*The Aeronautical Journal* (2025), **129**, pp. 2240–2262
doi:10.1017/aer.2025.30

ROYAL AERONAUTICAL SOCIETY

**RESEARCH ARTICLE**

# Multi-view representation learning for performance-based tactical conflict resolution in urban air mobility

Cheng Huang, Ivan Petrunin and Antonios Tsourdos

School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire, UK
**Corresponding author:** Cheng Huang; Email: cheng-huang.huang@outlook.com

**Received:** 2 November 2023; **Revised:** 17 November 2024; **Accepted:** 11 March 2025

**Keywords:** tactical conflict management; graph neural networks; multi-view representation

## Abstract

Ensuring safety and security in urban air mobility is of utmost significance. As air traffic becomes more concentrated in urban regions, instances of flight conflicts are on the rise. The complex urban morphology, micro-environmental factors and various flight risks significantly impact flight safety. This paper introduces a comprehensive framework to address these challenges. The framework incorporates random 3D city layouts, encompassing city buildings and terrains to establish a corridor graph structure. Leveraging this graph, a multi-view representation learning approach is proposed, which employs graph neural networks, recurrent neural networks (RNN) and contrastive learning to effectively manage tactical conflicts. Through rigorous testing across diverse scenarios, including the incorporation of uncertainties such as wind turbulence, the model's performance is extensively evaluated. The conclusive results underscore the robustness and efficacy of the proposed approach in ensuring safety and resolving conflicts within the dynamic urban air mobility landscape.

## Nomenclature

| | |
|---|---|
| x, y, z | coordinates in the three-dimensional space |
| $M_L$ | obstacle map at flight level $L$ |
| $P_L$ | set of selected collision-free points at level $L$ |
| $C_L$ | flight corridor at level $L$ |
| $C_{\text{UAM}}$ | overall flight corridor |
| $W$ | waypoints |
| $V$ | vertiports |
| $G$ | graph |
| $I$ | intersections of flight corridors |
| E | set of edges connecting nodes |
| $f_N$ | flight number |
| $\mathbf{H}$ | node feature |
| $\mathcal{L}_{CE}$ | cross-entropy |
| $\mathcal{L}_{CR}$ | contrastive loss |

## 1.0 Introduction

Modern transportation systems in many large cities face significant challenges, including traffic congestion, long travel times and environmental concerns due to high carbon emissions. The ground transportation network is struggling to keep up with the growing urban population, leading to increased traffic congestion and reduced efficiency. However, the emergence of vertical take-off and landing (eVTOL) technology offers an opportunity to solve these ground-related problems. Urban air

© The Author(s), 2025. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

mobility (UAM) leverages drones to operate, offering significant advantages over traditional ground transportation [1]. By using eVTOL vehicles, UAMs can bypass congested roads, dramatically reducing travel times. This not only saves passengers valuable time but also improves productivity and spurs economic growth in cities. In addition, UAM contributes to addressing environmental challenges by using electric propulsion, which reduces carbon emissions. This eco-friendly feature aligns with the global effort toward sustainability and plays a key role in creating greener and more livable urban environments. In this context, UAM offers an attractive solution with many advantages over traditional road transport.

The growth of UAM has gained significant momentum due to the increasing demand for more efficient transportation solutions and advancements in aerospace technology. Both industry leaders and startups are making substantial investments in the research and development of (eVTOL vehicles [2, 3], flight control systems [4], and airspace management solutions [5] to make UAM a reality. However, amidst these advancements, it is crucial to not overlook the importance of real-time security and safety factors in the implementation of UAM.

As UAM integrates into urban airspace, it presents unique challenges such as navigating complex urban environments and sharing airspace with other aircraft and drones. To address these challenges, robust safety management systems need to be implemented. This includes the use of sensing and avoidance technology and reliable flight control systems to prevent collisions and ensure smooth operation [1]. Furthermore, integrating UAM into urban environments necessitates the implementation of strict security measures.

