

General Multi-agent Reinforcement Learning Integrating Adaptive Manoeuvre Strategy for Real-time Multi-aircraft Conflict Resolution

Yutong Chen^{a,b}, Minghua Hu^a, Lei Yang^{a,*}, Yan Xu^b, Hua Xie^a

^a*Nanjing University of Aeronautics and Astronautics, Nanjing, China, 210000*

^b*Cranfield University, Cranfield, United Kingdom, MK43 0AL*

Abstract

Reinforcement learning (RL) techniques are under investigation for resolving conflict in air traffic management (ATM), exploiting their computational capabilities and ability to cope with flight uncertainty. However, the limitations of generalisation make it difficult for existing RL-based conflict resolution (CR) methods to be effective in practice. This paper proposes a general multi-agent reinforcement learning (MARL) method that integrates an adaptive manoeuvre strategy to enhance both the solution's efficiency and the model's generalisation in multi-aircraft conflict resolution (MACR). A partial observation approach based on the imminent threat detection sectors is used to gather critical environmental information, enabling the model to be applied in arbitrary scenarios. Agents are trained to provide the correct flight intention (such as increasing speed and yawing to the left), while an adaptive manoeuvre strategy generates the specific manoeuvre (speed and heading parameters) based on the flight intention. To address flight uncertainty and performance challenges caused by the intrinsic non-stationarity in MARL, a warning area for each aircraft is introduced. We employ a state-of-the-art Deep Q-learning Network (DQN) method, Rainbow DQN, to improve the efficiency of the RL algorithm. The multi-agent

*Corresponding author

Email addresses: chenyutong@nuaa.edu.cn (Yutong Chen), minghuahu@nuaa.edu.cn (Minghua Hu), laneyoung@nuaa.edu.cn (Lei Yang), yanxu@cranfield.ac.uk (Yan Xu), marcuse@163.com (Hua Xie)

system is trained and deployed in a distributed manner to adapt to real-world scenarios. A sensitivity analysis of uncertainty levels and warning area sizes is conducted to explore their impact on the proposed method. Simulation experiments confirm the effectiveness of the training and generalisation of the proposed method.

Keywords: Air traffic management; Multi-aircraft conflict resolution; Multi-agent reinforcement learning; Deep q-learning network; Generalisation; Uncertainty

1. Introduction

Conflict resolution (CR) technology is a key air traffic management (ATM) technique that improves air traffic flow and allows for more flights to be operated in limited airspace. In high-density scenarios, multi-aircraft conflict resolution
5 (MACR) technology is especially critical for efficient and safe air traffic operations.

CR methods are mainly classified into two categories based on the control mode: centralised and distributed. The centralised method aims to build a global optimisation model that generates conflict-free paths for all conflicting
10 aircraft. This method assumes that all aircraft follow a predetermined trajectory, and it can obtain a globally optimal solution using an exact method such as linear programming. However, this method is often time-consuming and may not be able to complete the solution in a limited time [1]. Alternatively, the centralised method can use a heuristic algorithm to obtain a satisfying solution
15 quickly, but it does not guarantee optimality [2]. Since the optimisation models need to be built before flight operation, it is challenging for them to handle operational dynamics and uncertainty in practice. Although uncertainty can be considered in the model based on probabilities, it is more difficult to solve and does not guarantee safety. In contrast, the distributed method deploys pro-
20 grammed agents using a rule-based method to generate conflict-free paths. This method can better cope with the dynamics of the environment due to the quick

response of the agents. However, agile agents come at the expense of their short-sightedness, which may only solve local problems. For example, a manoeuvre to resolve a conflict may result in secondary conflicts, leading to poor stability [3]. The effectiveness and safety of the rule-based distributed method in complex scenarios need further validation. Due to the various shortcomings of the existing methods, real-time conflict resolution still relies on controller instructions in the real world. However, human controllers are not always reliable, and their workload limits the operational efficiency of the airspace. Therefore, it is important to develop a real-time multi-aircraft conflict resolution method that can be applied in practice.

In recent years, significant progress has been made in research on reinforcement learning (RL) methods, revealing the potential of RL-based CR methods for practical applications. The RL method to solving CR problems involves two primary steps: training the agent (neural network) in a designed reinforcement learning environment to acquire a good policy (training process), and deploying the trained agents to a CR problem to solve it (solving process). While the training process is time-consuming, the solving process is generally calculated quickly, providing an acceptable solution. RL methods transfer a significant amount of solution time to the training stage, allowing them to respond quickly to actual scenarios in operation. The basis of RL is the Markov decision process (MDP). Traditional MDP-based CR methods usually cannot circumvent the curse of dimensionality [4] [5] (i.e., as the dimensionality of the state and action space increases, the difficulty of solving problems by traditional MDP-based methods increases exponentially) [6]. In contrast, RL methods can handle inexperienced states with the aid of neural networks, which somewhat mitigates the negative effects of the increased dimensionality. Compared with exact methods, RL methods have advantages in computational performance, allowing them to iterate in real-time to cope with flight uncertainty. RL methods have more potential for better optimisation performance than heuristic or rule-based methods because RL methods allow flexibility in learning the optimal policy without being overly constrained by designed rules.

However, there are still several challenges that need to be addressed when applying RL methods to CR problems. One of the most pressing issues is generalisation, which refers to the ability of the trained agent to be deployed in arbitrary scenarios. The number of intruders in a given scenario can vary greatly, and most existing RL-based CR methods require observation of information from all intruders. However, the dimension of the observation matrix in RL methods is typically fixed, which limits the generalisation of the trained model. Another challenge is multi-aircraft conflict resolution, which is difficult for existing RL-based CR methods since they usually train the agent for a specific aircraft. Therefore, they cannot handle MACR problems or employ a rule-based mechanism, which limits the solvable scenarios and narrows the solution space. In addition, other challenges, such as ensuring satisfactory success rates of conflict resolution and minimising additional flight costs, must also be considered. (More information on existing RL-based CR methods can be found in [Section 2.1.](#))

This paper proposes a general multi-agent reinforcement learning (MARL) method that integrates an adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution. The main contributions of this study can be summarised as follows:

1. The proposed model can be applied to arbitrary real-time multi-aircraft conflict resolution problems, regardless of the number of aircraft or the airspace structure in a scenario. This is mainly possible due to the partial observation approach based on the imminent threats in detection sectors.
2. The multi-agent reinforcement learning integrates an adaptive manoeuvre strategy, which reduces the potential learning tasks for agents, thereby reducing the learning difficulty.
3. A warning area of aircraft is introduced, serving as a buffer to handle the flight uncertainty and performance challenges posed by the intrinsic non-stationarity in MARL. The law that the uncertainty level and warning area size affect the performance of the proposed method is explored.

4. Simulation experiments validate the proposed method’s generalisation in arbitrary scenarios with a pre-set uncertainty level and warning area size.

85 The study also reveals the trends in the effectiveness and optimality of the proposed method with a function of airspace density. The proposed method produces more competitive results than the state-of-the-art RL-based CR methods in a limited experimental environment with a limited amount of publicly available data.

90 This paper is structured as follows. In [Section 2](#), we provide a literature review and discuss the motivation behind this study. [Section 3](#) presents the problem formulation for multi-aircraft conflict resolution using MARL. In [Section 4](#), we detail how to construct the multi-agent reinforcement learning environment, the reinforcement learning architecture, and the neural network training process. In [Section 5](#), we present our RL training process, performance test results, sensitivity analysis, and comparative experiments with similar methods. Finally, 95 we summarise our conclusions and future work in [Section 6](#).

2. Literature review

In this section, we present a literature review of relevant research. Subsequently, we highlight the motivation behind our study, which aims to address 100 the gaps between existing RL-based CR methods and their practical application to multi-aircraft conflict resolution.

2.1. Related work

Numerous studies have been conducted on conflict resolution in the field of air traffic management (ATM), and a review of these studies can be found in the 105 work of Ribeiro et al. [\[14\]](#). RL methods have been applied in various aspects of ATM, such as air traffic flow management [\[15\]](#) [\[16\]](#) [\[17\]](#), assistant control tool [\[18\]](#) [\[19\]](#) [\[20\]](#) and collision avoidance [\[21\]](#) [\[22\]](#) [\[23\]](#). As mentioned, RL methods possess significant research value and potential for application in conflict resolution 110 due to their ability to deal with dynamic and uncertain scenarios. Several

Table 1: Features of CR methods based on RL.

No.	Papers	MARL	MACR	General- isation	2D scenario	3D scenario	Uncertainty	Trajectory recovery	Shared policy	RL methods	Action space
1	Pham [7]	◦	◦	◦	✓	◦	✓	✓	N/A	DDPG	continuous
2	Wang [8]	◦	◦	◦	✓	◦	◦	✓	N/A	AC	continuous
3	Lai [9]	✓	✓	◦	✓	◦	◦	✓	◦	DDPG	continuous
4	Dalman [10]	✓	✓	✓	✓	◦	◦	◦	✓	PPO	discrete
5	Sui [11]	✓	✓	◦	✓	✓	◦	◦	✓	DQN	discrete
6	Brittain [12]	✓	✓	✓	◦	◦	✓	N/A	✓	PPO	discrete
7	Zhao [13]	◦	✓	✓	✓	◦	◦	◦	✓	PPO	discrete
8	this paper	✓	✓	✓	✓	◦	✓	✓	✓	DQN	discrete

representative RL-based CR methods are summarised in Table 1, where a check mark (✓) denotes the presence of a feature and a circle mark (◦) indicates its absence. Pham et al. employed single-agent RL to train a policy that identifies a rerouting point to resolve conflicts while considering the flight uncertainty of the controlled aircraft [7]. In Wang’s method, the RL environment is similar to Pham’s, but the agent selects multiple rerouting points to enhance the CR performance [8]. However, Pham and Wang’s methods utilise a single agent, which can only control one aircraft to resolve conflicts related to itself, making them unsuitable for handling the MACR problem. Moreover, MACR methods outperform single-aircraft CR [24]. To address this, Lai et al. proposed a MARL method that deploys an agent on each aircraft to choose a rerouting point to resolve conflicts [9]. However, Lai’s model is not scalable since it can only handle scenarios with a specific number of aircraft. This is because the input layer dimensions are fixed for most neural networks used for RL. Additionally, MACR methods that utilise periodic or time step manoeuvres have a greater solution space and better CR performance than those that plan one or several rerouting points. To this end, Dalman and Allard introduced a MARL method based on message-passing neural networks, where the aircraft manoeuvre once per time step (if required), and attention mechanisms are employed in graph neural networks to make the method invariant to the number of aircraft in the environment [10]. However, Dalman’s method does not consider flight uncertainty and does not guarantee that the aircraft will reach the exit point

(i.e. trajectory recovery). Sui et al. overcame the observation matrix’s dimensional limitation by presenting a cuboid-based observation model, which can
135 be applied to 3D scenarios [11]. Nonetheless, Sui’s method’s conflict resolution success rate in their published available test is only approximately 60%, which is most likely due to the agent’s difficulty in understanding its state (e.g., position and velocity) accurately via the observation matrix. Furthermore, some elements in the observation matrix are not mathematically significant, such as
140 the flight call sign, resulting in a loss of functionality in generalisation. Brittain et al. proposed a generalised method for conflict resolution by employing the Long Short-Term Memory (LSTM) model, which is capable of adapting to changing input layer dimensions [12]. However, this method is limited to simple route scenarios as the agent only considers speed regulation. Zhao’s method
145 utilises a solution space diagram that integrates intruder information, such as quantity, speeds, heading angles, and positions, into an image. This diagram with a specific pixel is then fed into the neural network to resolve conflicts [13]. Although Zhao’s single-agent RL framework achieves generality to the MACR problem by sharing a policy network among the agents and applying a rule-
150 based mechanism, only one aircraft can manoeuvre to resolve the conflict in each time step, which is selected in sequence for MACR scenarios. This could result in high communication requirements if the time step is too short or missing the optimal timing for aircraft manoeuvring if the time step is too long. In addition, the effectiveness and efficiency of Zhao’s method in MACR have not
155 been adequately verified, and uncertainty is not considered.

The choice of RL algorithm is critical for RL-based CR methods, in addition to the design of the environment model. It heavily depends on the type of action space, which is usually classified into two categories: continuous and discrete. A commonly used model for continuous action space is to select a point
160 within a continuous two-dimensional space as the aircraft’s rerouting point, as seen in the works of Pham, Wang, and Lai. These scholars prefer to use Deep Deterministic Policy Gradient (DDPG) [25] and Actor-Critic (AC) [26] to handle this continuous action space. While the continuous action space provides

a wider range of manoeuvres for the aircraft than the discrete, their CR method
of selecting only one or several rerouting points dramatically reduces the solution
space, as mentioned before. An experiment in Zhao’s research has shown that
models based on discrete action space are less challenging to train by RL than
those based on continuous action space for the CR problem [13]. Therefore, an
increasing number of scholars have opted for discrete action spaces, where they
typically use a set of discrete velocity vectors (Zhao’s) or discrete variations of
velocity vectors (Dalmau’s, Sui’s, and Brittain’s). Proximal Policy Optimisation
(PPO) [27] is often used for MACR problems based on continuous action space,
likely because it is less affected by the non-stationarity of MARL. Deep Q-
learning Network (DQN) [28] is also favoured due to its inherent suitability for
discrete action spaces with a small number of actions.

In addition, other studies have also contributed to the development of RL
methods in the field of aircraft CR. Guo et al. conducted research on safety
enhancement for deep RL in autonomous separation assurance [29], while Brit-
tain et al. proposed a deep MARL method in this field [30] [31]. Pham et al.
discussed the RL method for two-aircraft CR in the presence of uncertainty [32].
Yilmaz et al. presented a deep RL method for air traffic optimisation using the
MuZero algorithm [22]. Ribeiro et al. discussed the improvement of CR at high
density through RL [33].

2.2. Motivation

Based on the discussion of related work, it remains challenging for existing
RL-based CR methods to solve real-time multi-aircraft conflict resolution prob-
lems with arbitrary scenarios (i.e., method’s generalisation). Moreover, there
is a need for RL-based CR methods to further improve their conflict resolution
success rate for flight safety, while also considering flight uncertainty. From a
practical perspective, our study on the RL-based CR method focuses on the
following issues:

1. *Generalisation*: The trained agent should be able to handle multi-aircraft
conflict resolution problems for any scenario (e.g., with an arbitrary num-

ber of aircraft). Only generalised RL models can be deployed directly to
195 take advantage of their rapid solutions in a real-time dynamic environment.

2. *High performance*: The method should not only achieve a satisfactory success rate in conflict resolution but also minimise additional flight costs.
3. *Flight uncertainty*: The trained agents should be able to handle flight un-
200 certainty (e.g., caused by navigational or mechanical errors). In practical applications, uncertainty is a factor that must be considered.

Based on the above issues, this paper proposes a general multi-agent reinforcement learning method integrating an adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution(the features of our method are also summarised
205 in [Table 1](#)).

