actual distance/planned route distance

5.2. Model training results

Based on the settings detailed in Section 5.1, we conducted model training to evaluate the effectiveness of various input states in the Independent Deep Deterministic Policy Gradient (IDDPG) algorithm. The choice of state observations fed into the neural network is crucial for training a robust and efficient model. To identify the optimal observation configuration, we compared several representative state inputs within the IDDPG framework.

The four primary types of observations considered were: (i) Aircraft's own state: This includes position, speed, heading, and other relevant information about the aircraft itself; (ii) Weather radar sensor data: This includes information from the onboard weather radar, detecting nearby thunderstorm cells and feeding this data into the neural network; (iii) Other aircraft's state: This includes the state information (e.g., position, speed, heading) of other nearby aircraft, combined with the aircraft's own state data; (vi) ADS-B sensor data: This includes information obtained from ADS-B (Automatic Dependent Surveillance–Broadcast) sensors, which provide the state of nearby aircraft as perceived through the sensor.

Increased input information can enhance the learning capability of the model by providing more comprehensive situational awareness. However, this also increases the complexity of the neural network, potentially reducing learning efficiency and leading to the inclusion of redundant or irrelevant information, which may degrade the model's overall performance. Therefore, determining an optimal balance of input information is essential for improving the effectiveness and efficiency of the neural network model.

To investigate the impact of different input configurations, we defined four variations of the training algorithms, including IDDPG: Inputs only the aircraft's own state and weather radar sensor data. This serves as the baseline model. IDDPG-n: Inputs the aircraft's own state, weather radar sensor data, and direct state information of other aircraft. The "n" denotes the inclusion of other aircraft's state information directly into the neural network. IDDPG-s: Inputs the aircraft's own state, weather radar sensor data, and ADS-B sensor data for nearby aircraft. The "s" denotes the inclusion of other aircraft's information perceived through sensors like ADS-B. IDDPG-ns: Combines all the inputs from the previous variations—aircraft's own state, weather radar sensor data, other aircraft's state, and ADS-B sensor data. This variation represents the most comprehensive state input configuration.

It is important to note that all variations were tested under identical conditions regarding the number of aircraft and thunderstorm cells, origin-destination pairs, and movement trajectories to ensure a fair comparison. Additionally, variations in air route length across different episodes may cause fluctuations in reward curves. Episodes with longer routes require more time steps to reach the destination, leading to a higher accumulation of negative rewards due to time-step penalties.

The obtained training curves and convergence results for the four IDDPG variations are shown in **Fig. 6**. The baseline IDDPG model (red line), which only includes the aircraft's own state and weather radar sensor data, shows the slowest convergence and the lowest final reward, indicating its limited ability to handle complex multi-aircraft scenarios. Introducing other aircraft's state information directly into the neural network training in IDDPG-n (blue line) improves performance, with faster convergence and higher final rewards, demonstrating that direct awareness of other aircraft in the neural network enhances conflict avoidance. However, this improvement plateaus indicating challenges in scalability under higher traffic densities. The IDDPG-s model (purple line), which incorporates other aircraft's states and weather radar information into sensor data as input, delivers the best performance with the highest rewards. This model's ability to adapt flexibly to dynamic situations by leveraging real-time input from the sensor's state makes it the most robust and scalable for multi-aircraft trajectory planning under rapidly evolving thunderstorm conditions. In contrast, the IDDPG-ns model (green line), which combines all input sources, fails to outperform IDDPG-s, as the added complexity may lead to inefficiencies and overfitting, reducing its overall effectiveness.

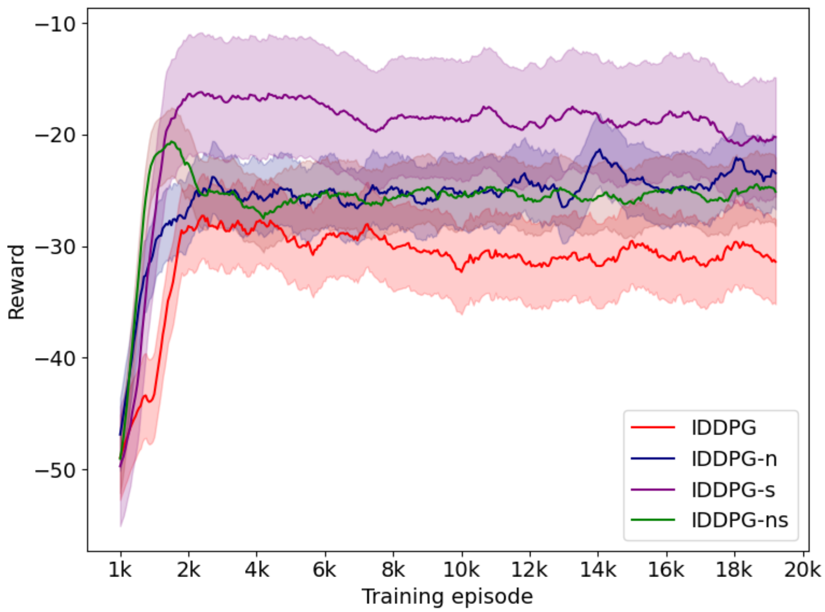


Fig. 6. Training curves for different state variations of the IDDPG algorithm.

The training results underscore the significance of carefully selecting state observations for deep reinforcement learning models in dynamic, complex airspace environments. While adding more input information can improve the model's performance, it also increases the complexity of the neural network, which might introduce inefficiencies or overfitting. The IDDPG-s variation, which balances the input of sensor data with state information, demonstrates the best performance, making it the optimal choice for real-time multi-aircraft trajectory planning under diverse conditions. Its performance is tested in the following sections.

5.3. Effectiveness and scalability in real-world scenarios from Singapore FIR

In this section, we evaluate the effectiveness and scalability of the trained IDDPG-s in a real-world en route airspace within the Singapore Flight Information Region (FIR), as depicted in **Fig. 7**. This area is situated within the oceanic portion of the FIR, where radar coverage and ATC services are unavailable. The multi-agent method proposed in this study is particularly critical for autonomously managing multi-aircraft co-operative trajectory planning, especially in the context of future high-density traffic and increased convective weather conditions.

We selected a 200×200 nm airspace within this FIR that includes three major air routes: N884, M767, and M768, intersecting at two critical waypoints, LAGOT and TODAM. This configuration represents one of the most complex traffic scenarios in the region. Each air route includes designated entry and exit waypoints, such as $w_{\mathrm{N884}}^{\mathrm{in}}$ and $w_{\mathrm{N884}}^{\mathrm{exit}}$ for route N884. Simulated aircraft traffic enters the airspace through the entry waypoints and exits via the corresponding exit waypoints to continue their remaining flights. A minimum longitudinal separation of 10 minutes is typically applied on these routes [13]. In this work, we reduce the separation to 5 minutes (25 time steps) to test our model in a doubled-capacity scenario. Four aircraft are randomly introduced from three entry waypoints, reflecting state-of-the-art simulation scenarios, which include four aircraft within a 200×200 nm area over a 30-minute interval [28]. Additionally, two dynamic thunderstorm cells with varying speed, size, and shape are simulated, moving from northwest to southeast across the region.

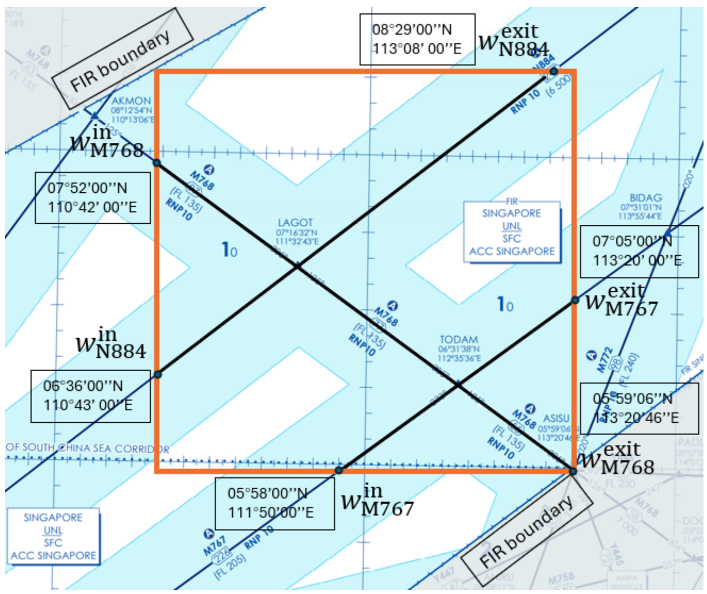
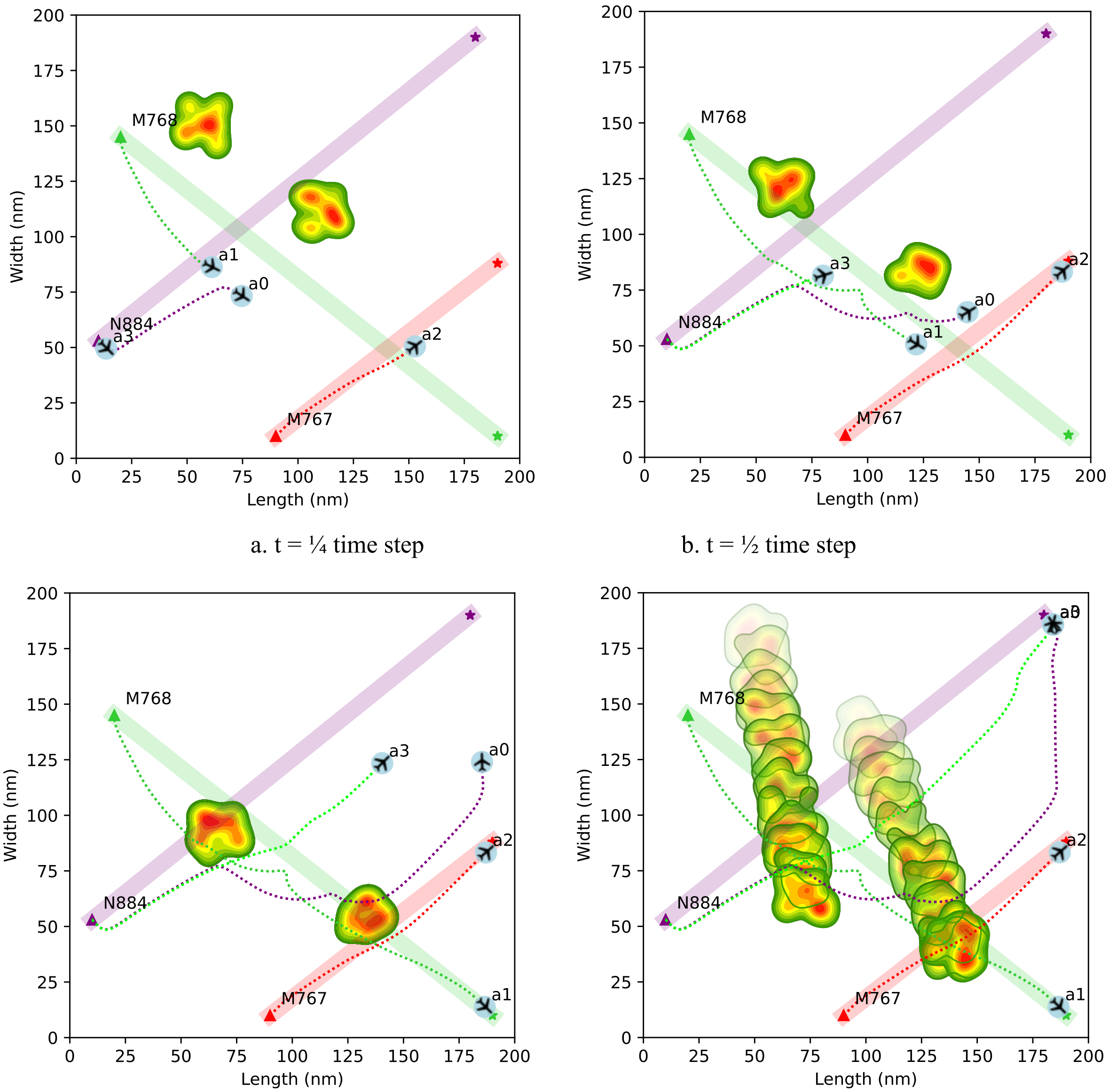