Conflict management in UAM is a critical aspect of ensuring safe and efficient operations in complex urban airspace. Effective conflict management in UAM can be achieved through a three-stage approach [6]. The first stage involves strategic planning, which includes comprehensive pre-flight planning and airspace management. This entails establishing conflict-free flight paths, defining restricted zones and coordinating with relevant authorities. The second stage, known as the tactical stage, focuses on real-time monitoring and communication. Advanced tracking systems and communication technologies are utilised to ensure continuous surveillance and prompt responses to potential conflicts. Lastly, onboard collision avoidance and emergency response play a crucial role in detecting and mitigating conflicts. By implementing this three-level strategy, the integration of UAM into urban transportation systems can be achieved in a safe and efficient manner.

Tactical conflict resolution strategies used for conventional aircraft in high-altitude airspace may not be directly applicable to low-altitude urban airspace due to the unique challenges posed by buildings, terrains and dynamic obstacles in urban environments. In low-altitude urban airspace, the operating conditions are vastly different, and traditional conflict resolution methods like simple altitude or heading changes [7] might not be practical or feasible. Therefore, many studies proposed specific airspace structures and flight corridor designs, which aim to establish a regulated and efficient framework for UAM operations in urban environments, considering factors such as buildings, terrains and dynamic obstacles. Typical airspace structures include (a) multi-layered urban airspace with different structures (sky-lanes, sky-corridors) proposed by NASA to regulate UAM flights, while UOE provides flexibility depending on metropolitan needs [8]; (b) UAM corridors proposed by FAA, which enables point-to-point operations with separation maintained by UAM operators, paving the way for more efficient and complex networks [1]; (c) route networks above existing roads, river and infrastructures [9, 10]. In addition to airspace structure, the concept of flight corridors is proposed to regulate the paths that UAM vehicles can take within urban environments [11]. Simply, flight corridors define predetermined routes or virtual pathways that UAM vehicles should follow while navigating through densely populated areas. Based on this, several conflict detection and resolution algorithms have been proposed to enhance the safety and efficiency of UAM operations within the designated airspace structures [12].

As the common feature in various airspace structures is the representation of flight paths using graphs [12], where nodes represent waypoints, and edges represent flight routes, we propose a brand-new multi-view presentation learning framework that leverages multiple graph-based representations for enhanced tactical learning in UAM operations. By considering geospatial, temporal and traffic flow characteristics

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

as integrated graph views, the framework enables a comprehensive understanding of the dynamic urban airspace, thereby enhancing the efficiency of tactical conflict resolution.

The contributions of this work are summarised as follows:

1. A general UAM operation environment workflow is designed to generate the flight corridor graph structure. By integrating spatial, temporal and traffic flow characteristics into the graph, our proposed pipeline enables a holistic understanding of the dynamic urban airspace, enhancing the efficiency of tactical conflict resolution.
2. A novel multi-view representation learning approach is proposed for tactical conflict resolution, which utilises graph neural networks (GNNs) and recurrent neural networks (RNNs) to acquire knowledge from multi-view graph representations, facilitating effective information aggregation and node-level feature learning.
3. The inclusion of uncertain wind turbulence in urban airspace is being considered to enhance the precision and effectiveness of tactical conflict resolution for UAM operations under practical conditions.

The subsequent sections of this paper are structured as follows: Section 2.0 presents an overview of related research efforts. In section 3.0, a detailed workflow is demonstrated, encompassing the establishment of the operational environment, formulation of graph-based problems and introduction of the multi-view representation learning approach for conflict resolution. The outcomes of some case studies are analysed in section 4.0. Finally, section 5.0 concludes the paper.

## 2.0 Related works

### *2.1 Tactical conflict resolution in air traffic management*

In air traffic management, tactical conflict resolution is primarily carried out by skilled air traffic controllers (ATCOs). These professionals work in air traffic control centeres and are responsible for ensuring the safe and efficient movement of aircraft in their designated airspace. When conflicts between aircraft occur, air traffic managers rely on their expertise and real-time data from radar systems to make quick and informed decisions. They issue clear instructions to pilots, considering factors like aircraft speed, altitude and trajectory, to resolve conflicts and maintain safe distances between aircraft.