In terms of generalisation, RL-based CR methods are typically restricted by the fixed dimension of their observation matrix. Many existing methods employ an intruder-centric observation approach, meaning that the agent can only observe a certain number of intruders. If the number of intruders changes in a
210 given scenario, the trained agent will no longer be applicable. To address this generalisation issue, we suggest a partial observation approach based on detection sectors that pose imminent threats (refer to [Figure 4](#)). This approach is area-centric and only observes the intruder in each detection sector that poses the greatest threat to the ownship (e.g., the intruder closest to the ownship). As
215 the number of detection sectors is pre-defined and fixed, the model can be generalised for arbitrary scenarios. The effectiveness and optimality of the proposed method are explored with a function of airspace density (refer to [Section 5.3](#)).

In terms of high performance, RL-based CR methods are typically constrained by the difficulty of learning. Giving the agent a high degree of freedom
220 in its actions (such as a continuous action space in speed and heading) and allowing the agent to independently solve the CR problem leads to a near-complete solution space, which can improve the method’s optimisation performance. However, this can be very challenging for RL-based agents to learn a

good policy, and as a result, the effectiveness and efficiency are often low. For
 225 instance, the success rate of conflict resolution is low [11], and even when there
 is no potential conflict, the aircraft may not be flying straight to its destination
 [34]. This is because the agent has at least three learning tasks in that case: (a)
 how to resolve conflicts, (b) how to reach their destinations (trajectory recovery),
 and (c) how to manoeuvre within the aircraft’s performance constraints.
 230 To reduce the learning tasks and improve the method’s functionality, we integrate
 an adaptive manoeuvre strategy into the reinforcement learning model. This strategy
 achieves the function of trajectory recovery and ensures that the manoeuvre meets
 the aircraft performance constraints through model design rather than learning. As
 a result, the proposed method only requires learning a
 235 single task (i.e., deciding flight intentions such as speeding up and yawing to the
 left), which reduces the learning difficulty compared to the three learning tasks
 and enhances the proposed method’s potential to learn an acceptable policy.
 Furthermore, our model has a discrete action space with a small number of actions,
 and we use the state-of-the-art algorithm based on DQN, Rainbow DQN
 240 [35]. This algorithm significantly improves RL efficiency by integrating multiple
 DQN enhancement technologies (as described in Section 4.4). We compare the
 performance of the proposed method with representative state-of-the-art RL-based
 CR methods (as described in Section 5.5).

In terms of flight uncertainty, RL-based CR methods have an advantage
 245 over conventional optimisation methods due to their data-driven method. To
 further enhance the proposed method’s ability to handle flight uncertainty, we
 include a layer of warning area outside each aircraft’s protected area (refer to
 Section 4.1.3). If an intruder enters the warning area (but not the protected
 area), the agent receives a minor penalty (less than the penalty received if
 250 the intruder enters the protected area). The warning area acts as a buffer
 to prevent conflicts. Additionally, constructing a distributed decision-making
 system with multiple agents may lead to non-stationarity in MARL, resulting in
 uncertainty in the system. It is challenging to solve this non-stationarity from
 an intrinsic mechanism [36]. In the proposed MACR method, the non-stationarity

255 is specifically manifested as the agent’s inability to predict accurately the actions
of other aircraft in air traffic management operations, potentially reducing the
method’s conflict resolution performance. Therefore, we designed the warning
area to address the performance challenges caused by intrinsic non-stationarity
in MARL rather than designing some MARL interaction mechanism to eliminate
260 the non-stationarity in principle. Finally, we perform sensitivity analysis on
uncertainty levels and warning area sizes to explore their impact on the proposed
method (refer to [Section 5.4](#)).

3. Multi-aircraft conflict resolution problem

In this section, we first provide a clear definition of the MACR problem
265 to clarify the research scope of this study. Then, we propose a multi-agent
reinforcement learning framework and carefully formulate the MACR problem
as the basis of the MARL environment.

3.1. Problem definition

The problem addressed in this study is real-time multi-aircraft conflict res-
270 olution with uncertainty. To further clarify the issue, the following notes and
assumptions are provided:

1. The operational context is free route airspace (FRA) [\[37\]](#), where aircraft
manoeuvres are not restricted to fixed routes.
2. Aircraft operate autonomously, and operational delays due to human fac-
275 tors (pilots and controllers) are not considered.
3. The kinematic state of an aircraft (e.g., speed and heading) changes in-
stantaneously.
4. Communication delays and failures are not taken into account.
5. Only horizontal manoeuvres constrained by horizontal separation mini-
280 mum are considered.
6. Differences in aircraft performance are taken into account. Different air-
craft have different airspeed ranges.

7. Constraints in aircraft performance, such as airspeed ranges and heading deflection rate, are considered.
- 285 8. Aircraft must depart from a pre-defined entry point and arrive at a pre-defined exit point.
9. Flight uncertainty (e.g., caused by navigational or mechanical errors) is considered. This study models flight uncertainty as random speed error and heading error for a standard normal distribution.
- 290 10. Aircraft can obtain flight information about other aircraft in the airspace through ADS-B IN and OUT [38], such as position, heading, and speed. Additionally, it is assumed that intent (e.g., the location of the exit point) and performance (e.g., speed range) information can be shared between aircraft.

295 The multi-aircraft conflict resolution problem in this study is to avoid the loss of safe separation by manoeuvring within the performance constraints of the aircraft during the tactical stage. The objective of the optimisation is to minimise the average extra flight distance.

3.2. Problem formulation

300 The multi-agent reinforcement learning framework for multi-aircraft conflict resolution in this study is shown in Figure 1. All N aircraft are modelled as having the same mass in a two-dimensional plane. An agent is deployed on each aircraft to form a distributed decision-making system. The agents share a common neural network (i.e., shared policy). The aircraft update their
305 location information every Δt seconds. At each time step t , the i th aircraft ($i \in \{0, 1, \dots, N\}$) has access to a partial observation $\mathbf{o}_i^t$ that only provides partial global state information. Note that since partial observations are used in this study, we use $\mathbf{s}_i^t$ to denote the global state (observation) and $\mathbf{o}_i^t$ to denote the partial state (observation) for ease of differentiation, where $(\mathbf{o}_i^t \sim \mathcal{O}(\mathbf{s}_i^t))$ (i.e.,
310 $\mathbf{o}_i^t$ is obtained from $\mathbf{s}_i^t$ based on the partial observation rule $\mathcal{O}$). The partial observation $\mathbf{o}_i^t$ is the agent’s input state in RL. Based on the partial observation $\mathbf{o}_i^t$, the i th aircraft selects an action a_i^t that guides itself toward the exit point

$\mathbf{p}_i^{\text{Exit}}$ from its current position $\mathbf{p}_i^t$. It must ensure that the horizontal distance from any other aircraft is greater than the minimum horizontal safety distance $D^{\text{Horizontal}}$ at all time. For each action, the environment provides feedback in the form of a reward (which may be positive or negative) to the agent, which is then used to update the neural network.

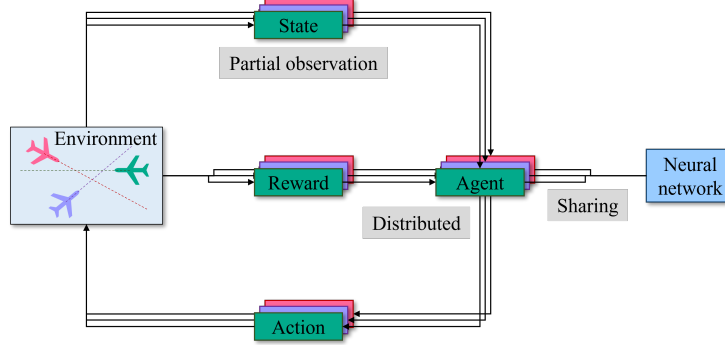


Figure 1: Multi-agent reinforcement learning framework for multi-aircraft conflict resolution. The framework utilizes a distributed decision-making system, where each agent shares a common neural network. The agents use partial observations to obtain state information.

Based on the problem definition presented in [Section 3.1](#) and even in real-world scenarios, aircraft can obtain the global state $\mathbf{s}_i^t$ if it includes state information of all aircraft. However, in our method, aircraft only use local states $\mathbf{o}_i^t$ for two main reasons. Firstly, aircraft are generally more affected by nearby intruders. If agents can focus on critical information by only partially observing within an appropriate range, it can significantly improve the efficiency of learning to solve local conflict resolution, even if there is no guarantee of the global optimal solution. Secondly, and more importantly, global observations centred around a particular aircraft are usually incompatible with fixed-dimensional observation matrices. A partial observation approach using detection sectors based on imminent threats was proposed (as shown in [Figure 2](#)), inspired by the way autonomous collision avoidance robots perform partial observation through laser scanning [\[34\]](#). The aircraft performing detection is referred to as the ownship. The detection area is a circle with the ownship as the centre and $R^{\text{Detection}}$

the action a^t is sampled from a policy π shared among all aircraft:

$$a^t \sim \pi_\theta(\mathbf{a}^t | \mathbf{o}^t) \quad (3)$$

where θ represents the parameters of the neural network. The computed action a^t represents a flight intention, which can be transferred into a manoeuvre that guides the aircraft towards its exit point of the airspace while resolving conflicts with other aircraft within the time horizon Δt until the next observation $\mathbf{o}^{t+1}$ is received. The aircraft makes decisions based only on its current partial observation at each time step until it exits the airspace. This study assumes that heading and speed changes are instantaneous, so the manoeuvre of the aircraft can be represented by its velocity $\mathbf{v}^t$, which consists of heading h^t and speed v^t :

$$\mathbf{v}^t = [h^t, v^t] \quad (4)$$

The flight management system (FMS) on board performs aircraft manoeuvres autonomously based on the velocity $\mathbf{v}^t$.

In this study, flight uncertainty is taken into consideration to achieve greater realism. Although it is beyond the scope of this study to incorporate entirely realistic uncertainty factors, we refer to the flight uncertainty modelling approach in a state-of-the-art MACR method [39]. Flight uncertainty is simplified to heading error $\varepsilon_h^t \in [-E_h, E_h]$ and speed error $\varepsilon_v^t \in [-E_v, E_v]$. Here, ε_h^t represents the absolute value of the heading error, while ε_h^t represents the ratio of the speed error. Both follow a standard normal distribution, and the upper and lower bounds (i.e., $[-E_h, E_h]$ and $[-E_v, E_v]$) correspond to the 95% confidence interval of the standard normal distribution. During simulation, sampling results within the interval are used, and re-sampling occurs otherwise. Using $\mathbf{v}^t$, the actual velocity $\mathbf{u}^t$ is obtained by sampling ε_h^t and ε_v^t as follows:

$$\mathbf{u}^t \sim \mathbf{v}^t(\varepsilon_h^t, \varepsilon_v^t) \quad (5)$$

More specifically, it can be represented as:

$$\mathbf{u}^t = [h^t + \varepsilon_h^t, v^t(1 + \varepsilon_v^t)] \quad (6)$$

The position of the aircraft at each time step can then be calculated using:

$$\mathbf{p}^t = \mathbf{p}^{t-1} + \mathbf{u}^t \Delta t \quad (7)$$

Since we assume that the aircraft is flying in a straight line at a constant speed
 370 in each flight segment, Equation 7 can be extended to:

$$\mathbf{p}^t(x) = \mathbf{p}^{t-1} + x\mathbf{u}^t \Delta t, x \in (0, 1] \quad (8)$$

where $\mathbf{p}^t(x)$ denotes the position of the aircraft at time $(t-1+x)\Delta t$. Note
 that Equation 7 is based on discrete time steps, while Equation 8 is based on
 continuous time. Based on the sequential decisions and sampling errors, the
 trajectory of the aircraft l can be computed from its entry point $\mathbf{p}^{\text{Entry}}$ to exit
 375 point $\mathbf{p}^{\text{Exit}}$ of the airspace as follows:

$$l = \left\{ \overrightarrow{\mathbf{p}^{t-1} \mathbf{p}^t} \left| \begin{array}{l} \mathbf{u}^t \sim \mathbf{v}^t(\varepsilon_h^t, \varepsilon_v^t), \mathbf{v}^t \sim \pi_\theta(a^t | \mathbf{o}^t), \\ t = t^{\text{Entry}} + 1, \dots, t^{\text{Exit}} \end{array} \right. \right\} \quad (9)$$

where $\overrightarrow{\mathbf{p}^{t-1} \mathbf{p}^t}$ represents the trajectory from $\mathbf{p}^{t-1}$ to $\mathbf{p}^t$. t^{Entry} and t^{Exit} , re-
 spectively, denote the time steps of the aircraft entering and exiting the airspace
 (i.e., $\mathbf{p}^{\text{Entry}} = \mathbf{p}^{t^{\text{Entry}}}$ and $\mathbf{p}^{\text{Exit}} = \mathbf{p}^{t^{\text{Exit}}}$).

To summarise the above formulation, we define $\mathbb{L}$ as the set of trajectories for
 380 all aircraft. $\mathbb{L}$ is subject to the aircraft performance constraints (e.g., maximum
 turning rate $\omega_i^{\max}$ and speed limit $[v_i^{\min}, v_i^{\max}]$) and conflict-free constraints
 (the conflict detection in this study is based on continuous time):

$$\mathbb{L} = \left\{ l_i, i = 1, \dots, N \left| \begin{array}{l} \forall j \in \{1, \dots, N\}, j \neq i, \\ \forall x \in (0, 1] : \\ \|\mathbf{p}_i^t(x) - \mathbf{p}_j^t(x)\| > D^{\text{Horizontal}} \\ \wedge |\mathbf{v}_i^t| \in [v_i^{\min}, v_i^{\max}] \\ \wedge |h_i^t - h_i^{t-1}| \leq \omega_i^{\max} \Delta t \end{array} \right. \right\} \quad (10)$$

where $\|\mathbf{p}_1 - \mathbf{p}_2\|$ represents the Euclidean distance between two points.

To find an optimal policy shared by all aircraft, we adopt an objective by
 385 minimising the expectation of the mean flight distance of all aircraft in the same

scenario, which is defined as:

$$\arg \min_{\pi_{\theta}} \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N l_i | \pi_{\theta} \right] \quad (11)$$

4. Reinforcement learning approach integrated with an adaptive manoeuvre strategy

In this section, we first introduce the key ingredient of our RL environment. Then, we propose the adaptive manoeuvre strategy. Next, we describe the agent's flow, where the focus is on the neural network's structure. Finally, we present the MARL training algorithm based on a Rainbow DQN.