Fig. 7. A real-world en route airspace area selected from Singapore FIR for model testing.

Fig. 8 presents an example of simulation results with conflict-free trajectories for four aircraft (labeled as a0, a1, a2, and a3) navigating through the considered en route airspace under the influence of two dynamic thunderstorm cells (represented as heatmap contours). The transparency of the contours decreases over time to indicate their temporal evolution. More specifically, this figure provides four screenshots of the simulation at ¼ time step, ½ time step, ¾ time step, and the final step. The air routes (M767, M768, and N884) are marked as shaded bands with their respective entry and exit waypoints represented by solid triangles and stars, respectively. The trajectory flown by each aircraft is represented by a dashed line, with the paths changing dynamically in response to the evolving positions of the thunderstorm cells.

Results show that all aircraft successfully navigate around the thunderstorm cells while maintaining safe separation distances from both the storms and other aircraft. Initially, the aircraft follow their predetermined paths along the air routes, as seen in **Fig. 8**(a). As the thunderstorm cells move and change shape, the aircraft adjust their headings to avoid these hazards. This avoidance behavior is evident in **Fig. 8**(b) and **Fig. 8**(c), where the aircraft begin to deviate from their original paths to ensure they maintain a safe distance from the thunderstorm cells. **Fig. 8**(d) represents the final step of the simulation, and the trajectories show that the aircraft have effectively rerouted around the thunderstorm cells and are on course to safely reach their respective exit waypoints.



a. t = ¼ time step

b. t = ½ time step

c. t = ¾ time step

d. t = final step

Fig. 8. Conflict-free trajectories of multiple aircraft and moving thunderstorm cells at different time steps. Note that the enter and exit waypoints are represented by the symbols of solid triangle and star, respectively. The dynamics of the thunderstorm cells are depicted using evolving contours, with decreasing transparency to represent their progression over different time steps.

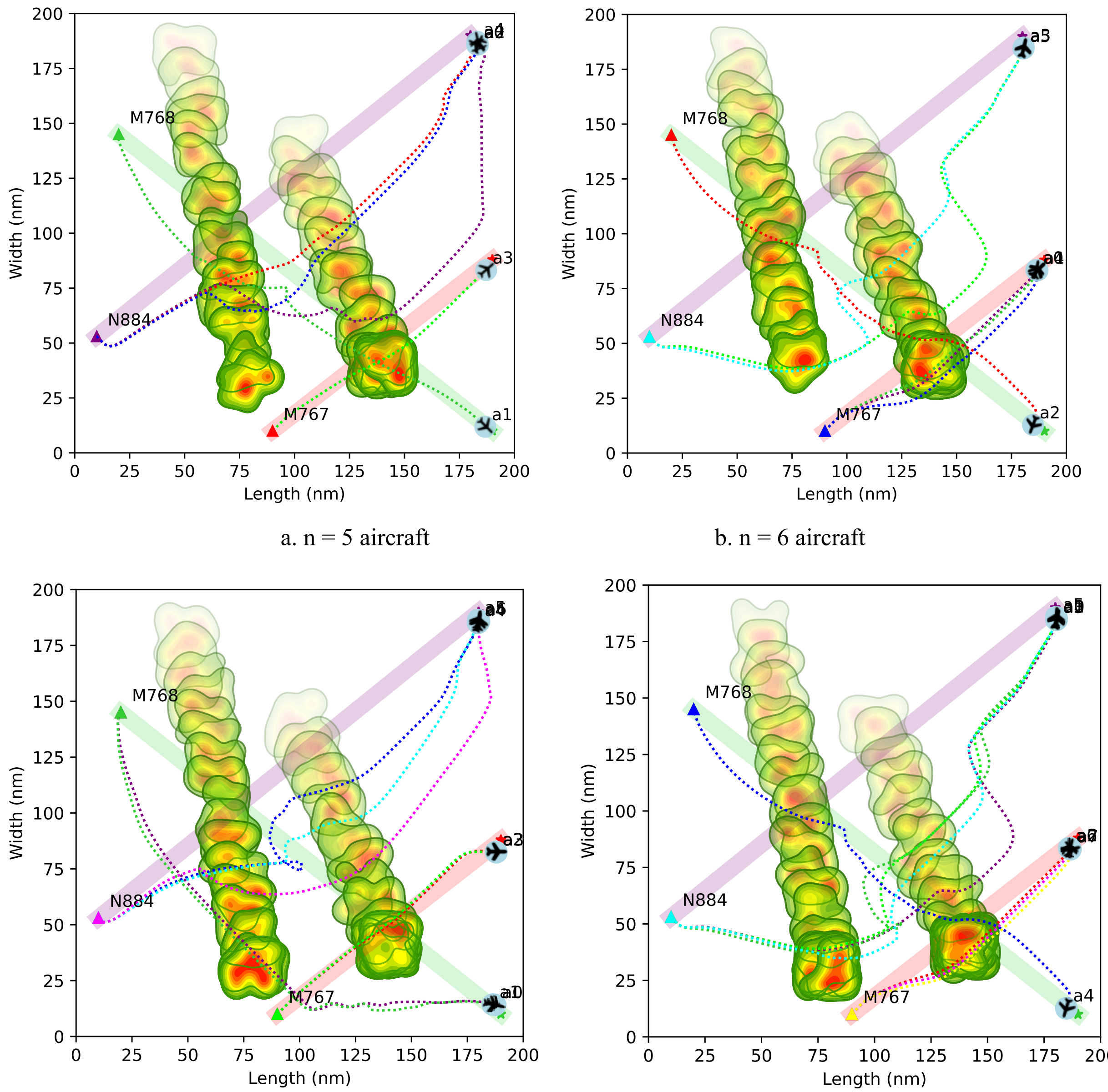
These dynamic adjustments are driven by our reward function, which penalizes proximity to thunderstorm cells and other aircraft, thereby encouraging the model to find safe and efficient alternative paths. The state inputs, which include information from onboard sensors (such as ADS-B data for other aircraft and radar data for thunderstorm cells), allow the model to perceive the evolving environment in real-time and make informed decisions on trajectory adjustments. The obtained results demonstrate the effectiveness of the proposed multi-agent deep reinforcement learning framework in handling complex, dynamic airspace scenarios. The aircraft can reroute in real-time, avoiding LOS with both thunderstorm cells and other aircraft, while still adhering to their flight objectives.

To evaluate the scalability of the proposed model, we conducted additional simulation tests with increasing numbers of aircraft (5, 6, 7, and 8) under the same dynamic weather conditions. Each scenario was tested 100 times, with aircraft randomly entering the airspace from three different waypoints, simulating a variety of traffic arrival processes. This setup allowed us to assess how well the model performs as airspace becomes more congested and the complexity of the traffic scenario increases.

In **Fig. 9**, we present conflict-free trajectories for varying numbers (n = 5, 6, 7, and 8) under the influence of moving thunderstorm cells. Each sub-figure represents a different traffic density scenario. The aircraft are labeled as a1 to a8 depending on the number of aircraft in each subfigure. Each aircraft adjusts its trajectory to avoid both other aircraft and the thunderstorm cells. In particular, aircraft like a5 in each scenario deviates significantly from the original routes to maintain safe distances. In some scenarios, there appear to be sharp turns and holding patterns in the trajectories (e.g., a5 (blue dotted line) in the 7-aircraft scenario as shown in **Fig. 9**(c)). These behaviors are necessary due to the proximity of multiple moving obstacles (both other aircraft and a dynamic thunderstorm cell). The system resolves these potential conflicts with preemptive maneuvers, resulting in sharp course changes to avoid separation losses.

A detailed statistical result from 100 test runs for each traffic density scenario is provided in **Table 4**. Results show that the LOS rate with other aircraft remained consistently low across all scenarios, with a slight increase to 1% in the scenario with 7 aircraft. However, the LOS rate with thunderstorm cells increased with the number of aircraft, reaching 5% in the scenario with 8 aircraft. This can be attributed to the higher weight assigned to aircraft conflicts compared to thunderstorm cell conflicts in the reward function, as defined in subsection 4.1.3. In such congested airspace, the model prioritizes avoiding aircraft conflicts, which inadvertently increases the likelihood of thunderstorm cell conflicts.

In terms of goal reach rate, there was a slight decrease as the number of aircraft increased. While the goal reach rate was 100% for 4 aircraft, it dropped to 95% when 8 aircraft were involved. This slight decline shows the growing difficulty in ensuring that all aircraft can safely navigate to their destinations, as the airspace becomes more crowded. The flight distance ratio, which compares the actual flight distance to the planned route distance, also showed slight increases with the number of aircraft (see **Table 4**). This reflects the additional maneuvers required to avoid conflicts, particularly with thunderstorm cells. However, despite these increases, the ratios remained within an acceptable range, with a mean of 1.17 and a standard deviation of 0.17 for the most dense traffic scenario, indicating that the model continues to manage rerouting efficiently, even as traffic density increases, as shown in **Fig. 10**.



a. n = 5 aircraft

b. n = 6 aircraft

c. n = 7 aircraft

d. n = 8 aircraft

Fig. 9. Conflict-free trajectories of varying numbers of aircraft under moving thunderstorm cells.