With the development of artificial intelligence, some intelligent approaches attempt to take over some responsibilities for conflict resolution to reduce the workload of ATCOs. Especially, reinforcement learning is widely applied. The heading change is normally considered as a valid action for the resolution with the involvement of uncertainties, e.g. the deviation of the actual manoeuver [7]. In addition, speed adjustment, such as accelerating or decelerating, is another action for the multi-agent system incorporating the long short-term memory (LSTM) network to learn effective feature representations [13]. The probabilistic risk map of the intruder locations can also be integrated into the multi-agent environment to perform pairwise aircraft and multi-aircraft conflicts [14].

### *2.2 Tactical conflict resolution for urban air mobility*

Traditional air traffic primarily involves commercial airlines operating at higher altitudes, conducting flights between established airports and adhering to predefined flight routes within controlled airspace. In contrast, UAM marks a transformative shift, concentrating on short-distance flights within urban and densely populated regions. UAM vehicles operate at lower altitudes, manoeuvering through intricate and ever-changing urban landscapes that encompass buildings, infrastructure and obstacles. This dynamic environment demands adaptation to numerous variables, including wind turbulence and architectural layouts, which significantly influence flight paths [5].

Nevertheless, certain conflict resolution methods in UAM often maintain analogous principles to those in traditional air traffic management models. Various intersection types, such as merging, divergent

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

and splitting, are typically classified for conflict detection [12]. Moreover, flight conflicts are addressed using techniques like departure delays or flight-level assignments, which are tackled through approaches like Mixed Integer Linear Programming (MILP) [15] and the Rolling Window algorithm [16]. Other methodologies, like prediction-based conflict-free adaptive navigation (PCAN) [17], have been introduced to forecast the future positions of unmanned aerial vehicles (UAVs) and adapt their velocity vectors to address conflicts. Furthermore, the research delves into approaches that incorporate realistic elements, such as weather maps, risk maps and geographical data, to enhance strategic and tactical planning processes [18]. Various simplistic planners are devised to manage conflicts across diverse environments.

These investigations primarily explore non-learned strategies for tactical conflict resolution, often involving adjustments to flight speed or altitude. Intelligent models have also gained attention in numerous studies. For instance, multi-agent reinforcement learning is leveraged to resolve tactical conflicts by altering acceleration vectors [19], direct headings [20] or speed values [21]. Notably, the integration of attention mechanisms within the multi-agent reinforcement learning framework enhances feature learning efficiency [19].

It is evident that learning-based strategies for resolving tactical conflicts in small UAVs or UAM predominantly revolve around reinforcement learning techniques. Interestingly, other intelligent models that find extensive use in smart transportation systems have yet to be explored for adaptation to the UAM domain. Take GNNs for instance, a potent tool in transportation contexts capable of comprehending intricate spatial and temporal connections inherent in transportation networks. While traditional planning algorithms excel at pre-flight trajectory optimisation, they cannot effectively address the dynamic challenges of real-time conflict resolution among multiple vehicles. In the realm of traffic flow, GNNs have proven adept at predicting congestion patterns, optimising routes and approximating travel durations by comprehending the underlying graph layout of road networks [22–24]. Given that the UAM operational environment shares similarities with ground transport structures, there's potential for some intelligent models applicable to 2D ground settings to be adapted for the complexities of the 3D low-altitude airspace.

While traditional planning algorithms excel at pre-flight trajectory optimisation, they cannot effectively address the dynamic challenges of real-time conflict resolution among multiple vehicles. Our learning-based approach, implemented through GNN, enables the system to learn from experience and provide immediate, coordinated resolution strategies for multiple interacting vehicles during actual operations. This capability for real-time tactical decision-making and multi-vehicle coordination is essential for practical urban air mobility operations, where conflicts must be resolved dynamically while considering the complex interactions between multiple aircraft.

## 3.0 Methodology

In this section, a comprehensive framework is designed for creating a dynamic scenario for testing and evaluating intelligent approaches in conflict resolution for UAM. As shown in Fig. 1, the flight environment is designed at first, which involves generating terrains, building structures, random city layouts, weather conditions and traffic simulations to create dynamic scenarios for tactical conflict resolution. Next, the tactical conflict problem is formulated, identifying potential conflicts and challenges faced by UAM vehicles in the given environment. Finally, the multi-view representation learning approach is proposed to address tactical conflict resolution efficiently and enhance the safety and efficiency of UAM operations within the designed flight environment.