4.1. Reinforcement learning setup

As mentioned in the previous section, the agent's inputted state is a partial observation $\mathbf{o}_i^t$. Therefore, the action space, observation space, and reward function of the RL environment are defined as follows:

4.1.1. Action space

The action space comprises a set of flight intentions for resolving conflicts. There are nine actions in the discrete action space, represented by $a^t = \{1, \dots, 9\}$. Each number corresponds to a flight intention that includes two dimensions: speed intention and heading intention. The speed intention includes low speed, economical speed, and high speed, while the heading intention consists of left yaw, aiming, and right yaw. As such, a^t can also be represented by a two-tuple:

$$a^t = \left((a^H)^t, (a^S)^t \right) \quad (12)$$

$$(a^H)^t = \begin{cases} 1 & \text{left yaw} \\ 2 & \text{aiming} \\ 3 & \text{right yaw} \end{cases} \quad (13)$$

$$(a^S)^t = \begin{cases} 1 & \text{low speed} \\ 2 & \text{economical speed} \\ 3 & \text{high speed} \end{cases} \quad (14)$$

where $(a^H)^t$ and $(a^S)^t$ denote the flight intention of heading and speed, respectively. It is assumed in this study that high and low speeds are obtained by adding and subtracting a fixed speed from the economical speed, respectively, according to Equation 15 and Equation 16:

$$V^L = V^E - \Delta V \quad (15)$$

$$V^H = V^E + \Delta V \quad (16)$$

where V^L , V^E and V^H represent low, economical, and high speeds, respectively, and ΔV is a fixed value in this study. The adaptive manoeuvre strategy transfers the flight intention to a specific manoeuvre, namely the speed vector $\mathbf{v}^t$, as discussed in Section 4.2. It is important to note that this study selects the least number of actions in terms of the degree of dispersion of actions in the same dimension for three main reasons: (a) to reduce the size of the neural network and, thus, lower the training cost, (b) to improve the differentiation between different actions and, thus, reduce uncertainty, and (c) to enhance the smoothness of the trajectory.

4.1.2. Observation space

The observation $\mathbf{o}^t$ is obtained at the beginning of the time step t . As mentioned in Section 3.2 by Equation 1, $\mathbf{o}^t$ is composed of the state of the ownship $\mathbf{o}_O^t$ and the state of intruders $\mathbf{o}_I^t$. $\mathbf{o}_O^t$ comprises of the observation of the ownship's intention $\mathbf{o}_{oi}^t$ and performance $\mathbf{o}_{op}^t$:

$$\mathbf{o}_O^t = [\mathbf{o}_{oi}^t, \mathbf{o}_{op}^t] \quad (17)$$

As mentioned by Equation 2, $\mathbf{o}_I^t$ comprises the sub-observation of each detection sector $\mathbf{o}_d^t$. The mapping between the sub-observation $\mathbf{o}_d^t$ and the detection sector is shown in Figure 3. Detection sectors are always arranged in counter-clockwise order starting from the line between the ownship and its exit point (i.e., reference heading). The sub-observation matrices corresponds to the detection sectors in the same order. It is important to note the distinction between heading and reference heading in this paper:

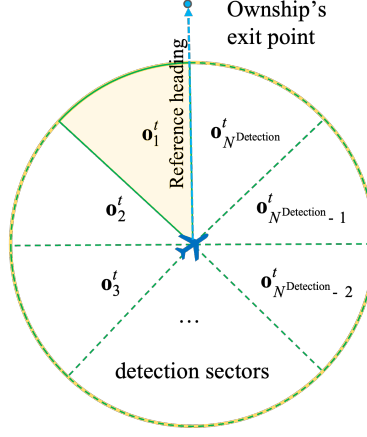


Figure 3: Mapping between sub-observation matrices and detection sectors. The detection sectors are always arranged in counter-clockwise order starting from the line between the ownship and its exit point. The sub-observation matrices correspond to the detection sectors in the same order.

- **Heading** refers to the direction in which the aircraft is currently flying, i.e., the direction pointed by the nose.
- **Reference heading** refers to the direction pointed by the aircraft to its exit point (further details can be found in [Section 4.2](#)).

$\mathbf{o}_d^t$ consists of the observation of the intruder's location $\mathbf{o}_{il}^t$, dynamic $\mathbf{o}_{id}^t$, intention $\mathbf{o}_{ii}^t$ and performance $\mathbf{o}_{ip}^t$:

$$\mathbf{o}_d^t = [\mathbf{o}_{il}^t, \mathbf{o}_{id}^t, \mathbf{o}_{ii}^t, \mathbf{o}_{ip}^t] \quad (18)$$

The sub-observations (i.e., $\mathbf{o}_{oi}^t$, $\mathbf{o}_{op}^t$, $\mathbf{o}_{il}^t$, $\mathbf{o}_{id}^t$, $\mathbf{o}_{ii}^t$ and $\mathbf{o}_{ip}^t$) are represented as following (refer to [Table 2](#) and [Figure 4](#) for variable descriptions and schematics):

$$\mathbf{o}_{oi}^t = [\|\mathbf{p}_{\text{Ownship}}^t - \mathbf{p}_{\text{Ownship}}^{\text{Exit}}\|, a_{\text{Ownship}}^{t-1}] \quad (19)$$

$$\mathbf{o}_{op}^t = [V_{\text{Ownship}}^E] \quad (20)$$

$$\mathbf{o}_{il}^t = \left[d_{\text{Intru}}^t, \left\langle \overrightarrow{\mathbf{p}_{\text{Ownship}}^t \mathbf{p}_{\text{Ownship}}^{\text{Exit}}}, \overrightarrow{\mathbf{p}_{\text{Ownship}}^t \mathbf{p}_{\text{Intru}}^t} \right\rangle \right] \quad (21)$$

Table 2: Vectors of sub-observations.	
Vectors	Contained factor(s)
ownship intention $\mathbf{o}_{oi}^t$	distance between the ownship and its exit $\ \mathbf{p}_{Ownship}^t - \mathbf{p}_{Ownship}^{Exit}\ $ last action of the ownship $a_{Ownship}^{t-1}$
ownship performance $\mathbf{o}_{op}^t$	economical speed of the ownship $V_{Ownship}^E$
intruder location $\mathbf{o}_{il}^t$	intrusion distance d_{Intru}^t intrusion angle $\langle \overrightarrow{\mathbf{p}_{Ownship}^t \mathbf{p}_{Ownship}^{Exit}}, \overrightarrow{\mathbf{p}_{Ownship}^t \mathbf{p}_{Intru}^t} \rangle$
intruder dynamic $\mathbf{o}_{id}^t$	current reference heading of the intruder $\overrightarrow{\mathbf{p}_{Intru}^t \mathbf{p}_{Intru}^{Exit}}$
intruder intention $\mathbf{o}_{ii}^t$	distance between the intruder and its exit $\ \mathbf{p}_{Intru}^t - \mathbf{p}_{Intru}^{Exit}\ $ last action of the intruder a_{Intru}^{t-1}
intruder performance $\mathbf{o}_{ip}^t$	economical speed of the intruder V_{Intru}^E

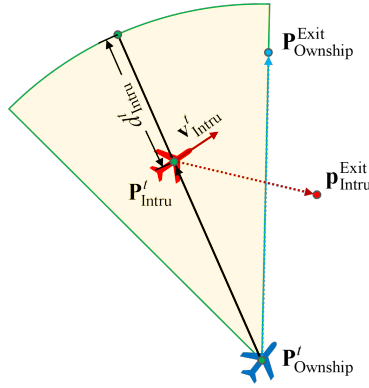


Figure 4: Symbol schematic in the observation of a detection sector.

$$\mathbf{o}_{\text{id}}^t = \left[\overrightarrow{\mathbf{p}_{\text{Intru}}^t \mathbf{p}_{\text{Intru}}^{\text{Exit}}} \right] \quad (22)$$

$$\mathbf{o}_{\text{ii}}^t = [\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Intru}}^{\text{Exit}}\|, a_{\text{Intru}}^{t-1}] \quad (23)$$

$$\mathbf{o}_{\text{ip}}^t = [V_{\text{Intru}}^E] \quad (24)$$

445 where $\mathbf{p}_{\text{Ownship}}^t$ and $\mathbf{p}_{\text{Intru}}^t$ are the positions of the ownship and the intruder at time step t , respectively; $\mathbf{p}_{\text{Ownship}}^{\text{Exit}}$ and $\mathbf{p}_{\text{Intru}}^{\text{Exit}}$ are the exit points of the ownship and intruder; d_{Intru}^t is the intruder distance:

$$d_{\text{Intru}}^t = R^{\text{Detection}} - \|\mathbf{p}_{\text{Ownship}}^t - \mathbf{p}_{\text{Intru}}^t\| \quad (25)$$

Therefore, $\mathbf{o}^t$ comprises $(6 \times N^{\text{Detection}} + 3)$ elements in total. If there is no intruder in the d th detection sector, $\mathbf{o}_d^t$ is a zero vector. It is important
 450 to note that, regardless of the aircraft's current heading, the reference heading $\overrightarrow{\mathbf{p}^t \mathbf{p}^{\text{Exit}}}$ is used as the forward direction for constructing the detection sectors. For instance, we start with the reference heading and number the detection sectors counterclockwise. As this study assumes that the speed interval of the aircraft is predetermined (refer to [Equation 15](#) and [Equation 16](#)), the agent only
 455 needs to observe the aircraft's economical speed to determine the low and high speeds. The intrusion distance d_{Intru}^t is used in $\mathbf{o}_{\text{ii}}^t$ instead of the separation distance $\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Ownship}}^t\|$ in order to maintain consistency of values in the conflict-free case for ease of agent identification (for more explanation, refer to [Appendix A](#)). The ownship only observes the intruder's current reference
 460 heading rather than the actual heading $\mathbf{v}_{\text{Intru}}^t$ for two reasons:

1. the current reference heading is the most desired heading of the aircraft (since the optimisation objective is to minimise the flight distance), and
2. the position of the aircraft's exit point can be determined by reference heading and the distance of the aircraft from the exit point (therefore,
 465 the ownship does not need to observe the position of the intruder's exit point).

In this study, we aim to reduce the number of elements in the observation model as much as possible in order to minimise the size of the neural network and hence the cost of training. For instance, in order to understand the intentions of the intruder, the agent would ideally need to observe the coordinates of the intruder’s exit point (two elements, e.g., (p_x, p_y)). However, since the agent already knows the intruder’s position and heading by using $\mathbf{o}_{ij}^t$ and $\overrightarrow{\mathbf{p}_{\text{Intru}}^t \mathbf{p}_{\text{Intru}}^{\text{Exit}}}$, it is sufficient for the agent to additionally observe the distance between the intruder and its exit $\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Intru}}^{\text{Exit}}\|$ (only one element).

4.1.3. Reward design

Since non-stationarity is challenging to avoid in multi-agent reinforcement learning systems [36], and flight uncertainty is considered in this study, the proposed method is expected to have greater robustness. Therefore, each aircraft is equipped with a protected area and a warning area (as shown in Figure 5). The radii of the warning and protected areas are denoted as R^{Warning} and $R^{\text{Protected}}$ respectively, where $R^{\text{Protected}} = D^{\text{Horizontal}}$. The warning area aims to provide a buffer for aircraft that are on the verge of losing safe separation.

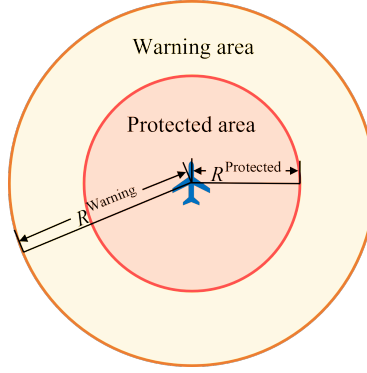


Figure 5: Warning area and protected area. The warning area provides a buffer to manage flight uncertainty and performance challenges resulting from the inherent non-stationarity in MARL. The radius of the protected area $R^{\text{Protected}}$ is typically the same as minimum safe separation. The radius of the warning area R^{Warning} can be set according to the situation and will be discussed in more detail in Section 5.4.

As discussed in [Section 3.1](#), the proposed method aims to minimise the average additional flight distance of all aircraft while avoiding conflicts. A reward function has been developed to guide the aircraft towards achieving this objective, which is expressed as:

$$r_i^t = (r^{\text{Approaching}})_i^t + (r^{\text{Warning}})_i^t + (r^{\text{Protected}})_i^t + (r^{\text{Economical}})_i^t + (r^{\text{Smooth}}_{\text{Speed}})_i^t + (r^{\text{Smooth}}_{\text{Heading}})_i^t \quad (26)$$

r_i^t denotes the reward received by the i th aircraft at the t th time step. r_i^t is composed of six parts, $(r^{\text{Approaching}})_i^t$, $(r^{\text{Warning}})_i^t$, $(r^{\text{Protected}})_i^t$, $(r^{\text{Economical}})_i^t$, $(r^{\text{Smooth}}_{\text{Speed}})_i^t$ and $(r^{\text{Smooth}}_{\text{Heading}})_i^t$. The reward function is designed based on four considerations.

1. Efficient in approach to the exit (evaluated by $(r^{\text{Approaching}})_i^t$): Aircraft should fly efficiently towards the exit, such as flying straight to the exit, unless manoeuvres are required to avoid conflicts.
2. Maintenance of safe separation (evaluated by $(r^{\text{Warning}})_i^t$ and $(r^{\text{Protected}})_i^t$): Aircraft should avoid entering warning and protected areas to ensure flight safety.
3. Economy of flight (evaluated by $(r^{\text{Economical}})_i^t$): Aircraft are encouraged to fly at economical speeds to reduce fuel consumption.
4. Trajectory smoothness (evaluated by $(r^{\text{Smooth}}_{\text{Speed}})_i^t$ and $(r^{\text{Smooth}}_{\text{Heading}})_i^t$): Frequent heading and speed changes are generally undesirable in air traffic management, considering factors such as traffic stability and passenger comfort. Therefore, aircraft are encouraged to maintain a stable flying state, where the actions of adjacent time steps of an agent remain consistent.