The scalability tests demonstrate that our proposed model effectively handles multi-aircraft trajectory planning under dynamic weather conditions, even as the number of aircraft increases. Notably, our model maintains low LOS rates with other aircraft, demonstrating its capability to ensure safe separation in dense traffic with only a small and acceptable efficiency loss on rerouting distance.

Table. 4. Scalability analysis with an increased number of aircraft in 100 test runs.

Performance metrics	Number of aircraft				
	4	5	6	7	8
Aircraft LOS rate	0	0	0	1%	0
Thunderstorm LOS rate	0	2%	2%	3%	5%
Goal reach rate	100%	98%	98%	96%	95%
Flight distance ratio (Mean/standard deviation)	1.08/0.06	1.17/0.15	1.17/0.16	1.16/0.17	1.17/0.17

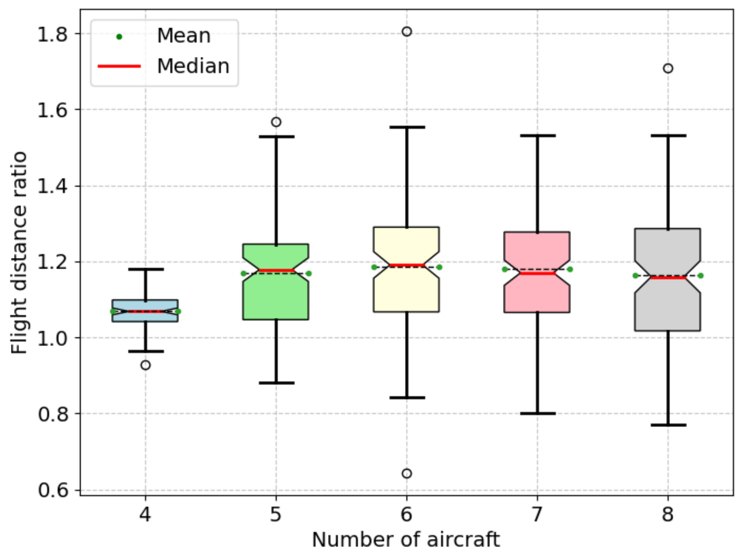


Fig. 10. Flight distance ratios remained within an acceptable range with the increased traffic density.

5.4. Generalization across complex air route structures

In this section, we evaluate the generalization ability of our trained model across complex airspace structures and weather patterns. Two representative air route configurations (ARS-1 and ARS-2), generated based on EuroControl guidelines [51], are used to test the model under varying traffic densities, with aircraft numbers increasing from 4, 6, to 8. Additionally, the thunderstorm cells in these simulations are larger, with a radius of 25 nautical miles, and move in different directions. These settings aim to assess the model's robustness and adaptability in managing multi-aircraft trajectory planning under more diverse and challenging conditions compared to the real-world scenario previously analyzed. The obtained results are presented in **Fig. 11** and **Table 5**.

The results indicate that the model consistently maintained a zero LOS rate with other aircraft across all tested scenarios, regardless of the air route structure or the number of aircraft. This highlights the model's ability to ensure safe separation between aircraft, even in highly congested airspace. However, the LOS rate with thunderstorm cells slightly increased as the number of aircraft rose, reaching 2% in ARS-1 with 8 aircraft and 1% in ARS-2 with 8 aircraft. This increase suggests that while the model effectively prioritizes avoiding LOS between aircraft, the added complexity of more aircraft and larger thunderstorm cells slightly compromises its ability to avoid weather-related hazards.

The goal reach rate remained high across all scenarios, with 100% in scenarios with 4 and 6 aircraft, and a slight decrease to 98% or 99% with 8 aircraft. The mean distance ratio ranged from approximately 1.19 to 1.24, indicating that while the model generally maintains efficient rerouting, the presence of larger thunderstorm cells and increased traffic density require longer detours. Notably, the ARS-2 configuration exhibited slightly lower flight distance ratios than ARS-1, suggesting that this air route structure might offer more efficient paths under the given conditions.

Overall, the results from the generalization testing affirm the model's effectiveness in adapting to new and complex air route structures while maintaining high levels of safety and efficiency. This demonstrates the potential application of our model in different airspace structures, reducing the need for redevelopment and retraining of the DRL model.

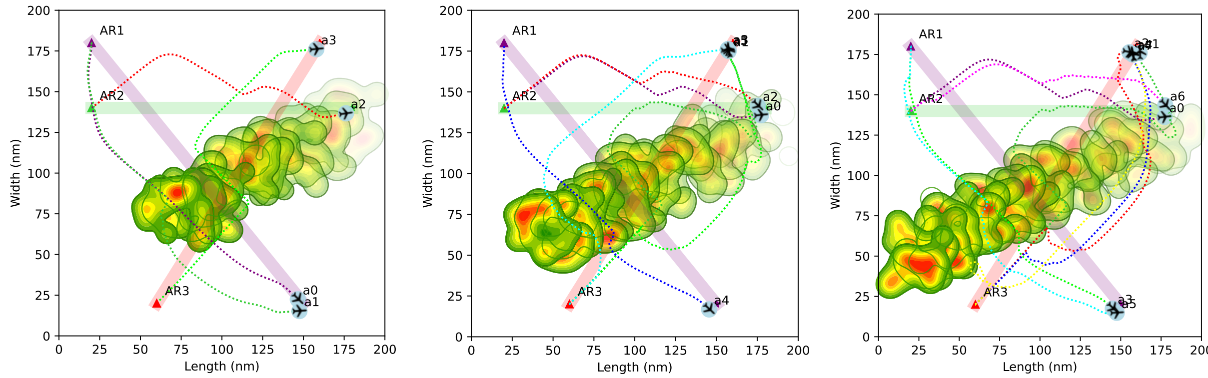
5.5. Algorithm comparisons with Fast Marching Tree (FMT) and single-agent Deep Deterministic Policy Gradient (DDPG)

In this subsection, we compare our proposed decentralized multi-agent IDDPG algorithm with a FMT algorithm and a single-agent DDPG model. We used the FMT algorithm as one baseline for comparison, as it represents the state-of-the-art approach for aircraft rerouting under thunderstorm constraints [28]. While the single-agent DDPG represents the state-of-the-art reinforcement learning algorithm in aircraft conflict resolution [32] without considering thunderstorms. To ensure a fair comparison, we adopted the single-agent DDPG structure to address the problem setting in this study.

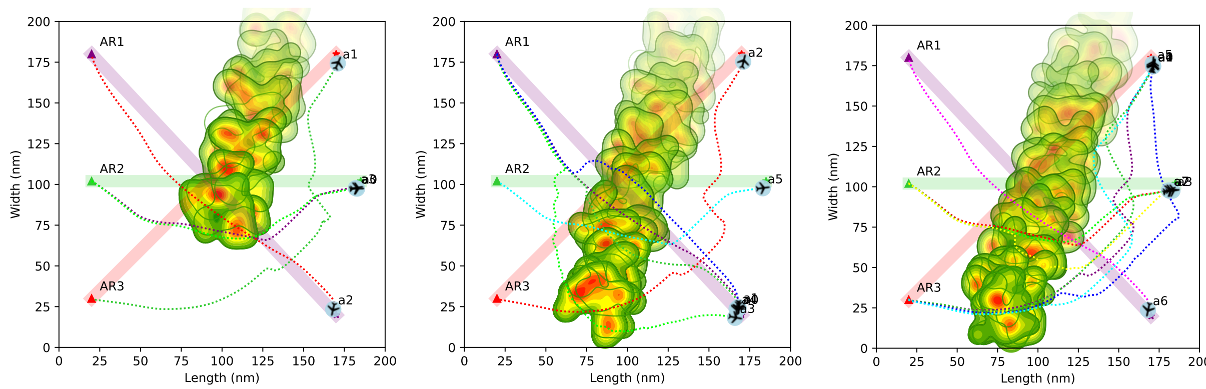
(i). Comparison with FMT algorithm

The Fast Marching Tree algorithm is a sampling-based path-planning method that incrementally builds a tree of feasible trajectories by connecting sampled waypoints to minimize costs, such as distance or risk while avoiding obstacles like thunderstorms. It efficiently searches for solutions by leveraging pre-sampled nodes and fast propagation techniques. We compare it with the proposed IDDPG model.

The comparison is conducted using the ARS-2 airspace structure, with increasing traffic density scenarios involving 4, 6, and 8 aircraft. Each simulation includes one dynamic thunderstorm cell with varying, but predictable, trajectories. The other settings remain consistent with those described in Section 5.4. In the FMT algorithm, 2000 sampling nodes are used to generate potential paths for trajectory planning. A local neighborhood radius of 10 nm is defined, within which an aircraft can evaluate nearby nodes for path expansion. During trajectory planning, each aircraft considers both other aircraft and dynamic thunderstorm cells as obstacles, ensuring safe separation and efficient navigation around weather disturbances while reaching its designated destination. 100 test runs were conducted for both algorithms and their performance is evaluated based on defined performance metrics. An illustration of planned trajectories by FMT algorithm with different aircraft is presented in **Fig. 12**, while detailed metrics achieved by FMT are provided in **Table 6**.

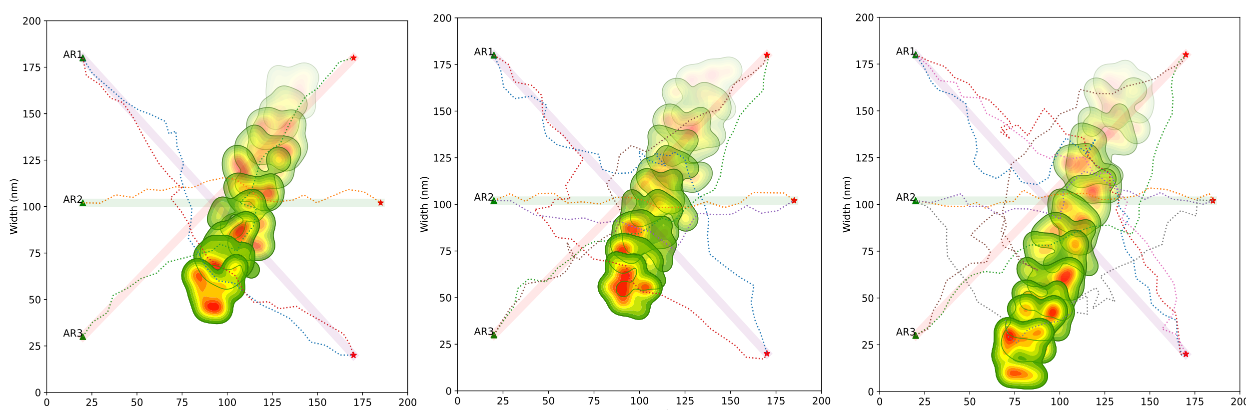


a. Air route structure-1 (ARS-1) with increased number of aircraft from 4, 6, to 8.



b. Air route structure-2 (ARS-2) with increased number of aircraft from 4, 6, to 8.