In particular, the aircraft is awarded by $(r^{\text{Approaching}})_i^t$ for approaching its exit, where $V_i^{\max}$ is the given maximum speed of aircraft i :

$$(r^{\text{Approaching}})_i^t = \frac{\|\mathbf{p}_i^{t-1} - \mathbf{p}_i^{\text{Exit}}\| - \|\mathbf{p}_i^t - \mathbf{p}_i^{\text{Exit}}\|}{V_i^{\max} \Delta t} \quad (27)$$

When the aircraft enters into the warning areas of other aircraft, it is warned

by $(r^{\text{Warning}})_i^t$:

$$(r^{\text{Warning}})_i^t = \begin{cases} R_W & \|\mathbf{p}_i^t - \mathbf{p}_j^t\| \leq R^{\text{Warning}} \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

When the aircraft violates the separation criteria from other aircraft, it is heavily
 510 penalised by $(r^{\text{Protected}})_i^t$:

$$(r^{\text{Protected}})_i^t = \begin{cases} R_P & \|\mathbf{p}_i^t - \mathbf{p}_j^t\| \leq R^{\text{Protected}} \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

To promote the economic operation of aircraft, a minor penalty $(r^{\text{Economic}})_i^t$
 is introduced to discourage the uneconomical speed:

$$(r^{\text{Economic}})_i^t = \begin{cases} 0 & (a^S)_i^t = 2 \\ R_E & \text{otherwise} \end{cases} \quad (30)$$

To encourage smooth manoeuvring of aircraft, two minor penalties, $(r^{\text{Smooth}}_{\text{Speed}})_i^t$
 and $(r^{\text{Smooth}}_{\text{Heading}})_i^t$, are introduced to discourage excessive changes in speed and
 515 heading, respectively:

$$(r^{\text{Smooth}}_{\text{Speed}})_i^t = \begin{cases} 0 & (a^S)_i^t = (a^S)_i^{t-1} \\ R_S^S & \text{otherwise} \end{cases} \quad (31)$$

$$(r^{\text{Smooth}}_{\text{Heading}})_i^t = \begin{cases} 0 & (a^H)_i^t = (a^H)_i^{t-1} \\ R_H^S & \text{otherwise} \end{cases} \quad (32)$$

Only $(r^{\text{Approaching}})_i^t$ represents the positive feedback in all six components
 of the reward function. It is not necessary to give a large positive reward when
 the aircraft reaches its exit, as it will always reach its exit using the adap-
 520 tive manoeuvre strategy (refer to [Section 4.2](#)). Representing $(r^{\text{Approaching}})_i^t$
 as a ratio has the advantage of providing a more stable quantitative relation-
 ship between $(r^{\text{Approaching}})_i^t$ and the other components of the reward when
 the trained agent is deployed on different aircraft. This makes it suitable
 for policy sharing. Additionally, if the trained agent is deployed on a new

525 set of aircraft with speed performances different from those in the training scenarios, it is unnecessary to adjust the reward function parameters. The weights of the five negative feedback components are designed based on their importance and the value of $(r^{\text{Approaching}})_i^t$. Since the aircraft's distance is a hard constraint, large weights are assigned to $(r^{\text{Protected}})_i^t$ and $(r^{\text{Warning}})_i^t$ with
530 $R^{\text{Protected}} = -20$ and $R^{\text{Warning}} = -5$. As economy and smoothness are soft constraints, small weights are assigned to $(r^{\text{Economical}})_i^t$, $(r^{\text{Smooth}}_{\text{Speed}})_i^t$ and $(r^{\text{Smooth}}_{\text{Heading}})_i^t$ for $R_E = -0.1$, $R_S^S = -0.1$ and $R_H^S = -0.2$. It is important to note that the weight coefficients can be fine-tuned based on air traffic control and airline operational preferences. This paper only provides a hypothetical example based
535 on feasible numerical relationships.

4.2. Adaptive manoeuvre strategy

The adaptive manoeuvre strategy comprises of the adaptive yaw angle and action supervisor, which are utilised to produce a specific manoeuvre based on the current state and flight intention of the aircraft. As mentioned earlier, the
540 flight intention of the aircraft includes low, economical, and high speeds. The values of these three speeds are influenced by several factors such as aircraft performance, airline preference, and aircraft load. Hence, this study assumes that each aircraft has three designated speeds, denoted as V^L , V^E and V^H for low speed, economical speed, and high speed, respectively.

545 At the start of the t th time step, the heading action is determined based on the reference heading to enable the aircraft to easily reach its exit (as illustrated in Figure 6). If the heading action is aiming, then the heading for the t th time step will be the same as $\overrightarrow{\mathbf{p}^t \mathbf{p}^{\text{Exit}}}$. In case of a left yaw or right yaw heading action, the heading will deflect by α^t to the left (anticlockwise) or to the right
550 (clockwise, that is, $-\alpha^t$) based on the reference heading, respectively. Therefore, the aircraft's turning angle at the t th time step is represented as $\langle \mathbf{v}^t, \mathbf{v}^{t-1} \rangle$, which is the change in heading from the $(t-1)$ th time step to the t th time step.

An action supervisor is employed to ensure smooth heading changes, with

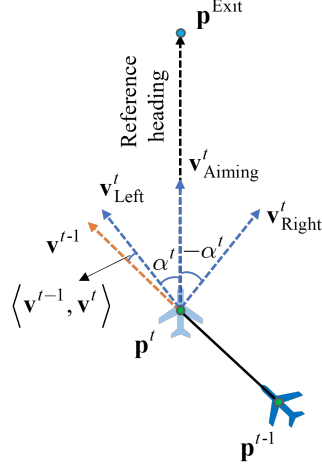


Figure 6: Heading manoeuvre based on the agent's action. For example, if the agent's heading action is yawing to the left, its target heading $\mathbf{v}^t$ is $\mathbf{v}_{\text{Left}}^t$. Suppose the heading of the last time step is $\mathbf{v}^{t-1}$. The turning angle of the current time step is $\langle \mathbf{v}^t, \mathbf{v}^{t-1} \rangle$.

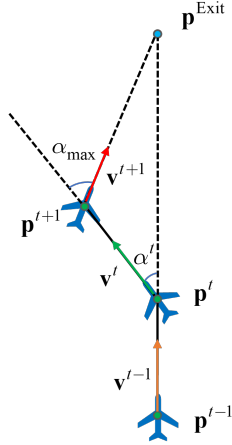


Figure 7: Adaptive yaw angle. The turning angle α^t is adaptive to guarantee that the subsequent turning angle can meet aircraft performance constraints.

555 the constraint $\left| (a^H)^t - (a^H)^{t-1} \right| \leq 1$. To effectively resolve conflicts, we assume that when the aircraft chooses to yaw, it always yaws to the maximum allowed angle. The maximum yaw angle $\alpha^{\max}$ is determined based on the aircraft's performance and flight distance for a time step. To ensure that $\langle \mathbf{v}^t, \mathbf{v}^{t-1} \rangle \leq \alpha^{\max}$ in all cases, we only need to consider the worst-case scenario, and we
 560 propose an adaptive yaw angle (as shown in Figure 7). When $(a^H)^{t-1} = 2$, $(a^H)^t = 1$, $(a^H)^{t+1} = 2$ and $(a^S)^t = 3$, $\langle \mathbf{v}^{t+1}, \mathbf{v}^t \rangle$ is the maximum yaw angle. Let $\langle \mathbf{v}^{t+1}, \mathbf{v}^t \rangle = \alpha_{\max}$ and the value of α^t can be calculated using the sine theorem as follows:

$$\alpha^t = \alpha^{\max} - \arcsin\left(\frac{\|\mathbf{p}^{t+1} - \mathbf{p}^t\| \times \sin \alpha^{\max}}{\|\mathbf{p}^{\text{Exit}} - \mathbf{p}^t\|}\right) \quad (33)$$

where $\|\mathbf{p}^{t+1} - \mathbf{p}^t\|$ can be expressed as:

$$\|\mathbf{p}^{t+1} - \mathbf{p}^t\| = \begin{cases} V^L \Delta t & (a^S)^t = 1 \\ V^E \Delta t & (a^S)^t = 2 \\ V^H \Delta t & (a^S)^t = 3 \end{cases} \quad (34)$$

565 Note that although the yaw angle decreases as the aircraft gets closer to the exit point, the adaptive yaw angle does not significantly reduce the aircraft's manoeuvrability in general (refer to Appendix B for more analysis).

Based on Equation 33, if we map a^t to $(a^S)^t$ and $(a^H)^t$ as:

$$a^t = (a^H)^t + 3 \times \left[(a^S)^t - 1 \right] \quad (35)$$

the schematic of the velocity vectors corresponding to the nine actions is shown
 570 in Figure 8, where the numbers present the value of a^t . It can be observed that the yaw angle α^t changes with the speed action due to the adaptive yaw angle. As the adaptive manoeuvre strategy ensures that the aircraft always reaches the exit point, agents only need to learn how to resolve conflicts effectively and efficiently.

575 4.3. Workflow and neural network

The agent's workflow is illustrated in Figure 9, highlighting the architecture of the neural network. Initially, the observation matrix $\mathbf{o}_i^t$, which is comprised of

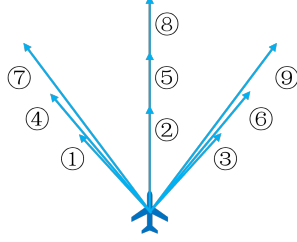


Figure 8: Velocity vectors schematic.

$\mathbf{o}_O^t$ and $\mathbf{o}_I^t$, is obtained through partial observation. Subsequently, it is flattened and used as the input layer of the neural network. The number of neurons in the input layer is equal to the number of elements in the observation $\mathbf{o}_i^t$, that is, $(6 \times N^{\text{Detection}} + 3)$. The first hidden layer, referred to as hidden layer 1, consists of $(2 \times N^{\text{Detection}} + 1)$ neurons. The second hidden layer comprises of hidden layers 2_A and 2_V, which correspond to the advantage function and value function, respectively, in the duelling DQN technology (refer to Section 4.4). Hidden layers 2_A and 2_V consist of nine neurons and one neuron, respectively. The output layer is composed of nine neurons, with the value of each neuron obtained by a linear combination of the corresponding neuron's value in hidden layer 2_A and the neuron's value in hidden layer 2_V (refer to Algorithm 1, line 9). The input layer and hidden layers 1, 2_A and 2_V are fully connected layers. The selected action a^t corresponds to the neuron with the maximum value in the output layer. Then, a^t is adjusted by the action supervisor (if necessary) and mapped to the heading action $(a^H)^t$ and speed action $(a^S)^t$. Based on these two decomposition actions, the current position $\mathbf{p}^t$, and the exit position $\mathbf{p}^{\text{Exit}}$, the velocity vector $\mathbf{v}^t$ is calculated by means of the adaptive yaw angle. The errors of flight uncertainty $[\varepsilon_h^t, \varepsilon_v^t]$ are then sampled and applied to the velocity vector $\mathbf{v}^t$. Subsequently, the velocity vector $\mathbf{u}^t$ is obtained and used to update the environment. Finally, the environment provides feedback on the reward r_i^t for updating the parameters of the neural network.

Note that the proposed method does not employ convolution layers, which are commonly used in ray-based detection models [34]. This is because ray de-

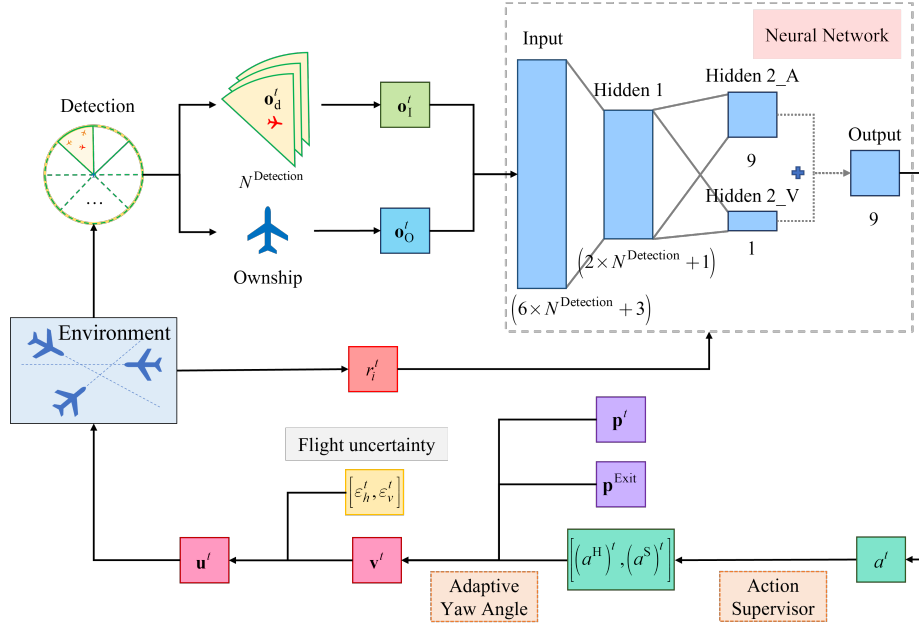


Figure 9: Workflow of a single agent, which serves as a part of Figure 1. The architecture of the neural network is highlighted. The formula or number below the blue box in the neural network area indicates the dimension of the corresponding layer.

tectors typically have many rays, resulting in an input layer with thousands of neurons. To extract data characteristics from the sparse matrix, one or more convolution layers are needed to reduce the input data dimension. In contrast, our sector-based detection model divides the detection area into decades of de-
605 tection sectors, resulting in an input layer with only hundreds of neurons and a less sparse input matrix than the ray-based detection model. Therefore, convolution layers are unnecessary for the proposed method, and one or two hidden layers are sufficient to achieve satisfactory results. The number of neurons in the fully connected layers is determined by referencing similar studies [34][40]. The
610 number of neural network layers is based on the basic Duelling DQN structure. Since the primary objective of this study is to demonstrate the feasibility of the proposed method, this paper only provides an acceptable parameter setting of the neural network as a reference for the reader.

4.4. Training algorithm

615 Because the proposed RL model’s action is discrete with only nine possible options, DQN is an ideal algorithm for efficient RL with discrete actions [41]. In recent years, several techniques for improving DQN performance have been proposed. We have adaptively employed and combined Double-DQN [42], Duelling DQN [43], Prioritised Replay Buffer [44], and Multi-Step Learning [45] to en-
620 hance the performance of DQN. These techniques were inspired by an advanced DQN method that was recently proposed, Rainbow DQN [35].

Algorithm 1 presents the proposed learning approach for MACR. Each episode in the algorithm consists of two parts. One is the simulation and collecting transitions, and the other is the network update.

625 During the simulation and collecting transitions, the simulation scenario is first reset (line 4). From the first time step (line 5), for all aircraft in the airspace, the value of each action $Q(\mathbf{o}_i^t, a; \theta)$ is evaluated based on duelling DQN (line 9) [43]. Next, action a_i^t with the greatest value is chosen by the ϵ -greedy method (line 10). Then, velocity vector $\mathbf{u}_i^t$ is calculated using the method mentioned in
630 Section 4.3 to update the position and state of aircraft i (line 11). The agents

deployed on each aircraft parallelly execute the procedure of lines 9, 10 and 11. After all the aircraft are updated, aircraft in the airspace collect reward r_i^t and the observation of next step $\mathbf{o}_i^{t+1}$ (line 15). The transition $(\mathbf{o}_i^t, a_i^t, r_i^t, \mathbf{o}_i^{t+1})$ is then stored in the buffer (line 16). The procedure of lines 15 and 16 is also
635 executed in parallel by each agent.

During the network update, transitions in the buffer are sampled using SumTree, one of the key technologies in Prioritised Replay Buffer (line 21) [44]. Then, the loss function L is calculated based on Double DQN [42] and Multi-Step Learning [45] (line 22). Next, evaluation network parameter θ is updated
640 by gradient descent every step (line 23), while target network parameter θ' is updated by copying parameters θ periodically (line 24) [42]. Finally, a trained neural network π_θ is obtained.