Fig. 11. Generalization testing of proposed IDDPG algorithm in two complex and representative air route structures with increased traffic density.



a. 4 aircraft b. 6 aircraft c. 8 aircraft

Fig. 12. Planned trajectories using Fast Marching Tree (FMT) algorithm in ARS-2 airspace structure with increased traffic density from 4, 6, to 8 aircraft. Note that the red circle represents thunderstorm cells, which move from northeast down to southwest in the environment.

The comparison between the proposed IDDPG algorithm and the Fast Marching Tree (FMT) algorithm highlights distinct advantages in handling multi-aircraft trajectory planning under dynamic thunderstorm conditions.

When observing **Fig. 11**(b) (IDDPG) and **Fig. 12** (FMT), one major difference lies in trajectory planning. The FMT algorithm frequently results in unrealistic sharp turns and zigzag paths (e.g., sudden 180-degree changes), which are unsuitable for actual aircraft performance. This limitation is especially evident as traffic density increases, where the FMT algorithm produces increasingly complex and zigzag trajectories. In contrast, the IDDPG algorithm generates smooth and natural trajectories, even with increased traffic densities. This difference suggests that IDDPG handles complex dynamic environments more efficiently, leading to more realistic flight paths.

Another observation is about traffic flow organization. The FMT algorithm simply generates avoidance actions without a sense of re-organizing the traffic flow. This lack of coordination leads to numerous trajectory intersections and conflict hotspots (see **Fig. 12**(c)), significantly increasing airspace complexity and the risk of mid-air collisions. On the other hand, IDDPG's cooperative decision-making strategy produces more organized traffic flows, which reduces the number of conflicting trajectory intersections (see **Fig. 11**(b)). This highlights the ability of IDDPG to manage not just conflict avoidance but also to coordinate traffic in a way that mitigates future risks.

The results from **Table 5** (IDDPG in ARS-2) and **Table 6** (FMT in ARS-2) further illustrate key performance differences. Loss of Separation (LOS) rates are notably lower with IDDPG. While IDDPG achieves a 0% aircraft LOS rate even under dense traffic conditions (up to 8 aircraft), the FMT algorithm's LOS rate increases significantly, reaching up to 17% with 8 aircraft. This highlights the capability of IDDPG in maintaining safe separation between aircraft, especially when dealing with dynamic thunderstorm cells.

Additionally, the avoidance threshold used by the FMT algorithm affects its performance. Smaller thresholds (e.g., 20 nm) result in faster computation time (3.36 s) but lead to higher LOS rates (7%). In contrast, larger thresholds (e.g., 40 nm) slightly lower the LOS rate (6%) but increase the computation time (8.32s). IDDPG does not require a fixed avoidance threshold; instead, it dynamically adjusts avoidance actions based on the current scenario, offering more flexibility and better efficiency. This adaptability allows the IDDPG to consistently ensure separation without being constrained by the computational overhead of threshold settings.

Lastly, computational efficiency is another major differentiator. As traffic density increases from 4 to 8 aircraft, the average computation time of FMT at each time step grows significantly (from 5.04s to 41.29s at a 30 nm threshold). The growing computational burden is a result of repeated recalculations as aircraft must recheck their trajectories in response to dynamic changes in the environment. IDDPG, on the other hand, maintains better scalability and is more computationally efficient (less than 20 milliseconds per solution), making it more suitable for time-sensitive applications.

Table. 5. Generalization analysis of proposed IDDPG algorithm in diverse air route structures under 100 test runs.

Performance metrics (Average)	**Air route structures**					
	ARS-1			ARS-2		
Number of AC in each testing	4	6	8	4	6	8
Aircraft LOS rate	0	0	0	0	0	0
Thunderstorm LOS rate	0	1%	2%	0	0	1%
Goal reach rate	100%	99%	98%	100%	100%	99%
Flight distance ratio (Mean/standard deviation)	1.23/0.16	1.24/0.18	1.24/0.19	1.19/0.12	1.21/0.15	1.24/0.18

Table. 6. Performance analysis of the Fast Marching Tree (FMT) algorithm in ARS-2 scenarios under 100 test runs.

Number of aircraft	4			6	8
Avoidance threshold (nm)	20	30	40	30	30
Aircraft LOS rate	7.00%	6.00%	6.00%	12.00%	17.00%
Thunderstorm LOS rate	1.00%	1.00%	1.00%	1.00%	1.00%
Goal reach rate	92%	93%	93%	87%	82%
Flight distance ratio (Mean/standard deviation)	1.13/0.02	1.16/0.03	1.26/0.07	1.25/0.14	1.28/0.20
Computation time* (s)	3.36	5.04	8.32	18.99	41.29

* Computation time measures the average time needed for trajectory planning at each time step.

In conclusion, IDDPG demonstrates superior performance compared to FMT in terms of trajectory smoothness, safety (lower LOS rates), and scalability in real-time dynamic airspace scenarios. While FMT performs well under static conditions [28], it may struggle to manage the complexities and unpredictability of dynamic weather and increasing traffic, where the proposed DRL-based IDDPG excels.

(ii). Comparison with the centralized single-agent DDPG model

We first present the state definitions and network set up for IDDPG and single-agent DDPG models. The proposed IDDPG employs centralized training and decentralized execution (CTDE) paradigm [52,53]. During training, a shared actor-critic network uses combined experiences from all aircraft to optimize joint behavior. During execution, each aircraft acts independently based on its own observations, with coordination ensured through the shared policy. The single-agent DDPG model uses a centralized approach, where a single network controls all aircraft. Actor, critic, and experience replay are shared, with inputs comprising all aircraft observations. Training and execution use the same centralized state. We conducted 100 independent evaluations for both the decentralized multi-agent IDDPG and centralized single-agent DDPG models, with results presented in **Fig. 13**. The IDDPG consistently outperforms the single-agent DDPG in most key metrics as the number of aircraft increases, demonstrating its advantages in scalability and robustness.

In terms of goal reach rate (**Fig. 13**(a)), the IDDPG maintains a high success rate above 92% even with 10 aircraft, whereas the single-agent DDPG shows a significant decline, dropping below 50% when handling 10 aircraft. This highlights the ability of decentralized IDDPG to handle increasing complexity and maintain operational efficiency. For loss of separation rates, both for aircraft and thunderstorms (**Fig. 13**(b) and **Fig. 13**(c)), the single-agent DDPG has significant performance issues. The aircraft LOS rate fluctuates and reaches up to 4%, while the thunderstorm LOS rate sharply increases to 49% with 10 aircraft. In contrast, IDDPG maintains much lower and more stable rates in both metrics, with aircraft LOS rates never exceeding 2% and thunderstorm LOS rates remaining below 10%. This demonstrates the robustness of IDDPG in ensuring safety and adaptability under increasing airspace demands and dynamic storm conditions. However, for flight distance ratio (**Fig. 13**(d)), the single-agent DDPG achieves lower mean values and smaller standard deviations compared to IDDPG, reflecting its centralized policy's ability to provide better cooperative decisions from a global perspective. IDDPG shows higher variance and longer distances due to its decentralized execution, where coordination between agents is inherently weaker than in a centralized method.

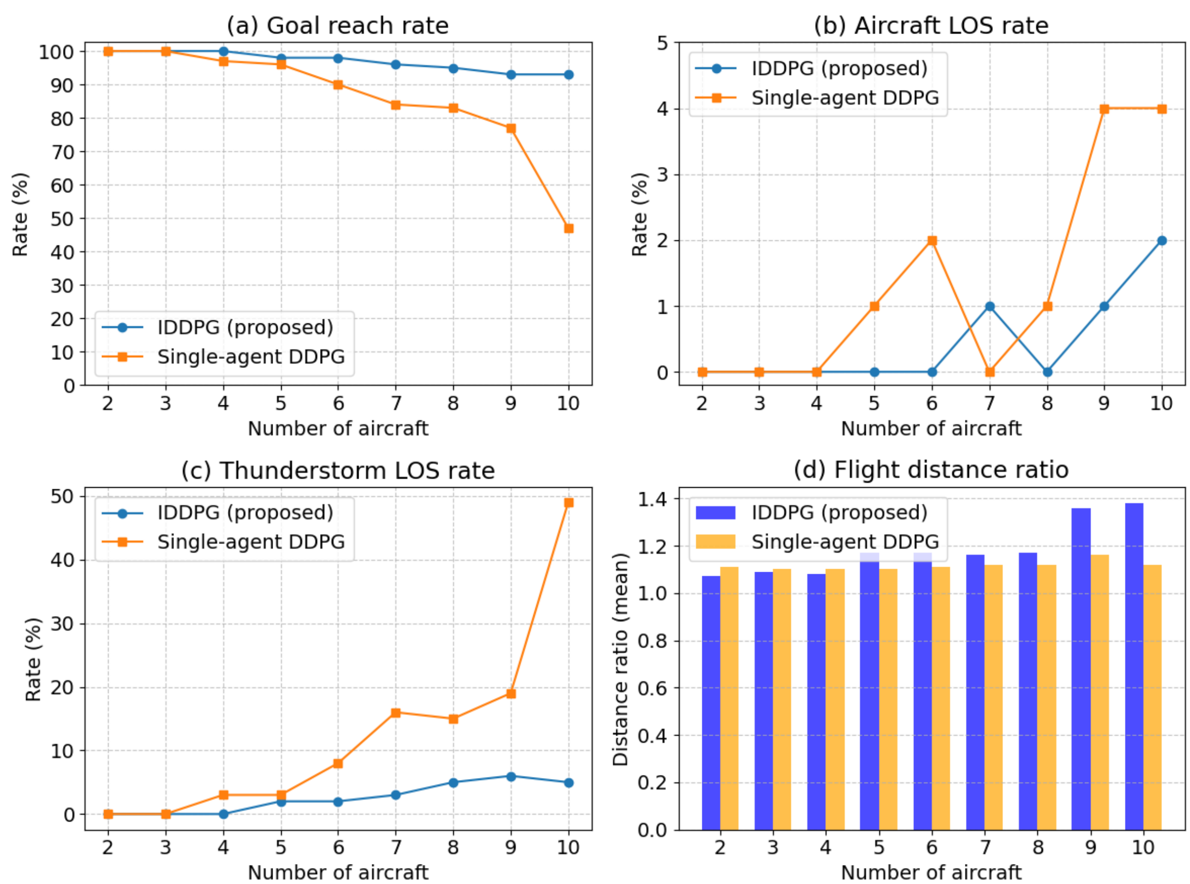


Fig. 13. Comparisons between decentralized multi-agent IDDPG and centralized single-agent DDPG.

Overall, if scalability and system robustness are not concerns, the single-agent DDPG may be a suitable choice for applications where distance minimization is a priority. However, for scenarios requiring scalability and resilience, such as managing large airspaces with many aircraft under thunderstorms, the decentralized decision-making approach of IDDPG is more robust, ensuring better overall performance under dynamic and complex conditions.

5.6. Robustness under diverse weather conditions

In this section, we demonstrate the robustness of our multi-aircraft trajectory planning model under varying weather complexities. These complexities are defined by different sizes, numbers, and moving trajectories of thunderstorm cells. Six distinct weather scenarios are created, each with varying levels of complexity: from a single small thunderstorm cell (15 nm radius) to scenarios with three large thunderstorm cells (25 nm radius). These scenarios are named accordingly (e.g., c_1S for one small cell, c_3L for three large cells). To quantify the severity of each scenario, the percentage of airspace covered by thunderstorm cells is calculated, with coverage ranging from 1.77% to 14.73%, as shown in **Table 7**.

Overall, results indicate that as the weather complexity increases, the LOS rates with thunderstorm cells rise accordingly. For example, in the scenario with three large thunderstorm cells (c_3L), which cover 14.73% of the airspace, the LOS rate with thunderstorm cells peaks at 34%, detailed in **Table 7**. Conversely, LOS rates with other aircraft remain low across all scenarios, with a maximum of 4% in the most severe weather conditions. Goal reach rates decrease as thunderstorm cell coverage increases, dropping from 94% in the least severe scenario (c_1S) to 63% in the most severe (c_3L). The flight distance ratio, which measures the efficiency of the rerouting, also increases with weather severity, with the highest mean ratio of 1.29 observed in the c_2L scenario. Despite the increased weather severity, the model maintains relatively low LOS rates with other aircraft, ensuring safety in highly congested airspace. However, the rising LOS rates with thunderstorm cells and decreasing goal reach rates indicate that the model's performance is challenged as environmental complexity increases.

Another notable observation is the significant impact of random trajectories of thunderstorm cells. In Section 5.3, where thunderstorm cell trajectories were predicted, the LOS rate with thunderstorm cells was 0%. In contrast, this section shows a notable increase in the LOS rate with thunderstorm cells, exceeding 10% even with small cells, and reaching 34% in the scenario with three large cells, although the LOS rate between aircraft remained low across all scenarios. This observation suggests that the model performs better with predicted thunderstorm cell trajectories, indicating a potential area for future research on more advanced prediction models.

Overall, these findings demonstrate the effectiveness of the proposed mode under moderate conditions while highlighting areas for further refinement, particularly in scenarios involving unpredictable and severe weather patterns. Future work should focus on integrating thunderstorm cell trajectory prediction into the model to further reduce LOS rates and improve the safety and efficiency of air traffic management in thunderstorm weather scenarios.

Table 7. Simulation setting and testing results under diverse weather scenarios.

Size of thunderstorm cell	Small (15 nm)			Large (25 nm)		
Number of thunderstorm cell	1	2	3	1	2	3
Name of scenario	c_1S	c_2S	c_3S	c_1L	c_2L	c_3L
Percentage of coverage	1.77%	3.54%	5.31%	4.91%	9.82%	14.73%
Aircraft LOS rate	1%	0%	3%	4%	4%	3%
Thunderstorm LOS rate	5%	12%	11%	10%	24%	34%
Goal reach rate	94%	88%	86%	86%	72%	63%
Flight distance ratio (Mean/standard deviation)	1.14/0.28	1.22/0.60	1.26/0.76	1.20/0.66	1.29/0.92	1.25/0.90

5.7. Impact of estimated time of arrival (ETA) uncertainty on model performance

Uncertainties in ETA may significantly disrupt pre-planned separations between aircraft on the same air route, which can lead to an increased rate of loss of separation at entry waypoints [54]. To address this, we conducted numerical simulations to evaluate the impact of varying levels of ETA uncertainty on model performance. The simulation environment utilizes the Singapore FIR airspace and air route structure, as defined in Section **5.1**. Each simulation batch consists of 30 aircraft, with a pre-planned time separation of 10 minutes between consecutive entries [13]. ETA uncertainty is introduced by adding sampled values to the pre-planned ETA for each aircraft. Based on the literature [55], ETA uncertainty follows a normal distribution and is classified into four levels in this study: negligible, low, medium, and high levels.

The negligible level assumes normal operations with accurate onboard trajectory prediction systems, leading to minimal deviations [55]. The high level represents severe ETA deviations caused by unpredictable trajectory changes under adverse weather conditions [56]. The low and medium uncertainty levels account for intermediate deviations that lie between the negligible and high levels. These uncertainty levels were modeled using normal distributions, with mean and standard deviation (S.D.): negligible (mean=0, S.D.=30 seconds), low (mean=0, S.D.=90 seconds), medium (mean=0, S.D.=210 seconds), and high (mean=0, SD=450 seconds). To evaluate the model's performance to these different levels of ETA uncertainty, we conducted 100 independent simulation runs for each uncertainty level, with each run including a batch of 30 aircraft.

The simulated ETA values for each uncertainty level are presented in **Fig. 14**. The results show that the negligible uncertainty level leads to deviations within ±1 minutes, while the low uncertainty level results in deviations within ±5 minutes. Medium uncertainty increases the deviations to ±10 minutes, and high uncertainty leads to deviations of up to ±20 minutes. These simulated ETA values closely align with real-world observations reported in the literature [55,56], which confirmed the validity of the defined uncertainty levels.

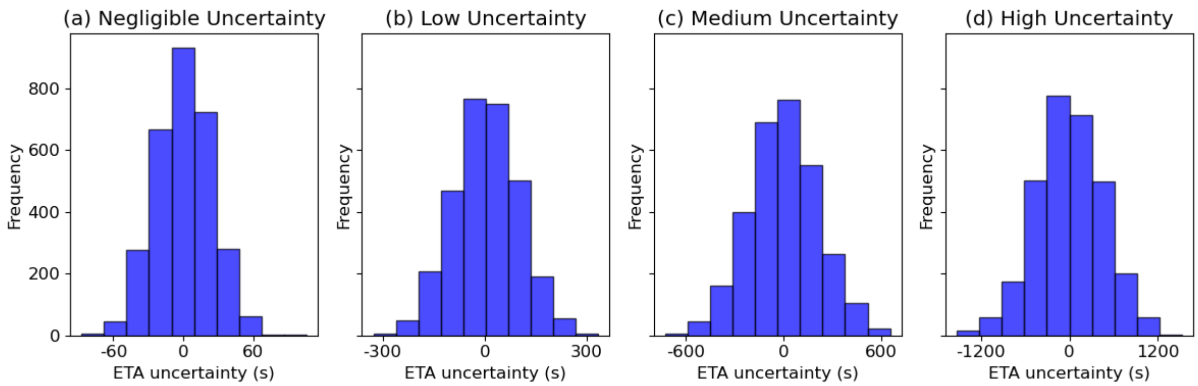


Fig. 14. Distributions of simulated ETA uncertainty at various uncertainty levels.

Table 8 and **Figs. 15-17** present simulation results under different uncertainty levels. Under high uncertainty, the model's performance is significantly impacted due to large deviations in aircraft separations, ranging from as high as 25 minutes to near zero, as shown in **Fig. 16**. The wide range of separations leads to underutilized airspace during periods of large separations, where preceding aircraft exit the airspace well before the following ones enter. This underutilization is demonstrated in **Fig. 15**, where 5.62% of the time the airspace is completely unoccupied. On the other hand, small separations result in a significant increase in the loss of separation rate at entry waypoints, reaching 38% as reported in **Table 8**. This conflict arises as the model primarily reacts to aircraft already within the airspace and does not account for conflicts at entry points caused by uncertain ETAs. Consequently, the total time required to clear a batch of 30 aircraft is the longest among all levels, as shown in **Fig. 17**. These results suggest that future work should consider optimizing multisector scheduling to better manage ETA uncertainties at entry points and improve coordination across airspace sectors.

Table 8. Sensitivity of proposed model under various ETA uncertainty levels in 100 runs.

Performance metrics	Uncertainty levels			
	Negligible	Low	Medium	High
Aircraft LOS rate	1%	2%	6%	38%
Thunderstorm LOS rate	1%	3%	2%	2%
Goal reach rate	98%	95%	92%	60%
Flight distance ratio (mean/S.D.)	1.18/0.23	1.18/0.25	1.15/0.21	1.22/0.28