Algorithm 1 Learning for multi-aircraft conflict resolution

Input: The dataset of simulation scenarios, including flight schedule, aircraft
645 performances and airspace structure

- 1: Initialise evaluation network parameter θ and target network parameter θ' , and set hyper-parameters as shown in Table D.6
- 2: **for** episode = 1, 2, ..., **do**
- 3: # Simulation and collecting transitions
- 4: Reset the simulation scenario
- 5: $t \leftarrow 0$ # the first time step
- 6: **while** there is any aircraft not get its exit point **do**
- 7: **for** aircraft i in the airspace **do**
- 8: # in parallel
- 9: Evaluate the action value based on duelling DQN [43]: $Q(\mathbf{o}_i^t, a; \theta) =$
655 $V(\mathbf{o}_i^t) + A(\mathbf{o}_i^t, a) - \frac{\sum_{a' \in \mathbf{a}} A(\mathbf{o}_i^t, a')}{N_a}, a \in \mathbf{a}$
- 10: Choose (ϵ -greedy) the action with the greatest value [46]: $a_i^t =$
 $\arg \max_a Q(\mathbf{o}_i^t, \mathbf{a}; \theta)$
- 11: Calculate velocity vector $\mathbf{u}_i^t$ (refer to Section 4.3) to update the po-
660 sition and state of aircraft i
- 12: **end for**

```

13:   for aircraft  $i$  in the airspace do
14:       # in parallel
15:       Collect reward  $r_i^t$  and observation of next step  $\mathbf{o}_i^{t+1}$ 
665 16:       Store transition  $(\mathbf{o}_i^t, a_i^t, r_i^t, \mathbf{o}_i^{t+1})$  in the buffer whose capacity is  $n_{\text{BC}}$ 
17:   end for
18:    $t \leftarrow t + 1$  # next time step
19: end while
20: # Network update
670 21: Sample  $n_{\text{BS}}$  transitions by Prioritised Replay Buffer (SumTree) [44]
22: Calculate the loss function by Double DQN [42] and Multi-Step Learning
    [45]:

$$L = \left( (R^t)^{(n_{\text{MS}})} + \gamma^{n_{\text{MS}}} Q \left( \mathbf{o}^{t+n_{\text{MS}}}, \arg \max_{a^*} Q(\mathbf{o}^{t+n_{\text{MS}}}, a^*; \theta); \theta' \right) - Q(\mathbf{o}^t, a^t; \theta) \right)^2$$

    where  $(R^t)^{(n_{\text{MS}})} \equiv \sum_{k=0}^{n_{\text{MS}}-1} \gamma^k r^{t+k}$ 
675 23: Update evaluation network parameter  $\theta$ :  $\theta \leftarrow \theta - \alpha \frac{\partial l}{\partial \theta}$  # every step [42]
24: Update target network parameter  $\theta'$ :  $\theta' \leftarrow \theta$  # every  $n_{\text{TN}}$  steps [42]
25: end for
Output: Neural network  $\pi_\theta$ 

```

5. Simulation experiments

680 In this section, we will first introduce the experimental setup, including the generation of training scenarios. We will then describe four sets of experiments and analyse their results. The purpose of each set of experiments is as follows:

1. Section 5.2 is aimed at verifying the effectiveness of the training and identifying which manoeuvre (adjusting heading or adjusting speed or a combination of both) is better suited for the proposed method. 685
2. Section 5.3 is aimed at testing the proposed method's generalisation ability and obtaining its performance metrics in terms of conflict resolution success rate and extra flight distance.
3. Section 5.4 is aimed at exploring the impact of uncertainty levels and warning area sizes on the proposed method. 690

4. [Section 5.5](#) is aimed at comparing the differences between the proposed method and state-of-the-art RL-based conflict resolution methods.

5.1. Experimental setup

The training scenarios were designed based on the most commonly used MACR scenario [\[47\]](#). An example of the training scenarios in this study is shown in [Figure 10](#). The circular airspace, with a radius of 50 nautical miles, was divided evenly into several sections. The entry point of each aircraft was randomly generated on the arc of each section, with its entry and exit points being symmetrical about the circle’s centre. The reader is directed to [Appendix C](#) for more details about the training scenario parameters.

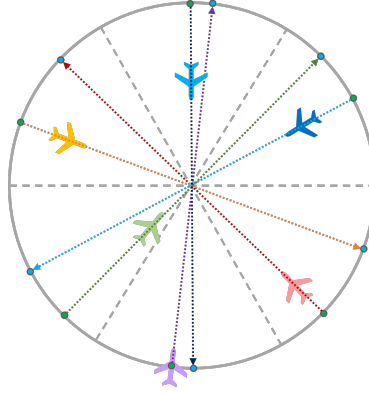


Figure 10: An example of training scenarios. The green and blue circles represent the entry and exit points of the aircraft, respectively. The planned trajectory of the aircraft is shown by the dotted lines. To ensure that the scenario is more general, the speed range of the aircraft and the position and time of entry into the sector are randomised.

The neural network was constructed using the [Pytorch](#) library in a Python environment and trained on a computer with an i5-8300H CPU. The hyperparameters were chosen based on related references and practical experience. The reader is directed to [Appendix D](#) for more details about RL hyperparameters. It should be noted that as a preliminary study to verify the feasibility of emerging technology, this paper only provides a set of acceptable hyper-

parameter settings for reference, with a systematic hyper-parameter study as further research work.

5.2. Training results

710 To verify the effectiveness of manoeuvres, we introduced and compared the training results of three manoeuvre models: MARL_HS, MARL_H and MARL_S. MARL_HS indicates that this method can use two manoeuvres to resolve conflicts, namely adjusting heading and speed (mentioned in [Section 4.1.1](#)). MARL_H and MARL_S can only adjust heading and speed, respectively. The training curves are presented in [Figure 11](#). Each neural network was trained for 715 ten thousand episodes. The average reward, which refers to the average reward of all actions performed by all aircraft in each episode, is used to evaluate the neural network’s performance:

$$r_j^{\text{Ave}} = \frac{\sum_i \sum_{t \in T_i^j} r_i^t}{\sum_i (n^{\text{A}})_i^j} \quad (36)$$

where r_j^{Ave} represents the average reward of the j -th episode, T_i^j denotes the 720 set of time steps of the i -th aircraft in the j -th episode, and $(n^{\text{A}})_i^j$ denotes the number of actions performed by the i -th aircraft in the j -th episode. It can be observed that all three models can converge in ten thousand episodes. The average rewards of MARL_HS and MARL_H increased rapidly in the first three thousand episodes and reached a steady at around episode eight thousand. 725 MARL_HS performed slightly better than MARL_H and was more stable. However, after the second thousand episodes, the performance of MARL_S had not improved significantly, which was unsatisfactory.

[Figure 12](#) displays selected samples from the first, fifth, and tenth thousand episodes of the training process, representing the early, intermediate, and final 730 stages of training, respectively. For MARL_HS and MARL_H, several conflicts cannot be resolved in the early stage of training, as shown in [Figure 12\(a\)](#) and [Figure 12\(d\)](#). During the intermediate stage of training, most conflicts can be resolved, but the aircraft’s trajectory deviates significantly from a straight line,

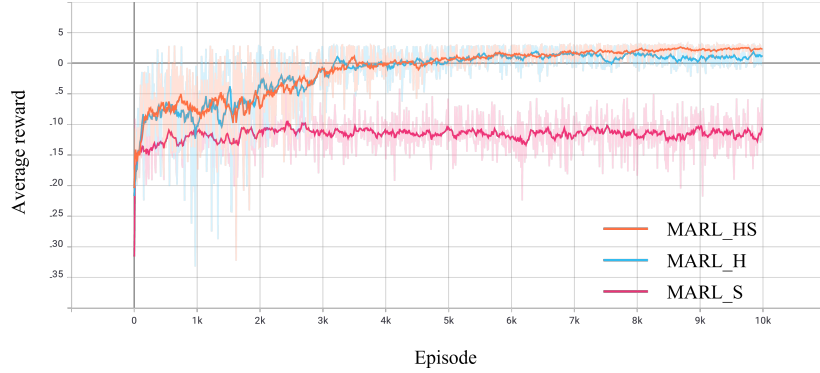


Figure 11: Training result. This figure is created using [TensorboardX](#), a Python toolkit for visualising machine learning data. The light curves represent the actual data obtained from the training process, while the dark curves show a 90% smoothed version of the light curves for ease of interpretation.

and extra flight distances are required to resolve the conflicts, as indicated in
735 [Figure 12\(b\)](#) and [Figure 12\(e\)](#). In the final stage of training, the advantage of
manoeuvrability allows MARL_HS to resolve all conflicts based on shorter extra
flight distances, as depicted in [Figure 12\(c\)](#). However, MARL_H still required
longer extra flight distances and is not as stable as MARL_HS, as shown in
[Figure 12\(f\)](#). Due to the high conflict density in the centre of the airspace and
740 most of the conflicts being concentrated in this area, adjusting the speed alone
was not sufficient to resolve all conflicts for MARL_S, as illustrated in (refer to
[Figure 12\(g\)](#), [Figure 12\(h\)](#) and [Figure 12\(i\)](#)).

In summary, for our method, the heading manoeuvre is far more crucial than
the speed manoeuvre, and the combination of the two yields the best results.

745 5.3. Performance test

We generated scenarios consisting of 2 to 10 aircraft in a random manner to
evaluate the generalisation of the trained neural network, and recorded its corre-
sponding performance indicators. The randomness of the scenarios encompassed
various factors, including aircraft performance (economical speed), aircraft en-
750 try and exit points, aircraft entry time, and operational flying uncertainties

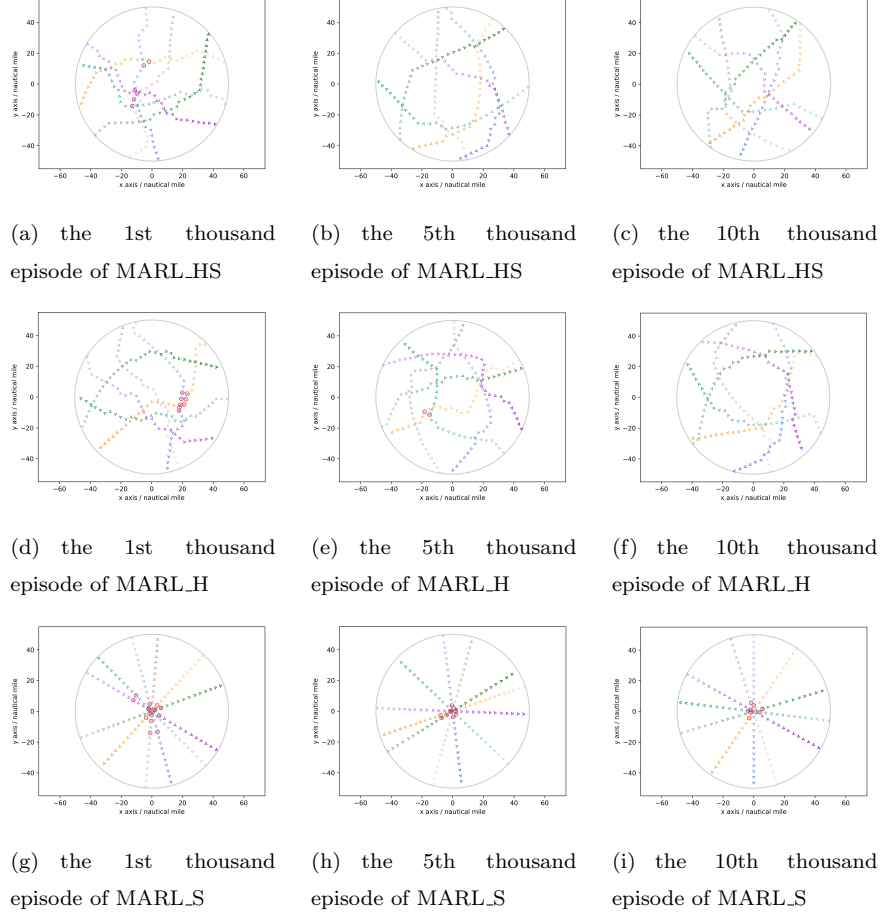


Figure 12: Training results sample. Each aircraft is represented by a different colour, and the direction of the arrow shows its movement direction at that time step. The transparency of the colours represents the sequence of time, with the more opaque colours indicating the later time steps. The arrow surrounded by a red circle indicates that the aircraft is closer than the minimum horizontal safety distance to another aircraft at that time step.

(heading and speed). To achieve a statistical assessment of the trained neural network's performance indicators, we generated each type of scenario (with a distinct number of aircraft) 100 times, resulting in 900 scenarios in total. The trained neural network was deployed on all the aircraft in each scenario, which were subsequently simulated, and the final result was tallied (as illustrated in Figure 13).

Figure 13(a) presents the success rate of conflict resolution for the trained neural network in scenarios with different numbers of aircraft. A successful resolution was recorded when all the conflicts in a scenario had been resolved. The success rate refers to the proportion of successful resolutions to the total number of tests. The success rate of conflict resolution was 100 percent in our scenarios where the number of aircraft is less than or equal to eight. Therefore, the airspace density that this method can handle (i.e., eight aircraft) is much higher than the highest density that exists in current practical operations (i.e., four or five aircraft) [14]. When the number of flights in the scenario increased to ten, the success rate was still able to reach 85 percent, although such a density scenario is unlikely to occur in the real world. Moreover, a failure to resolve a conflict by the neural network does not necessarily imply a collision occurred, only that the flight lost the safe separation from other flights. The flight can defuse the danger through the onboard collision avoidance program, such as the traffic collision avoidance system (TCAS) [48].

Figure 13(b) presents the average number of changes in flight states per aircraft, with the colour area indicating the corresponding indicator's error range (standard deviation). We define three types of changes as follows:

1. Speed change: the change of flight intention in speed (i.e., every time $(a^S)^{t+1} \neq (a^S)^t$, the speed change increments by 1).
2. Heading change: the change of flight intention in heading (i.e., every time $(a^H)^{t+1} \neq (a^H)^t$, the heading change increments by 1).
3. Action change: the change of flight intention in both heading and speed (i.e., every time $a^{t+1} \neq a^t$, the action change increments by 1).

These three indicators reflect the stability of air traffic to some extent. They increase with the number of aircraft, with the growth rate becoming faster for 2-4 aircraft, then slower for 4-7 aircraft, and then faster again for 7-10 aircraft. When the airspace density is low, there are few or no potential conflicts, and aircraft can resolve conflicts by only adjusting their speed or heading. As airspace density increases, aircraft have to manoeuvre frequently to resolve conflicts. Note that when an aircraft changes heading, the number of heading changes is usually greater than or equal to 2 because our model considers trajectory recovery. To further To further evaluate the acceptability of the number of flight state changes, we define a ‘conflict pair’ as two flights with potential conflicts (regardless of the conflict’s number and duration). For example, in scenarios with eight aircraft, the average number of conflict pairs is about 24.56 (meaning that each aircraft is involved in an average of 6.14 conflict pairs), while the average number of action changes per aircraft is about 8.41. We can roughly consider that each flight requires $8.41 \div 6.14 \approx 1.37$ changes in flight states to resolve a conflict pair. Considering extremely high-density scenarios, although this study focuses on autonomous operation, the average number of changes in flight states seems acceptable in current real operations.

Figure 13(c) and Figure 13(d) display the total and average extra flight distances, respectively. In our scenarios, the shortest flight distance for each aircraft is 100 nautical miles, which is the straight-line flight distance. Therefore, the extra flight distance represents the distance flown by an aircraft exceeding 100 nautical miles during testing. As depicted in Figure 13(c) and Figure 13(d), the extra flight distance increased with increasing airspace density. Even in extremely high-density scenarios with ten aircraft, the average extra flight distance of the aircraft only increased by about 15%. In comparison, this increase was roughly 8% in the scenario with four aircraft, whose density is similar to the peak of the current real world.

Figure 14 displays sample trajectories from the test results. As observed in the sample trajectory for the scenario with two aircraft, conflicts were resolved by adjusting speed only. In low-density airspace scenarios, it is often possible

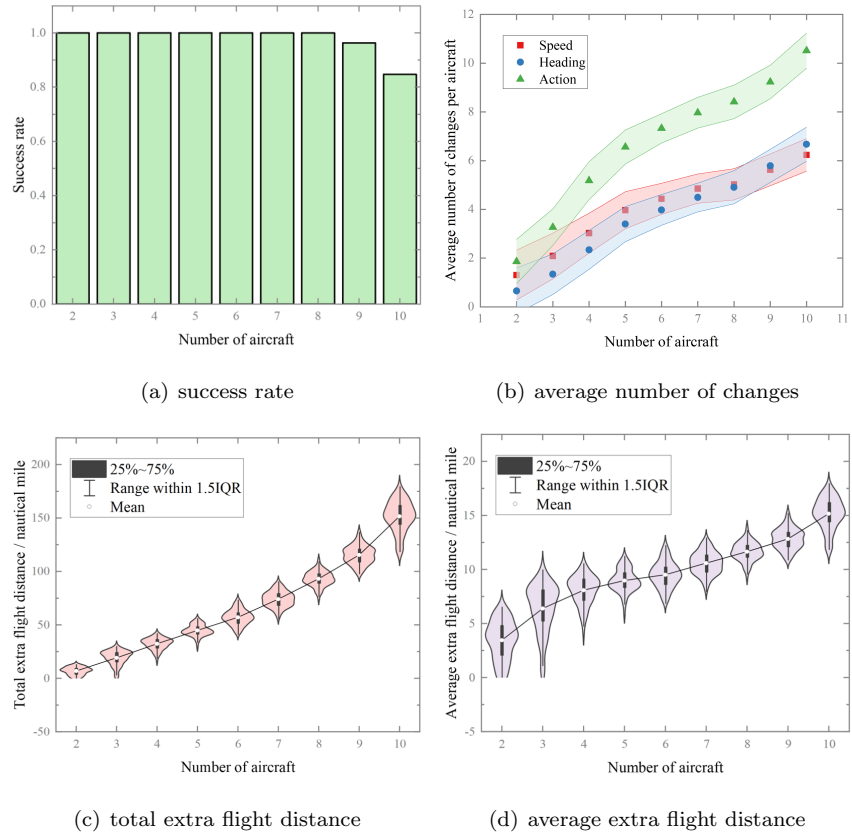


Figure 13: Statistics of test result.

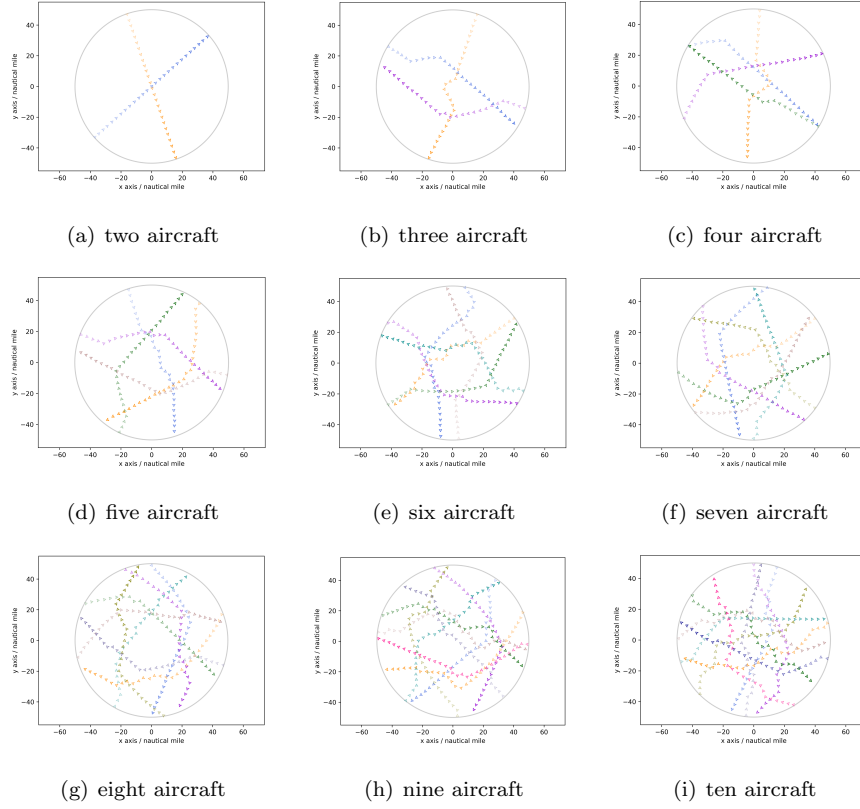


Figure 14: Test results sample. Each aircraft is represented by a different colour, and the direction of the arrow shows its movement direction at that time step. The transparency of the colours represents the sequence of time, with the more opaque colours indicating the later time steps.

for aircraft to resolve conflicts by adjusting their speed. However, as airspace density increases, the chances of resolving conflicts by adjusting speed alone decrease, and aircraft have to adjust their heading to resolve conflicts. Additionally, aircraft need to adjust their heading more frequently to resolve conflicts as airspace density increases. This explains why the average extra flight distance growth initially slows down, and then speeds up as airspace density increases (refer to [Figure 13\(d\)](#)).

It should be noted that in cases where aircraft are able to resolve conflicts by adjusting their speed alone, they also have the potential to change their heading. For example, among the 100 two-aircraft scenarios in the test, there were more than twenty scenarios where conflicts could be resolved by adjusting speed alone, but only six did not involve a heading change. There are three possible explanations for this:

1. In order to avoid penalties from the warning area, the aircraft increased their margin for safe separation (refer to [Equation 28](#)).
2. The speed manoeuvre may not be able to achieve the aircraft's economic goals (refer to [Equation 30](#)).
3. Since the agents are distributed in the proposed method, one aircraft cannot accurately predict the actions of other aircraft, so it may adopt a conservative policy.

Furthermore, since the computation time for a single manoeuvre in this distributed system is very low (typically less than 0.001 seconds), we did not perform statistical analysis of the computational performance in this paper.

In summary, we conducted numerous experiments in random scenarios to statistically evaluate the performance indicators of the proposed method. The results demonstrated that our method has a very high success rate of conflict resolution with acceptable extra flight distance, even in high-density scenarios.

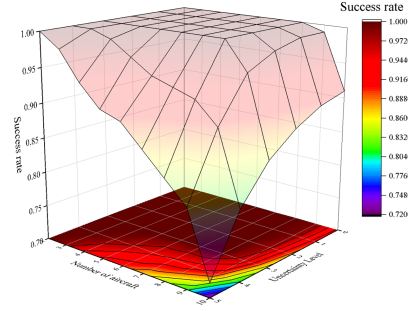
5.4. Sensitivity analysis

A sensitivity analysis was conducted to investigate the impact of uncertainty levels and warning area sizes on the proposed method. The uncertainty level

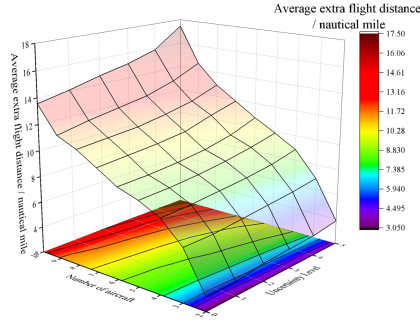
and warning area size were each divided into six categories, as presented in Table 3. The interval of uncertainty levels was set linearly, with reference to state-of-the-art research on flight uncertainty [7] [39]. The interval of warning area sizes was set based on accuracy values for alerting at different performance-based navigation (PBN) levels [49]. Level 0 of uncertainty corresponds to the absence of uncertainty. Likewise, size 0 of the warning area corresponds to the absence of a warning area because, in this case, the radius of the protected area is equal to the radius of the warning area (see Figure 5). The model used in Section 5.3 is the base model, with uncertainty level 3 and warning area size 3 (see Table C.5). In addition to the base model, we trained another ten models with different combinations of uncertainty levels and warning area sizes (i.e., uncertainty levels 0, 1, 2, 4, 5 with warning area size 3, respectively, and warning area sizes 0, 1, 2, 4, 5 with uncertainty level 3, respectively), while keeping other parameters constant as shown in Table C.5. The test scenarios for each model were identical to its own training scenarios.

Sensitivity analysis of uncertainty levels: Figure 15 presents the test results for the sensitivity analysis of uncertainty levels. As the uncertainty level increased, the extra flight distance decreased (refer to Figure 15(b) and Figure 15(d), while Figure 15(c) and Figure 15(e) show the corresponding variances). This is highly likely due to the aircraft taking more conservative actions to improve the policy’s robustness. However, the success rate of conflict resolution decreased with the increased uncertainty levels (refer to Figure 15(a)). Especially when the uncertainty level exceeded level 3, the success rate decreased at a faster pace. In this case, the uncertainty level probably exceeded the warning area’s buffering capacity ($R^{\text{Warning}} = 8$). Therefore, while training the neural network, if the success rate of conflict resolution is not ideal due to high uncertainty, it is recommended to increase the size of the warning area appropriately.

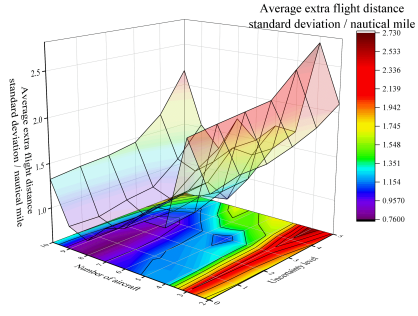
Sensitivity analysis of warning area sizes: The test results for the sensitivity analysis of warning area sizes are shown in Figure 16. It can be observed that as the warning area size increased, the trained agent acted more cautiously



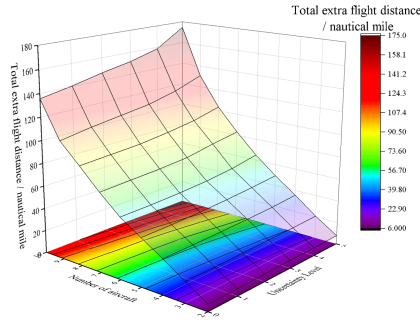
(a) success rate



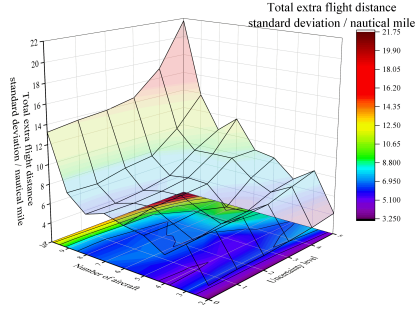
(b) average extra flight distance



(c) standard deviation of the average extra flight distance

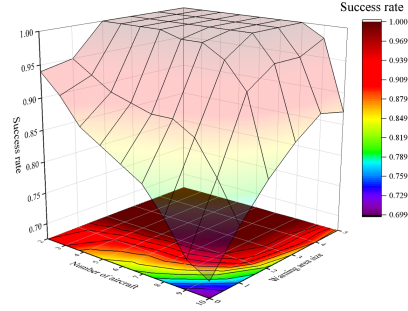


(d) total extra flight distance

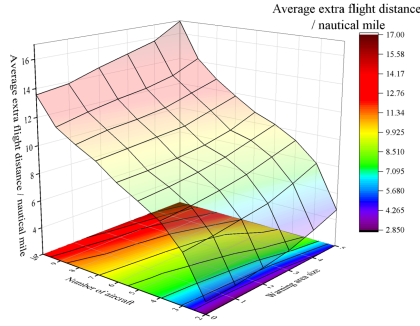


(e) standard deviation of the total extra flight distance

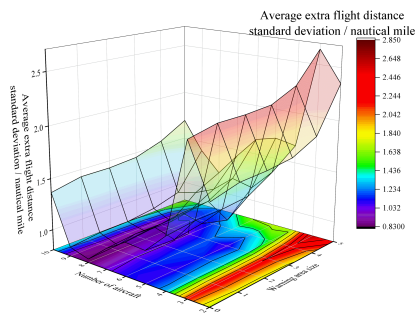
Figure 15: Sensitivity analysis of uncertainty levels.



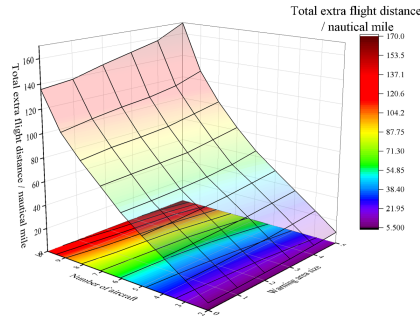
(a) success rate



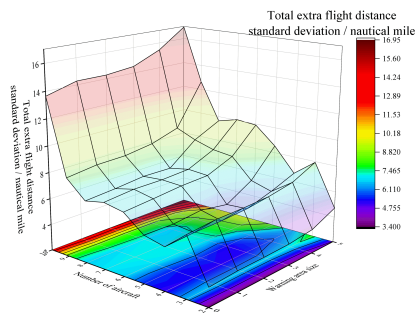
(b) average extra flight distance



(c) standard deviation of the average extra flight distance



(d) total extra flight distance



(e) standard deviation of the total extra flight distance

Figure 16: Sensitivity analysis of warning area sizes.

to avoid the penalty of the warning area, leading to an increase in the conflict resolution success rate (refer to Figure 16(a)) and the extra flight distance (refer to Figure 16(b) and Figure 16(d), while Figure 16(c) and Figure 16(e) show the corresponding variances). It is worth noting that the increase in the extra flight distance caused by the increase in the size of the warning area was more rapid in low-density airspace than in high-density airspace. Generally speaking, the lower the airspace density, the fewer potential conflicts, and the easier they are to resolve. Once the warning area has reached a certain size, its further expansion is progressively less beneficial for enhancing safety. Furthermore, an oversized warning area may result in high and unnecessary extra flight costs (refer to the results where the warning area size is greater than 3 in Figure 16).

Table 3: Parameter settings for sensitivity Analysis.