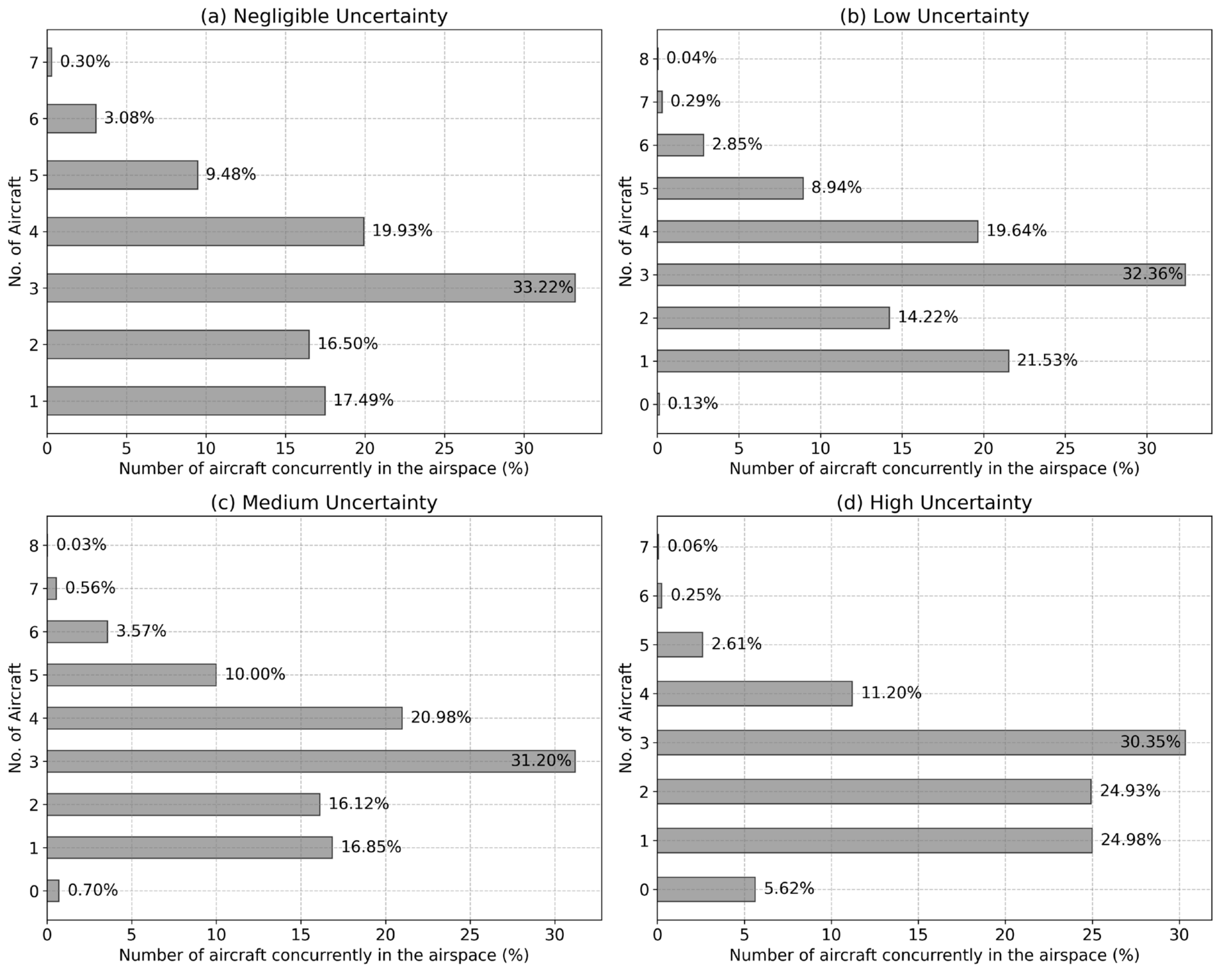


Fig. 15. Aircraft density in the airspace under different ETA uncertainty levels.

At the negligible level, aircraft separations remain consistent and well above the minimum required separation, as observed in **Fig. 16**. This stability ensures optimal airspace utilization, with minimal occurrences of empty airspace, as shown in **Fig. 15**, where only 0.13% of the time the airspace is unoccupied. Consequently, the aircraft maintain safe distances, resulting in the lowest loss of separation rate at 1%, as reported in **Table 8**. Additionally, the high goal reach rate of 98% indicates efficient and effective operations under this level of uncertainty. The total time required to clear a batch of 30 aircraft is also among the shortest, with minimal variance, as illustrated in **Fig. 17**. These findings demonstrate that negligible uncertainty conditions lead to both high safety and efficiency.

Under low and medium uncertainty levels, the model begins to experience moderate impacts on performance. While separations remain generally above the minimum threshold, deviations increase, leading to slight inefficiencies in airspace utilization. As shown in **Fig. 15**, the distribution of aircraft density within the airspace becomes less balanced, with noticeable increases in time periods where fewer or more aircraft occupy the airspace simultaneously. This imbalance results in a gradual rise in the loss of separation rates to 2% for low uncertainty and 6% for medium uncertainty (**Table 8**). Goal reach rates, while still high, decline slightly to 95% and 92% for low and medium uncertainty levels respectively, indicating a moderate degradation in operational effectiveness. The total clearance time for a batch of 30 aircraft also increases slightly, as shown in **Fig. 17**, though the variance remains manageable. These results suggest that while the model remains effective under low and medium uncertainty, its performance could benefit from additional proactive measures such as optimization of ETAs across multiple sectors.

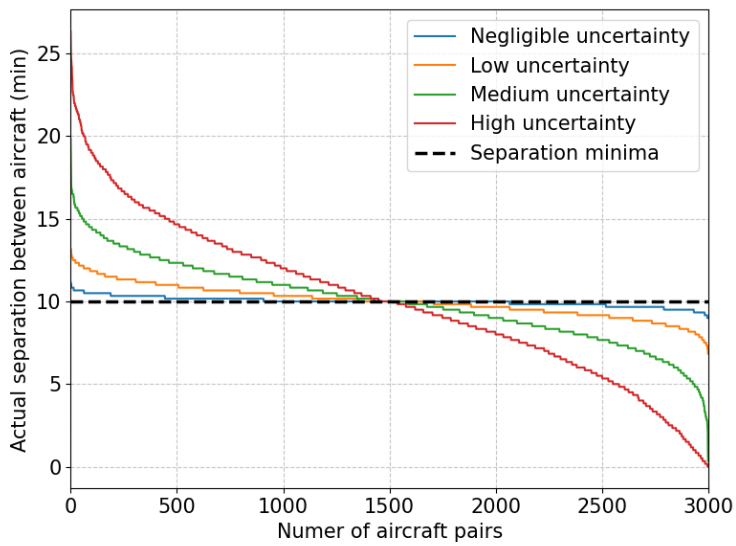


Fig. 16. Actual aircraft separations at entry waypoints under various ETA uncertainty levels. Note that the actual separation between aircraft is sorted and presented from the largest to the smallest for improved presentation clarity.

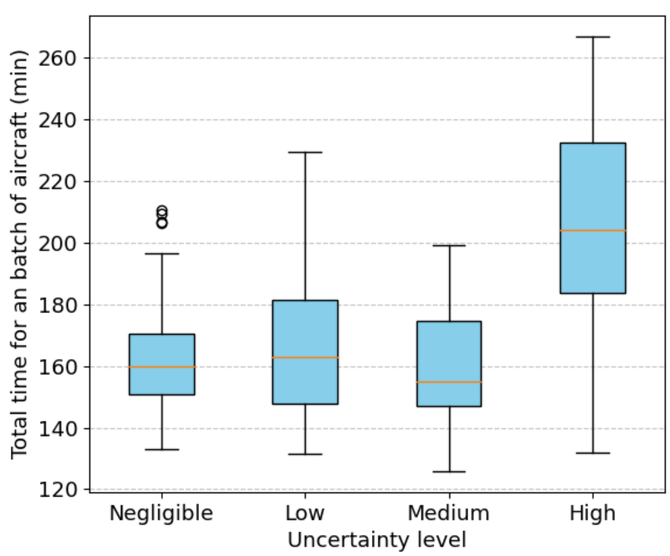


Fig. 17. Total time for completing a simulation run at different uncertainty levels.

The proposed model demonstrates robustness in handling ETA uncertainties at negligible, low, and medium levels, with minimal impacts on safety and efficiency. However, high ETA uncertainty significantly degrades performance due to the lack of optimization for ETAs across multiple sectors, as the model focuses only on within-sector dynamics. In single-sector scenarios, ETA uncertainty has little impact (**Fig. 15**) on demand and capacity balancing since all aircraft adhere to predefined entry points. In contrast, multi-sector scenarios may experience imbalances as rerouted aircraft arrive from non-designated entry points due to adverse weather. Future efforts should incorporate multi-sector coordination and optimize demand-capacity balancing to address these challenges.

5.8. Ablation study of key reward functions used in our model

Ablation studies are essential for evaluating how individual reward components impact model performance [57]. By removing specific reward components that are used in the proposed model, we assess their contributions to safety and efficiency indicators.

In this study, we exclude two foundational reward components from ablation testing, which are crash-related rewards and goal-reaching rewards. Crash-related rewards ensure safety by penalizing direct conflicts with thunderstorms or other aircraft, and their removal leads to non-convergence, invalidating the analysis. Similarly, the goal-reaching reward, which incentivizes aircraft to reach exit waypoints, is crucial for training convergence. These two rewards are indispensable and remain part of the model.

The ablation study focuses on four key reward components: (i) Near aircraft penalty: adds a gradient-based penalty as the separation between aircraft decreases below 30 nm but remains above 5 nm, which helps to reduce the sparse reward effect during training. Removing it will stop this gradient contribution, leading to potential degradation of learning. Similarly, (ii) Near storm penalty: provides a gradient for avoiding conflicts between aircraft and thunderstorms. (iii) Distance to goal: minimizes travel distance and supplies a gradient for each step toward the goal, and its removal may diminish the global gradient that is essential for effective learning. Lastly, (iv) Heading change: provides a small reward that discourages frequent heading adjustments. Besides, we also perform a reward scaling analysis to evaluate the model's robustness to vary reward magnitudes by testing baseline (1x), double (2x), fivefold (5x), and tenfold (10x) reward values.

Each ablation test involves training models with one reward removed and evaluating them under identical simulation environments and parameters as the full model. Each trained model is tested on 100 independent scenarios. Model performances are evaluated with results presented in **Fig. 18**. Note that the aircraft stray rate indicates the percentage of aircraft that have no conflict but do not reach the exit waypoint within a sufficient amount of time step in a simulation run.

As shown in **Fig. 18**, the full model, serving as the baseline, performs well with a goal reach rate of 98%, with only a 2% thunderstorm loss of separation (LOS) rate and no aircraft straying or conflict. In contrast, the most significant degradation occurs when the distance to goal reward is removed. In this case, the goal reach rate drops to 45%, while both aircraft and thunderstorm conflict rates rise to 9%. Additionally, 37% of aircraft stray, failing to reach their designated exit waypoints within the allocated time. This severe performance drop stems from the removal of the gradient provided by this reward component at each step, resulting in a sparse reward environment. In this setting, the agent only receives a one-time reward upon reaching the goal, with no intermediate feedback to guide task completion.

Ablation of the near aircraft penalty and near storm penalty similarly impacts performance, though to a lesser extent. Removing these components diminishes the gradient of proactive avoidance, causing the aircraft to react only when separations fall below critical thresholds with other aircraft or thunderstorms. This leads to increases in conflict rates: 4% aircraft LOS rate and 7% thunderstorm LOS rate for the absence of the near aircraft penalty, while 4% and 11% rates for the absence of the near storm penalty. Conversely, removing the heading change reward has minimal impact on performance. This component introduces only a minor penalty to discourage unnecessary heading adjustments, without contributing significant gradients for task completion.

The reward scaling results in **Table 9** demonstrate the robustness of the proposed model to varying reward magnitudes. Across all scaling levels (1x, 2x, 5x, 10x), the model maintains consistently high performance, with aircraft LOS rates remaining at or near 0%, thunderstorm LOS rates between 1-3%, and goal reach rates stable at 97-98%. These results indicate that the model effectively balances safety and efficiency, regardless of reward scaling.