Uncertainty			Warning area	
Level	Heading error	Speed error	Size	Radius (nautical mile)
0	none	none	0	5
1	$[-1^\circ, 1^\circ]$	$[-2\%, 2\%]$	1	6
2	$[-2^\circ, 2^\circ]$	$[-4\%, 4\%]$	2	7
3	$[-3^\circ, 3^\circ]$	$[-6\%, 6\%]$	3	8
4	$[-4^\circ, 4^\circ]$	$[-8\%, 8\%]$	4	9
5	$[-5^\circ, 5^\circ]$	$[-10\%, 10\%]$	5	10

In summary, using the same neural network, resolving conflicts becomes increasingly difficult as the uncertainty level of the aircraft increases. Increasing the warning area size improves the ability to handle higher uncertainty levels, but excessively large warning areas result in unnecessary extra flight distances. Therefore, the warning area size should be set to match the uncertainty level in the model. In practical applications, it may be necessary to consider the actual performance-based navigation (PBN) levels [49]. For example, if the practical PBN level is RNAV 1 (with an accuracy value for alerting of 1 nautical mile on continental En-route) and the warning area size should be set to size 2 (7

nautical miles), then the warning area size could be set to size 3 (8 nautical miles) in continental areas.

895 5.5. Method comparison

Our proposed method was compared with three state-of-the-art methods: MARL_P_U [7], MARL_SSD [13] and MARL_MP [8]. These methods were named based on their characteristics for clarity, where P_U refers to selecting one rerouting point and considering uncertainty, SSD refers to using a solution
900 space diagram, and MP refers to selecting multiple rerouting points. Implementing a fair comparison experiment of RL methods can be challenging. To make our comparative experiments as informative as possible, we carefully designed the experiments, and the necessary clarifications for the experimental setup are as follows:

- 905 1. The training process of the RL methods has a random nature. The source codes and the trained models of the three state-of-the-art methods are not publicly available. Therefore, replicating their models based on the description in the papers could result in inconsistency, reducing the credibility of the comparison experiments. In this study, we only tested our
910 methods with their experimental scenarios. This way, we can compare the performance indicators of our methods from the tests with the performance data published in their papers.
2. The comparison experiments focus on the methods' effectiveness (i.e., the success rate of conflict resolution) and efficiency (i.e., extra flight distance).
915 However, only some of performance data of interest of the three state-of-the-art methods are publicly available:
 - MARL_P_U only disclosed the success rate of conflict resolution but not the extra flight distance.
 - MARL_SSD published the average number of conflicts per two hours
920 flight as an indicator to quantify the model's effectiveness but did not disclose the extra flight distance.

- MARL_MP only disclosed the extra flight distance but not the success rate of conflict resolution.

Therefore, we only compared our method to the publicly available data of the three state-of-the-art methods.

3. The experimental scenarios of the three state-of-the-art methods were generated randomly. For example, in MARL_P_U's scenarios, the start and end points of the aircraft (both ownship and intruders) were generated randomly. Therefore, randomness is inevitable if we want to test our method in their scenarios. To mitigate the errors caused by randomness, we performed a large number of tests to obtain performance indicators statistically.
4. Probably unsolvable scenarios may occur due to the random position of the aircraft in the test scenario (refer to Figure 17 for an example of MARL_P_U's scenarios where intruders do not manoeuvre). Although proving theoretically whether a scenario is solvable is challenging, it is unlikely that a similar scenario to Figure 17 would occur in the real world. We have no way of knowing whether they have avoided creating unsolvable

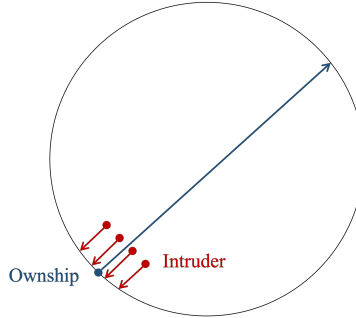


Figure 17: An example of probably unsolvable scenarios. Only the ownship can manoeuvre according to MARL_P_U's scenario setting.

scenarios (e.g., the start points of any two aircraft are set to at least 20 nautical miles apart) or removed such unlikely scenarios from their tests. Therefore, to maintain fairness (preferring to have a potential disadvantage for us), we allow potentially unsolvable scenarios to arise.

5. Parameters for comparative experiments can be found in Table 4, which shows the features of each method in the corresponding comparison experiment.

945

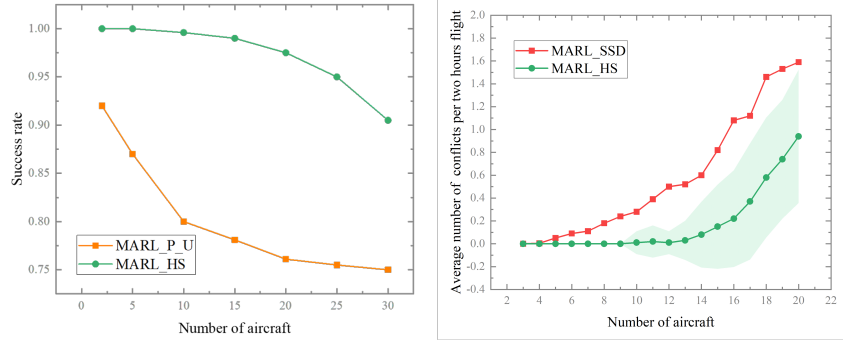
Table 4: Parameters for comparative experiments

State-of-the-art methods	Agents	Aircraft controlled	Uncertainty	MARL_HS	
				Uncertainty level	Warning area size
MARL_P_U	Single	Ownship	Yes	5	5
MARL SSD	Multiple	All	No	0	1
MARL_MP	Single	Ownship	No	0	1

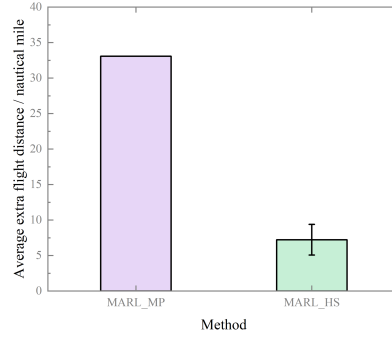
To make the comparison experiments more informative, we had to replicate their models or generate their test scenarios. As previously mentioned, recreating and retraining their models can result in inconsistency between the trained models, while generating their test scenarios can lead to inconsistencies in the test scenarios. As the error resulting from the inconsistency of test scenarios can be reduced by conducting a sufficient number of random tests, we opted to generate their test scenarios instead of retraining their models.

Comparison to MARL_P_U: Due to the uncertainty parameter being set to 10% in MARL_P_U, we adjusted the uncertainty level and the warning area size in our method to level 5 and size 5, respectively, and re-trained the neural network. It should be noted that the uncertainty model of the proposed method differs from that of MARL_P_U. Specifically, MARL_P_U’s uncertainty is static, which means it calculates uncertainty only twice (i.e., at yaw and return points). On the other hand, our method’s uncertainty is dynamic, and it calculates uncertainty at every time step, at least 30 times in total. Moreover, our method takes the speed error into account, while MARL_P_U does not. Numerically, MARL_P_U corresponds to having a maximum heading error of approximately 5.7 degrees in total, while the proposed method has a maximum heading error of about 5 degrees for each time step, at least 30 time steps in total. Therefore, the proposed method could be regarded as being tested under a ‘worse-case’

965



(a) result of the comparative experiment with MARL_P_U in the success rate
(b) result of the comparative experiment with MARL_SSD in the average number of conflicts



(c) result of the comparative experiment with MARL_MP in the average extra flight distance

Figure 18: Results of comparative experiments with three state-of-the-art RL methods. The data from the three state-of-the-art RL methods are taken from their published figures supported by [OriginLab](#), a software for graphing and analysis.

scenario of uncertainty compared to MARL_P_U. Additionally, since the agent in MARL_P_U only controlled one aircraft, we deployed our trained neural network in only one aircraft (refer to Table 4). As shown in Figure 18(a), our proposed method showed better results than MARL_P_U in terms of the success rate. This
970 is mainly because MARL_P_U only resolved conflicts by selecting a rerouting point, whereas our method involved time step manoeuvring (with an interval of 30 seconds), resulting in a much larger solution space for our method than for MARL_P_U. Furthermore, this experiment illustrated that our method can handle aircraft that cannot communicate and only fly in straight lines.

Comparison to MARL_SSD: In the comparative experiment between
975 MARL_SSD and our method, we set the uncertainty level to 0 in our method because MARL_SSD does not consider flight uncertainty (refer to Table 4). The warning area size was adjusted to 1 instead of 0 in our method as the warning area is used to handle not only flight uncertainty but also performance challenges
980 posed by the intrinsic non-stationarity in MARL. We then re-trained the neural network. Our method demonstrated better conflict resolution performance in the comparative test scenarios than MARL_SSD (as shown in Figure 18(b), where the green area represents the corresponding standard deviation). Both methods were able to resolve most conflicts in the test scenarios, particularly in
985 low-density scenarios. However, our method’s success rate in resolving conflicts was closer to 100% than that of MARL_SSD. This demonstrates that our method may have a higher potential for practical application.

Comparison to MARL_MP: The MARL_MP method also does not take into account flight uncertainty. Therefore, we used the same trained neural
990 network as with the comparison to MARL_SSD (refer to Table 4). While MARL_MP selects multiple rerouting points to resolve conflicts, MARL_P_U only selects one. We compared our method with the highest-performing method in MARL_MP (which chooses three rerouting points). Our method reduced the average extra flight distance by about 78% compared to MARL_MP (as shown
995 in Figure 18(c), where the error bar is based on the standard deviation). The experimental results in Figure 18(a) and Figure 18(c) demonstrate that con-

tinuous manoeuvring outperforms single or multiple manoeuvring in terms of conflict resolution success rate and extra flight distance.

In summary, based on limited publicly available data, we compared our
1000 method to three state-of-the-art methods. We fine-tuned and re-trained our
model in these experiments to ensure comparability with the scenarios or meth-
ods being compared. The proposed method achieved more competitive results
than the three state-of-the-art methods with regard to the featured performance
indicators. However, it should be noted that although we endeavoured to make
1005 the experiment fair or unfavourable to our method, the comparison experiment’s
environment is still limited. For example, although the parameters of the set
of test scenarios may be statistically consistent, each specific scenario’s param-
eters are not. Therefore, the experiments and results in this subsection are for
readers’ reference only.

1010 6. Conclusions

This paper introduces a general multi-agent reinforcement learning method
that integrates an adaptive manoeuvre strategy for real-time multi-aircraft con-
flict resolution. We evaluated the performance of our method using several
experiments and made the following key findings:

- 1015 1. The proposed method is applicable to real-time multi-aircraft conflict res-
olution problems, regardless of the number of aircraft or the airspace con-
struction in a scenario. Heading manoeuvres are more critical than speed
manoeuvres, and the combination of the two yields the best results.
2. As the uncertainty level increases, the difficulty of solving conflicts also
1020 increases. The agent’s ability to handle uncertainty is positively related
to the warning area size. However, an excessively large warning area may
lead to unnecessary extra flight distance.
3. In the limited experimental environment with a limited amount of publicly
available data, the proposed method produces more competitive results
1025 than the state-of-the-art RL-based CR methods.

Our work is a starting point for bridging the gap between theory and practice in multi-agent reinforcement learning methods used for multi-aircraft conflict resolution. However, we acknowledge that real-world scenarios are more complex than those in our experiments. In the future, we plan to systematically investigate neural network parameters adapted to the proposed method, introduce more challenging scenarios, enhance the model with realistic features, and extend it to three-dimensional scenarios, scenarios with multiple controlled sectors, and larger and multidimensional action spaces. Additionally, we suggest that shared and trusted autonomy, such as explainable artificial intelligence and hybrid-augmented intelligence, could facilitate the evolution of the autonomous ATM ecosystem, considering the critical nature of the air traffic system.

Acknowledgements

This study is funded by the National Natural Science Foundation of China (No.61903187), the National Key R&D Program of China (No.2021YFB1600500), the Natural Science Foundation of Jiangsu Province (BK20190414), and the China Scholarship Council (No.202006830095).

References

- [1] R. Marti, G. Reinelt, The linear ordering problem: exact and heuristic methods in combinatorial optimization, Applied Mathematical Sciences 175.
- [2] H. Liu, F. Liu, X. Zhang, X. Guan, J. Chen, P. Savinaud, Aircraft conflict resolution method based on hybrid ant colony optimization and artificial potential field, Science China Information Sciences 61 (12) (2018) 1–3.
- [3] E. Sunil, J. Ellerbroek, J. M. Hoekstra, Camda: Capacity assessment method for decentralized air traffic control, in: submitted to International Conference for Research of Air Transport (ICRAT 2018), Barcelona, 2018, pp. 26–29.

- 1055 [4] J. Bertram, P. Wei, J. Zambreno, A fast markov decision process-based algorithm for collision avoidance in urban air mobility, *IEEE Transactions on Intelligent Transportation Systems* 23 (9) (2022) 15420–15433.
- [5] X. Yang, P. Wei, Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility, *Journal of Guidance, Control, and Dynamics* 43 (8) (2020) 1473–1486.
- [6] R. Bellman, Dynamic programming, *Science* 153 (3731) (1966) 34–37.
- 1060 [7] 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, *Transportation Research Part C: Emerging Technologies* 135 (2022) 103463.
- 1065 [8] Z. Wang, H. Li, J. Wang, F. Shen, Deep reinforcement learning based conflict detection and resolution in air traffic control, *IET Intelligent Transport Systems* 13 (6) (2019) 1041–1047.
- 1070 [9] J. Lai, K. Cai, Z. Liu, Y. Yang, A multi-agent reinforcement learning approach for conflict resolution in dense traffic scenarios, in: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, IEEE, 2021, pp. 1–9.
- [10] R. Dalmau, E. Allard, Air traffic control using message passing neural networks and multi-agent reinforcement learning, in: *Proceedings of the 10th SESAR Innovation Days, Virtual Event, 2020*, pp. 7–10.
- 1075 [11] S. Dong, X. Weiping, K. ZHANG, Study on the resolution of multi-aircraft flight conflicts based on an idqn, *Chinese Journal of Aeronautics*.
- [12] M. W. Brittain, X. Yang, P. Wei, Autonomous separation assurance with deep multi-agent reinforcement learning, *Journal of Aerospace Information Systems* 18 (12) (2021) 890–905.

- [13] P. Zhao, Y. Liu, Physics informed deep reinforcement learning for aircraft
1080 conflict resolution, *IEEE Transactions on Intelligent Transportation Sys-*
tems.
- [14] M. Ribeiro, J. Ellerbroek, J. Hoekstra, Review of conflict resolution meth-
ods for manned and unmanned aviation, *Aerospace* 7 (6) (2020) 79.
- [15] C. Spatharis, A. Bastas, T. Kravaris, K. Blekas, G. A. Vouros, J. M.
1085 Cordero, Hierarchical multiagent reinforcement learning schemes for air
traffic management, *Neural Computing and Applications* (2021) 1–13.
- [16] Y. Xie, A. G. Gardi, R. Sabatini, Hybrid ai-based demand-capacity bal-
ancing for uas traffic management and urban air mobility, in: *AIAA AVI-*
ATION 2021 FORUM, 2021, p. 2325.
- [17] T. Duong, K. K. Todi, U. Chaudhary, H.-L. Truong, Decentralizing air
1090 traffic flow management with blockchain-based reinforcement learning, in:
2019 IEEE 17th International Conference on Industrial Informatics (IN-
DIN), Vol. 1, IEEE, 2019, pp. 1795–1800.
- [18] N. P. Tran, D.-T. Pham, S. K. Goh, S. Alam, V. Duong, An intelligent
1095 interactive conflict solver incorporating air traffic controllers’ preferences
using reinforcement learning, in: *2019 Integrated Communications, Navi-*
gation and Surveillance Conference (ICNS), IEEE, 2019, pp. 1–8.
- [19] S. Ghosh, S. Laguna, S. H. Lim, L. Wynter, H. Poonawala, A deep ensemble
multi-agent reinforcement learning approach for air traffic control, *arXiv*
1100 preprint arXiv:2004.01387.
- [20] K. Zhang, Y. Yang, C. Xu, D. Liu, H. Song, Learning-to-dispatch: Rein-
forcement learning based flight planning under emergency, in: *2021 IEEE*
International Intelligent Transportation Systems Conference (ITSC), IEEE,
2021, pp. 1821–1826.

- 1105 [21] S. Li, M. Egorov, M. Kochenderfer, Optimizing collision avoidance in dense airspace using deep reinforcement learning, arXiv preprint arXiv:1912.10146.
- [22] E. Yilmaz, O. Sanni, M. Kotwicz Herniczek, B. German, Deep reinforcement learning approach to air traffic optimization using the muzero algorithm, in: AIAA AVIATION 2021 FORUM, 2021, p. 2377.
- 1110 [23] H. Wen, H. Li, Z. Wang, X. Hou, K. He, Application of ddpg-based collision avoidance algorithm in air traffic control, in: 2019 12th International Symposium on Computational Intelligence and Design (ISCID), Vol. 1, IEEE, 2019, pp. 130–133.
- [24] S. Balasooriyan. [Multi-aircraft conflict resolution using velocity obstacles](#) [online] (2017).
- [25] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971.
- 1120 [26] J. Peters, S. Schaal, Natural actor-critic, Neurocomputing 71 (7-9) (2008) 1180–1190.
- [27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347.
- [28] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602.
- 1125 [29] W. Guo, M. Brittain, P. Wei, Safety enhancement for deep reinforcement learning in autonomous separation assurance, arXiv preprint arXiv:2105.02331.
- [30] M. Brittain, P. Wei, Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning, IEEE Transactions on Automation Science and Engineering 19 (4) (2022) 2837–2848.
- 1130

- [31] M. Brittain, P. Wei, Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning, in: 2018 International Conference on Research in Air Transportation (ICRAT), 2018, pp. 1–8.
- [32] D.-T. Pham, N. P. Tran, S. K. Goh, S. Alam, V. Duong, Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty, in: 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF), IEEE, 2019, pp. 1–6.
- [33] M. Ribeiro, J. Ellerbroek, J. Hoekstra, Improvement of conflict detection and resolution at high densities through reinforcement learning, in: ICRAT 2020: International conference on research in air transportation, 2020, pp. 1–4.
- [34] T. Fan, P. Long, W. Liu, J. Pan, Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios, The International Journal of Robotics Research 39 (7) (2020) 856–892.
- [35] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver, Rainbow: Combining improvements in deep reinforcement learning, in: Thirty-second AAAI conference on artificial intelligence, 2018, pp. 1–8.
- [36] G. Papoudakis, F. Christianos, A. Rahman, S. V. Albrecht, Dealing with non-stationarity in multi-agent deep reinforcement learning, arXiv preprint arXiv:1906.04737.
- [37] Z. Rezo, S. Steiner, South east common sky initiative free route airspace-implementation aftermath, Transportation Research Procedia 45 (2020) 676–683.
- [38] ICAO, Procedures for navigation services - air traffic management (16th edition of doc 4444), Tech. rep., ICAO (2016).

- [39] R. Wang, R. Alligier, C. Allignol, N. Barnier, N. Durand, A. Gondran, Co-
 1160 operation of combinatorial solvers for en-route conflict resolution, *Transportation Research Part C: Emerging Technologies* 114 (2020) 36–58.
- [40] M. W. Brittain, P. Wei, One to any: Distributed conflict resolution with
 deep multi-agent reinforcement learning and long short-term memory, in:
 AIAA Scitech 2021 Forum, 2021, p. 1952.
- [41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G.
 1165 Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski,
 et al., Human-level control through deep reinforcement learning, *nature*
 518 (7540) (2015) 529–533.
- [42] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with dou-
 1170 ble q-learning, in: *Proceedings of the AAAI conference on artificial intelligence*,
 Vol. 30, 2016, pp. 1–7.
- [43] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling
 network architectures for deep reinforcement learning, in: *International
 conference on machine learning*, PMLR, 2016, pp. 1995–2003.
- [44] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay,
 1175 arXiv preprint arXiv:1511.05952.
- [45] J. F. Hernandez-Garcia, R. S. Sutton, Understanding multi-step deep re-
 inforcement learning: a systematic study of the dqn target, arXiv preprint
 arXiv:1901.07510.
- [46] W. Dabney, G. Ostrovski, A. Barreto, Temporally-extended *epsilon*-greedy
 1180 exploration, arXiv preprint arXiv:2006.01782.
- [47] J. K. Archibald, J. C. Hill, N. A. Jepsen, W. C. Stirling, R. L. Frost, A
 satisficing approach to aircraft conflict resolution, *IEEE Transactions on
 Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (4)
 1185 (2008) 510–521.

- [48] M. Schäfer, X. Olive, M. Strohmeier, M. Smith, I. Martinovic, V. Lenders, Opensky report 2019: Analysing tcas in the real world using big data, in: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), IEEE, 2019, pp. 1–9.
- 1190 [49] ICAO, Performance-based navigation (pbn) manual (4th edition of doc 9613), Tech. rep., ICAO (2013).
- [50] EUROCONTROL, Performance review report an assessment of air traffic management in europe during the calendar year 2018, Tech. rep., EUROCONTROL, Brussels, Belgium (2018).
- 1195 [51] D. McCallie, J. Butts, R. Mills, Security analysis of the ads-b implementation in the next generation air transportation system, International Journal of Critical Infrastructure Protection 4 (2) (2011) 78–87.
- [52] F. Nadi. [Ads-b introduction / tutorial](#) [online] (2005).
- [53] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
- 1200

Appendix A. Explanation for intruder distance

The intrusion distance d_{Intru}^t is utilised in $\mathbf{o}_{\text{il}}^t$ instead of the separation distance $\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Ownship}}^t\|$ (i.e., the distance between the intruder and the ownship). This is because the value of d_{Intru}^t is more consistent with the value in non-intruder cases than the value of $\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Ownship}}^t\|$ (refer to Figure A.19). A smaller value of the first element of sub-observation $\mathbf{o}_{\text{il}}^t$ indicates a larger distance between the intruder and the ownship. A value of 0 implies no intruder within the detection range, thereby indicating a safe scenario.

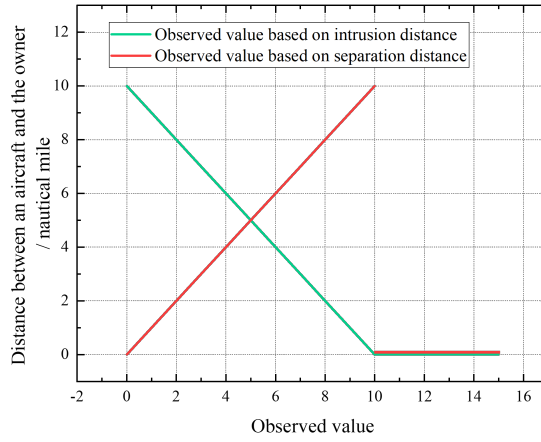


Figure A.19: An example of observed values based on different distance metrics, where the detection range $R^{\text{Detection}}$ is set to 10 nautical miles. Suppose the observed value is based on separation distance $\|\mathbf{p}_{\text{Intru}}^t - \mathbf{p}_{\text{Ownship}}^t\|$ (in red). When the distance between the aircraft and the ownship is very close to zero, it is challenging for the agent to distinguish whether the aircraft is outside the detection sector or very close to itself because the first element of sub-observation $\mathbf{o}_{\text{il}}^t$ is 0 in both cases (refer to Equation 19). If the observed value is based on intrusion distance d_{Intru}^t (in green), that problem will not occur.

Appendix B. Effect of adaptive yaw angle on aircraft manoeuvrability

1210

Figure B.20 shows an example of the function of yaw angle with distance to the exit point. Although the yaw angle decreases as the aircraft gets closer to the exit point, the rate of reduction in yaw angle will not begin to spike until the distance to the exit point is less than 20 nautical miles. Therefore, the adaptive yaw angle does not significantly reduce the manoeuvrability of the aircraft when the aircraft is far from the exit point. Furthermore, when the aircraft is close to the exit point, limiting the yaw angle of the aircraft not only ensures that it reaches the exit point, but also promotes a smooth trajectory.

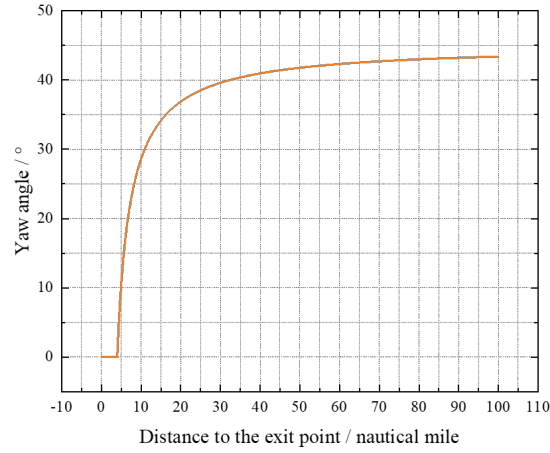


Figure B.20: An example of the function of yaw angle with distance to the exit point. Assuming that the aircraft can fly four nautical miles at each time step ($\|\mathbf{p}^{t+1} - \mathbf{p}^t\| = 4$ nautical mile) and yaw at a maximum angle of 45 degrees ($\alpha_{\max} = 45^\circ$), the yaw angle is represented by α^t , and the distance to the exit point is represented by $\|\mathbf{p}^{\text{Exit}} - \mathbf{p}^t\|$.

Appendix C. Training scenario parameters

1220 The circular airspace, with a radius of 50 nautical miles, was divided evenly
into n sections. The entry point of each aircraft was randomly generated on
the arc of each section, with its entry and exit points being symmetrical about
the circle’s centre. A study based on Dutch airspace predicted that by 2025,
the maximum density of upper airspace traffic (above 24500 feet) would be
1225 up to 45 aircraft per 10000 square nautical miles [14]. If we convert this to
traffic density at each flight level, based on the predicted distribution of traffic
by flight levels for 2025 in the 2018 Performance Review Report (published
by EUROCONTROL) [50], we can expect that the busiest flight level (FL370)
will have 4.8 aircraft in the circular airspace with a radius of 50 nautical miles
1230 in 2025. Therefore, we randomly generated six aircraft in our scenarios for
the longer-term future. Considering that most commercial aircraft fly at speeds
between 400 and 500 knots [13], we randomly set the economical speed of aircraft
as $V^E \in [430, 470]$ knots, while $V^L = (V^E - 30)$ knots and $V^H = (V^E + 30)$
knots. The time of each time step Δt was set as 30 seconds. Based on the
1235 performance of typical commercial aircraft [14], the maximum yaw angle $\alpha^{\max}$
was set to 45° . We set the range of uncertainty as $\varepsilon_h \in [-3^\circ, 3^\circ]$ for heading
and $\varepsilon_v \in [-6\%, 6\%]$ for speed [39]. Based on the set of aircraft speeds and the
time of each time step, it typically takes about 24-30 steps for an aircraft to fly
100 nautical miles in a straight line. However, considering the manoeuvres for
1240 conflict resolution, it may take up to 36 steps. In order to make the scenarios
more general, the entry time of each aircraft was randomly set between the first
to the tenth time step. The radii of the protected and warning areas were set
to 5 [38] and 8 nautical miles, respectively. The detection area was divided into
36 detection sectors with a radius of 50 nautical miles. The above parameters
1245 are summarised in Table C.5.

It should be noted that with regards to aircraft communication performance
(e.g., ADS-B), information is updated every second [51] or even faster (e.g.,
every 0.5 seconds in high update rates [52]). Since $\Delta t = 30$ seconds in the

Table C.5: Parameter settings for training scenarios.

Parameters	Values (Ranges)
Radius of the airspace R^{Airspace}	50 nautical miles
Number of aircraft n	6
Economical speed V^{E}	[430, 470] knots
Low speed V^{L}	$(V^{\text{E}} - 30)$ knots
High speed V^{H}	$(V^{\text{E}} + 30)$ knots
Maximum yaw angle α^{max}	45°
Time of each time step Δt	30 seconds
Uncertainty of heading ε_h	$[-3^\circ, 3^\circ]$
Uncertainty of speed ε_s	$[-6\%, 6\%]$
Radius of the protected area $R^{\text{Protected}}$	5 nautical miles
Radius of the warning area R^{Warning}	8 nautical miles
Radius of the detection area $R^{\text{Detection}}$	50 nautical miles
Number of detection sectors $N^{\text{Detection}}$	36

proposed method, with typical information synchronisation failure rates, infor-
1250 mation updated only once per time step (e.g., the last action of the intruder)
is rarely delayed. The delay in real-time information updates, such as the in-
truder’s current position, can almost be ignored ($0.5/30 \approx 1.67\%$) or treated as
a source of flight uncertainty. Therefore, the proposed method has the potential
to operate with the same level of performance as currently available real-world
1255 communication equipment.

Appendix D. Reinforcement learning hyper-parameters

The network parameters were updated using the Adam optimisation algorithm [53], which is a popular and commonly used optimiser in CR research based on RL [8] [12]. The hyper-parameters in Algorithm 1 are summarised in Table D.6. The hyper-parameters were chosen based on related references and practical experience. Specifically, the greedy rate ϵ was set to 0.9, which means the agent has a 10% chance of randomly selecting an action [11]. The buffer capacity n_{BC} was set to 2×10^5 . The batch size n_{BS} was set to 200. The discount rate γ was set to 0.9 [8] [23]. The learning step n_{MS} was set to 5. The learning rate α was set to 1×10^{-3} [7] [8] [11]. The target network update cycle n_{TN} was set to 1×10^5 .

Table D.6: Hyper-parameter settings for training algorithm.

Hyper-parameters	Values
Greedy rate ϵ in line 10	0.9
Buffer capacity n_{BC} in line 16	2×10^5
Batch size n_{BS} in line 21	200
Discount rate γ in line 22	0.9
Learning step n_{MS} in line 22	5
Learning rate α in line 23	1×10^{-3}
Target network update cycle n_{TN} in line 24	1×10^5



DSPACE

<https://dspace.org/>

General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution

Chen, Yutong; Hu, Minghua; Yang, Lei; Xu, Yan; Xie, Hua

2023-06-01

Elsevier

<https://doi.org/10.1016/j.trc.2023.104125>

Chen Y, Hu M, Yang L, et al., (2023) General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution. Transportation Research Part C: Emerging Technologies, Volume 151, June 2023, Article Number 104125