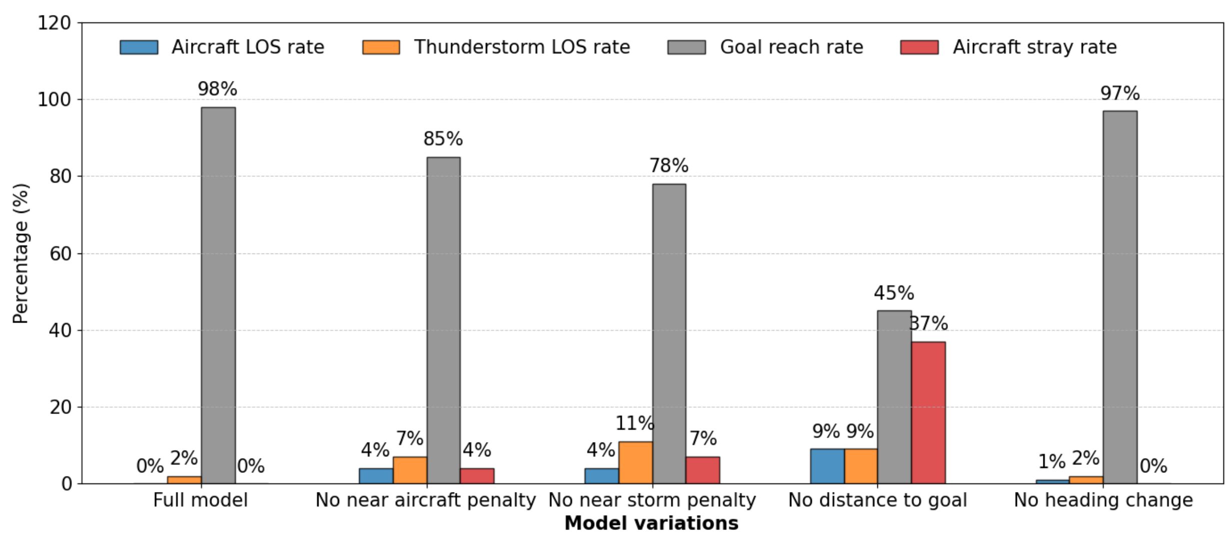


Fig. 18. Ablation study of reward functions.

Table 9. Proposed model is robust to reward scaling effects.

Performance metrics	Reward scale levels			
	1x (baseline)	2x	5x	10x
Aircraft LOS rate	0%	0%	0%	1%
Thunderstorm LOS rate	2%	3%	3%	1%
Goal reach rate	98%	97%	97%	98%
Distance ratio (Mean/S.D.)	1.17/0.15	1.31/0.28	1.16/0.21	1.14/0.11

In summary, ablation results highlight the critical role of the distance to goal reward in providing consistent learning gradients, while near aircraft and near storm penalties ensure proactive safety actions. These findings suggest that the key to designing effective reward functions lies in introducing appropriate gradients to mitigate the sparse reward limitations of reinforcement learning models.

6. Conclusions

This study addresses the growing challenges posed by increasingly frequent and severe thunderstorm cells, a consequence of climate change, on multi-aircraft co-operative trajectory planning in en route airspace. Addressing the limitations of existing centralized methods, we developed a decentralized Markov Decision Process model with a novel approach of the Independent Deep Deterministic Policy Gradient (IDDPG) framework to enhance safety, efficiency, and robustness. Results from extensive real-world and simulated scenarios demonstrated that the proposed framework effectively manages complex scenarios, maintaining low loss of separation (LOS) rates and ensuring high levels of safety even as the number of aircraft and weather severity increased. We summarize the key takeaways as follows.

(i) The proposed model demonstrated strong effectiveness and robustness when applied to various real-world simulated scenarios, particularly in oceanic airspace within the Singapore FIR, where radar coverage and air traffic control services are limited. Across all tested scenarios, the model consistently maintained low LOS rates with other aircraft, ranging from 0% to 4%, even as traffic density doubled compared with the state-of-the-art testing [28]. This indicates its robustness in ensuring safe separation in complex combinations of traffic and weather conditions.

(ii) Simulation results demonstrate the scalability of our model as it effectively handled an increasing number of aircraft, from 4 to 8, with minimal impact on safety. The goal reach rate only slightly decreased from 100% to 95% as traffic density doubled, demonstrating that the model can accommodate higher traffic volumes while maintaining operational efficiency.

(iii) In terms of rerouting efficiency, the model maintained a reasonable flight distance ratio, with a mean ratio of 1.1725 in the most congested scenario involving 8 aircraft with two dynamic thunderstorm cells. This suggests that while additional maneuvers were necessary to avoid conflicts, the overall efficiency of rerouting was maintained.

(iv) We observed the significant impact of trajectory unpredictability of thunderstorm cells on the model's performance. While the model maintained a low aircraft LOS rate across all scenarios, the LOS rate with thunderstorm cells increased notably when the thunderstorm paths were randomized. This finding highlights the importance of incorporating more accurate thunderstorm cell trajectory predictions in future work, as doing so could further enhance the robustness and reduce the likelihood of weather-related conflicts.

While this study represents a significant advancement in applying multi-agent DRL for multi-aircraft cooperative trajectory planning under dynamic thunderstorm cells, several areas remain for further enhancement and exploration. First, future research could enhance the robustness of the model by integrating other uncertainties, particularly regarding thunderstorm evolution [18] and uncertain trajectory positions [54]. Second, the focus of this work is on en route phase, expanding the model to consider different phases of flight, such as in terminal airspace, would provide a more collaborative solution for gate-to-gate air traffic management [58]. Additionally, this work assumes a fully autonomous decision-making environment. Future research could consider human-centered [59] and human-AI hybrid concepts [60], where pilot inputs and air traffic controller feedback are integrated into the decision-making process, enhancing the model's applicability in real-world operations.

Acknowledgements

This research is supported by the Italian Ministry of Foreign Affairs and International Cooperation (MAECI) and the Agency for Science, Technology and Research (A*STAR), Singapore, under the First Executive Programme of Scientific and Technological Cooperation between Italy and Singapore for the years 2023–2025. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Italian Ministry of Foreign Affairs and International Cooperation or the Agency for Science, Technology and Research (A*STAR), Singapore.

References

[1] Eurocontrol, Performance Review Report: An Assessment of Air Traffic Management in Europe, 2024. www.eurocontrol.int/air-navigation-services-performance-review.

[2] National Transportation Safety Board (NTSB), Aviation Investigation Final Report, 2020.

[3] E. Andrés Endériz, Aircraft Trajectory Planning Considering Ensemble Forecasting of Thunderstorms, PhD thesis, Universidad Carlos III de Madrid, 2022.

[4] D. González-Arribas, M. Soler, M. Sanjurjo-Rivo, M. Kamgarpour, J. Simarro, Robust aircraft trajectory planning under uncertain convective environments with optimal control and rapidly developing thunderstorms, Aerosp Sci Technol 89 (2019) 445–459. https://doi.org/10.1016/j.ast.2019.03.051.

[5] SKYbrary, Loss of Separation During Weather Avoidance, (2024). https://skybrary.aero/articles/loss-separation-during-weather-avoidance (accessed September 11, 2024).

[6] C.Y. Yiu, K.K.H. Ng, X. Li, X. Zhang, Q. Li, H.S. Lam, M.H. Chong, Towards safe and collaborative aerodrome operations: Assessing shared situational awareness for adverse weather detection with EEG-enabled Bayesian neural networks, Advanced Engineering Informatics 53 (2022). https://doi.org/10.1016/j.aei.2022.101698.

[7] E. Andrés, J. García-Heras, D. González, M. Soler, A. Valenzuela, A. Franco, J. Nunez-Portillo, D. Rivas, T. Radišić, P. Andraši, Probabilistic Analysis of Air Traffic in Adverse Weather Scenarios, in: International Conference on Research in Air Transportation, USA, 2022.

[8] P. Andraši, T. Radišić, K. Samardžić, D. Novak, Air Traffic Complexity Hotspot Detection in Uncertain Adverse Weather Scenarios, in: International Conference on Research in Air Transportation, Singapore, 2024.

[9] Y. Pang, J. Hu, C.S. Lieber, N.J. Cooke, Y. Liu, Air traffic controller workload level prediction using conformalized dynamical graph learning, Advanced Engineering Informatics 57 (2023). https://doi.org/10.1016/j.aei.2023.102113.

[10] M.H. Nguyen, S. Alam, Airspace Collision Risk Hot-Spot Identification using Clustering Models, IEEE Transactions on Intelligent Transportation Systems 19 (2018) 48–57. https://doi.org/10.1109/TITS.2017.2691000.

[11] R. Xiong, Y. Wang, P. Tang, N.J. Cooke, S. V. Ligda, C.S. Lieber, Y. Liu, Predicting separation errors of air traffic controllers through integrated sequence analysis of multimodal behaviour indicators, Advanced Engineering Informatics 55 (2023). https://doi.org/10.1016/j.aei.2023.101894.

[12] B. Liu, S.W. Lye, Z. Bin Zakaria, An integrated framework for eye tracking-assisted task capability recognition of air traffic controllers with machine learning, Advanced Engineering Informatics 62 (2024). https://doi.org/10.1016/j.aei.2024.102784.

[13] Civil Aviation Authority of Singapore, Aeronautical Information, Publication, 2024. https://www.caas.gov.sg/docs/default-source/docs---ats/aip-singapore---16-may-2024.pdf (accessed August 28, 2024).

[14] H. Erzberger, Automated Conflict Resolution for Air Traffic Control, 2005. http://vams.arc.nasa.gov/activities/aces.html.

[15] P. Zhao, H. Erzberger, Y. Liu, Multiple-aircraft-conflict resolution under uncertainties, Journal of Guidance, Control, and Dynamics 44 (2021) 2031–2049. https://doi.org/10.2514/1.G005825.

[16] H. Erzberger, T.A. Lauderdale, Y.C. Chu, Automated conflict resolution, arrival management, and weather avoidance for air traffic management, Proc Inst Mech Eng G J Aerosp Eng 226 (2012) 930–949. https://doi.org/10.1177/0954410011417347.

[17] H. Erzberger, T. Nikoleris, R.A. Paielli, Y.C. Chu, Algorithms for control of arrival and departure traffic in terminal airspace, Proc Inst Mech Eng G J Aerosp Eng 230 (2016) 1762–1779. https://doi.org/10.1177/0954410016629499.

[18] D. Hentzen, M. Kamgarpour, M. Soler, D. González-Arribas, On maximizing safety in stochastic aircraft trajectory planning with uncertain thunderstorm development, Aerosp Sci Technol 79 (2018) 543–553. https://doi.org/10.1016/j.ast.2018.06.006.

[19] H.K. Ng, S. Grabbe, A. Mukherjee, Design and evaluation of a dynamic programming flight routing algorithm using the convective weather avoidance model, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, 2009: p. 5862. https://doi.org/10.2514/6.2009-5862.

[20] M. Kamgarpour, V. Dadok, C. Tomlin, Trajectory Generation for Aircraft Subject to Dynamic Weather Uncertainty, in: 49th IEEE Conference on Decision and Control (CDC), IEEE, 2010: pp. 2063–2068.

[21] X. Zhang, S. Mahadevan, Aircraft re-routing optimization and performance assessment under uncertainty, Decis Support Syst 96 (2017) 67–82. https://doi.org/10.1016/j.dss.2017.02.005.

[22] C.P. Taylor, P. Coates, D. Larsen, S. Liu, C.R. Wanke, T. Stewart, Adaptive network design for dynamic rerouting, in: 2018 Aviation Technology, Integration, and Operations Conference, American Institute of Aeronautics and Astronautics Inc, AIAA, 2018: p. 4239. https://doi.org/10.2514/6.2018-4239.

[23] J.J. Pannequin, A.M. Bayen, I.M. Mitchell, C. Hoam, S. Sastry, Multiple aircraft deconflicted path planning with weather avoidance constraints, in: Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007, American Institute of Aeronautics and Astronautics Inc., 2007: pp. 2467–2488. https://doi.org/10.2514/6.2007-6588.

[24] S. Summers, Maryam Kamgarpour, John Lygeros, Claire Tomlin, A Stochastic Reach-Avoid Problem with Random Obstacles, in: Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control, ACM Digital Library, 2011: pp. 251–260.

[25] D. González-Arribas, M. Soler, M. Sanjurjo-Rivo, Robust aircraft trajectory planning under wind uncertainty using optimal control, Journal of Guidance, Control, and Dynamics 41 (2018) 673–688. https://doi.org/10.2514/1.G002928.
[26] D.B. Seenivasan, A. Olivares, E. Staffetti, Multi-aircraft optimal 4D online trajectory planning in the presence of a multi-cell storm in development, Transp Res Part C Emerg Technol 110 (2020) 123–142. https://doi.org/10.1016/j.trc.2019.11.014.
[27] E. Andrés, D. González-Arribas, M. Soler, M. Kamgarpour, M. Sanjurjo-Rivo, Informed scenario-based RRT∗ for aircraft trajectory planning under ensemble forecasting of thunderstorms, Transp Res Part C Emerg Technol 129 (2021). https://doi.org/10.1016/j.trc.2021.103232.
[28] A. Guitart, D. Delahaye, F.M. Camino, E. Feron, Collaborative Generation of Local Conflict Free Trajectories With Weather Hazards Avoidance, IEEE Transactions on Intelligent Transportation Systems 24 (2023) 12831–12842. https://doi.org/10.1109/TITS.2023.3289191.
[29] M. Brittain, X. Yang, P. Wei, A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance, (2020) 1–26. http://arxiv.org/abs/2003.08353.
[30] B. Pang, K.H. Low, C. Lv, Adaptive conflict resolution for multi-UAV 4D routes optimization using stochastic fractal search algorithm, Transp Res Part C Emerg Technol (2022). https://doi.org/10.1016/j.trc.2022.103666.
[31] M. Brittain, P. Wei, Autonomous Air Traffic Controller: A Deep Multi-Agent Reinforcement Learning Approach, in: International Conference on Machine Learning (ICML), 2019. http://arxiv.org/abs/1905.01303.
[32] D.T. Pham, P.N. Tran, S. Alam, V. Duong, D. Delahaye, Deep reinforcement learning based path stretch vector resolution in dense traffic with uncertainties, Transp Res Part C Emerg Technol 135 (2022). https://doi.org/10.1016/j.trc.2021.103463.
[33] P. Zhao, Y. Liu, Physics Informed Deep Reinforcement Learning for Aircraft Conflict Resolution, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 8288–8301. https://doi.org/10.1109/TITS.2021.3077572.
[34] Y. Chen, M. Hu, L. Yang, Y. Xu, H. Xie, General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution, Transp Res Part C Emerg Technol 151 (2023). https://doi.org/10.1016/j.trc.2023.104125.
[35] G. Papadopoulos, A. Bastas, G.A. Vouros, I. Crook, N. Andrienko, G. Andrienko, J.M. Cordero, Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts, Expert Syst Appl 236 (2024). https://doi.org/10.1016/j.eswa.2023.121234.
[36] Y. Guleria, D.T. Pham, S. Alam, P.N. Tran, N. Durand, Towards conformal automation in air traffic control: Learning conflict resolution strategies through behavior cloning, Advanced Engineering Informatics 59 (2024). https://doi.org/10.1016/j.aei.2023.102273.
[37] T. Kravaris, C. Spatharis, A. Bastas, G.A. Vouros, K. Blekas, G. Andrienko, N. Andrienko, J.M.C. Garcia, Resolving Congestions in the Air Traffic Management Domain via Multiagent Reinforcement Learning Methods, ArXiv (2019). http://arxiv.org/abs/1912.06860.
[38] D.-T. Pham, L. Long Chan, S. Alam, R. Koelle, Real-time departure slotting in mixed-mode operations using deep reinforcement learning : a case study of Zurich airport, in: Fourteenth USA/Europe Air Traffic Management Research and Development Seminar, 2021. https://dr.ntu.edu.sg.
[39] H. Ali, D.T. Pham, S. Alam, M. Schultz, A Deep Reinforcement Learning Approach for Airport Departure Metering Under Spatial-Temporal Airside Interactions, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 23933–23950. https://doi.org/10.1109/TITS.2022.3209397.
[40] J. Lee, K. Lee, I. Moon, A reinforcement learning approach for multi-fleet aircraft recovery under airline disruption, Appl Soft Comput 129 (2022). https://doi.org/10.1016/j.asoc.2022.109556.
[41] Y. Wang, W. Cai, Y. Tu, J. Mao, Reinforcement-Learning-Informed Prescriptive Analytics for Air Traffic Flow Management, IEEE Transactions on Automation Science and Engineering (2023). https://doi.org/10.1109/TASE.2023.3292921.
[42] C. Spatharis, A. Bastas, T. Kravaris, K. Blekas, G.A. Vouros, J.M. Cordero, Hierarchical multiagent reinforcement learning schemes for air traffic management, Neural Comput Appl 35 (2023) 147–159. https://doi.org/10.1007/s00521-021-05748-7.
[43] Y. Chen, Y. Xu, M. Hu, General multi-agent reinforcement learning integrating heuristic-based delay priority strategy for demand and capacity balancing, Transp Res Part C Emerg Technol 153 (2023). https://doi.org/10.1016/j.trc.2023.104218.
[44] Y. Ding, S. Wandelt, G. Wu, Y. Xu, X. Sun, Towards efficient airline disruption recovery with reinforcement learning, Transp Res E Logist Transp Rev 179 (2023). https://doi.org/10.1016/j.tre.2023.103295.

[45] B. Pang, W. Dai, X. Hu, F. Dai, K.H. Low, Multiple air route crossing waypoints optimization via artificial potential field method, Chinese Journal of Aeronautics 34 (2021). https://doi.org/10.1016/j.cja.2020.10.008.
[46] M. Petrik, S. Zilberstein, Average-Reward Decentralized Markov Decision Processes, in: International Joint Conferences on Artificial Intelligence, 2007: pp. 1997–2002.
[47] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: International Conference on Learning Representations (ICLR), 2016. http://arxiv.org/abs/1509.02971.
[48] J. Yuan, Y. Pei, Y. Xu, Y. Ge, Z. Wei, Autonomous interval management of multi-aircraft based on multi-agent reinforcement learning considering fuel consumption, Transp Res Part C Emerg Technol 165 (2024). https://doi.org/10.1016/j.trc.2024.104729.
[49] J.K. Gupta, M. Egorov, M. Kochenderfer, Cooperative Multi-Agent Control Using Deep Reinforcement Learning, in: Autonomous Agents and Multiagent Systems (AAMAS), 2017: pp. 66–83.
[50] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533. https://doi.org/10.1038/nature14236.
[51] EuroControl, EUROCONTROL MANUAL FOR AIRSPACE PLANNING-COMMON GUIDELINES, 2003. www.eurocontrol.int.
[52] R. Lowe, Y. Wu, A. Tamar, J. Harb, P.A. Uc, B. Openai, I.M. Openai, Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, n.d.
[53] C. Yan, C. Wang, X. Xiang, K.H. Low, X. Wang, X. Xu, L. Shen, Collision-Avoiding Flocking With Multiple Fixed-Wing UAVs in Obstacle-Cluttered Environments: A Task-Specific Curriculum- Based MADRL Approach, IEEE Trans Neural Netw Learn Syst 35 (2024) 10894–10908. https://doi.org/10.1109/TNNLS.2023.3245124.
[54] B. Pang, K.H. Low, V.N. Duong, Chance-constrained UAM traffic flow optimization with fast disruption recovery under uncertain waypoint occupancy time, Transp Res Part C Emerg Technol 161 (2024) 104547. https://doi.org/10.1016/j.trc.2024.104547.
[55] N. Takeichi, Adaptive prediction of flight time uncertainty for ground-based 4D trajectory management, Transp Res Part C Emerg Technol 95 (2018) 335–345. https://doi.org/10.1016/j.trc.2018.07.028.
[56] A. Muñoz Hernández, M. Polaina Morales, A. Güemes, M. Soler, D. González-Arribas, J. Pons, X. Prats, A. Tutku Altun, E. Koyuncu, A. Kuenz, R. Zopp, D. Delahaye, Data-driven methodology for uncertainty quantification of aircraft trajectory predictions, in: 2021 IEEE/AIAA 40th Digital Avionics Systems Conference., San Antonio, TX, USA., 2021: pp. 1–10.
[57] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D.S. Deepmind, Rainbow: Combining Improvements in Deep Reinforcement Learning, in: The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)3215, 2017. www.aaai.org.
[58] S.C. Johnson, B.E. Barmore, Nextgen far-term concept exploration for integrated gate-to-gate trajectory-based operations, in: 16th AIAA Aviation Technology, Integration, and Operations Conference, American Institute of Aeronautics and Astronautics Inc, AIAA, 2016: pp. 1–13. https://doi.org/10.2514/6.2016-4355.
[59] Q. Li, K.K.H. Ng, Z. Fan, X. Yuan, H. Liu, L. Bu, A human-centred approach based on functional near-infrared spectroscopy for adaptive decision-making in the air traffic control environment: A case study, Advanced Engineering Informatics 49 (2021). https://doi.org/10.1016/j.aei.2021.101325.
[60] J. Perez-Cerrolaza, J. Abella, M. Borg, C. Donzella, J. Cerquides, F.J. Cazorla, C. Englund, M. Tauber, G. Nikolakopoulos, J.L. Flores, Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey, ACM Comput Surv 56 (2024). https://doi.org/10.1145/3626314.