### *3.1 Operation environment*

#### *3.1.1 Three-dimensional cityscape generation*

To design the flight environment, the approach follows the methodology presented in [25] to generate 3D terrains and buildings. This process allows for the creation of random 3D cities, providing a diverse and

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

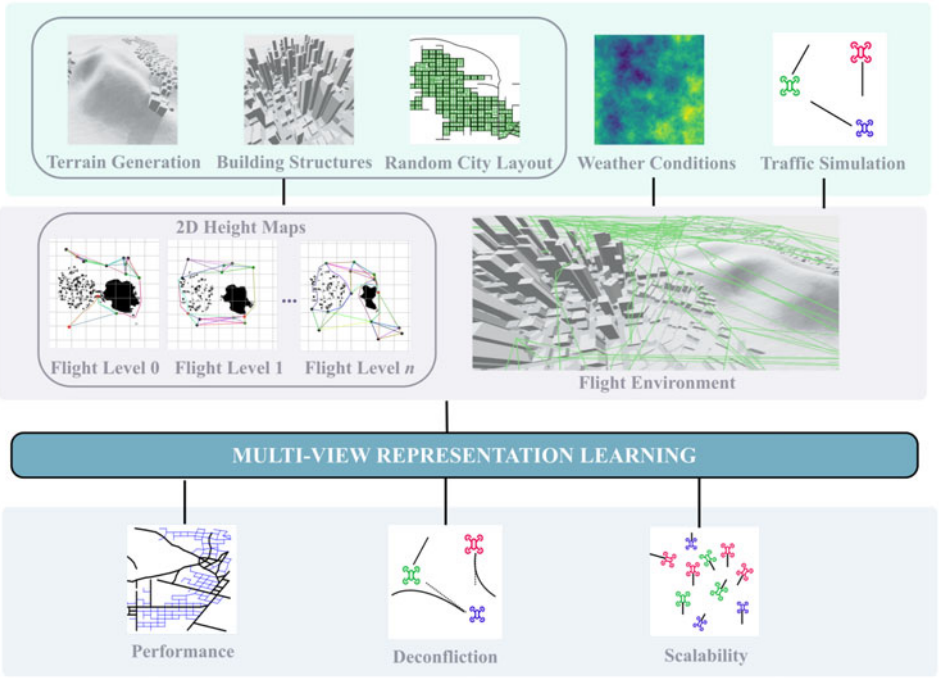



***Figure 1.*** *The flowchart of the tactical conflict resolution.*

realistic urban landscape. The use of procedural modeling ensures the simulation of complex city layouts, including varying building heights, structures and terrains, offering a comprehensive and dynamic scenario for low-altitude airspace management.

The figures depicted in Fig. 2 illustrate the procedural process utilised for the generation of a three-dimensional cityscape. It presents a step-by-step pipeline employed to construct a realistic urban environment. The initial stage in Fig. 2a involves the generation of a roadmap using an L-system rule, which entails the creation of a network of roads and streets that serve as the foundation for the layout of the city. Subsequently, the polygon generation phase involves the specification of the shape and size of individual city blocks as shown in Fig. 2b. The subsequent step is the generation of buildings as shown in Fig. 2c, where structures are added to the polygons, reflecting the architectural and design characteristics of the buildings within the city. Finally, the complete three-dimensional model is created as in Fig. 2d, which represents a visually appealing and realistic cityscape that can be utilised for low-altitude airspace organisation.

### *3.1.2 Flight routes generation*

After the generation of the 3D terrain and buildings model, the next critical step is to establish multi-layer flight routes based on the geographic information of the city. These flight routes are designed to enable safe and efficient operations of UAM vehicles within the complex urban environment. Each flight level corresponds to a specific altitude range, and multi-layer routes facilitate seamless flight at different heights.

To ensure safety, two-dimensional obstacle and risk maps are derived from the 3D terrain and building models. These maps provide crucial information on potential obstacles and risk areas that UAM vehicles need to avoid. By incorporating the terrain and building data into the maps, potential conflicts and hazardous zones can be identified, allowing for precise route planning.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

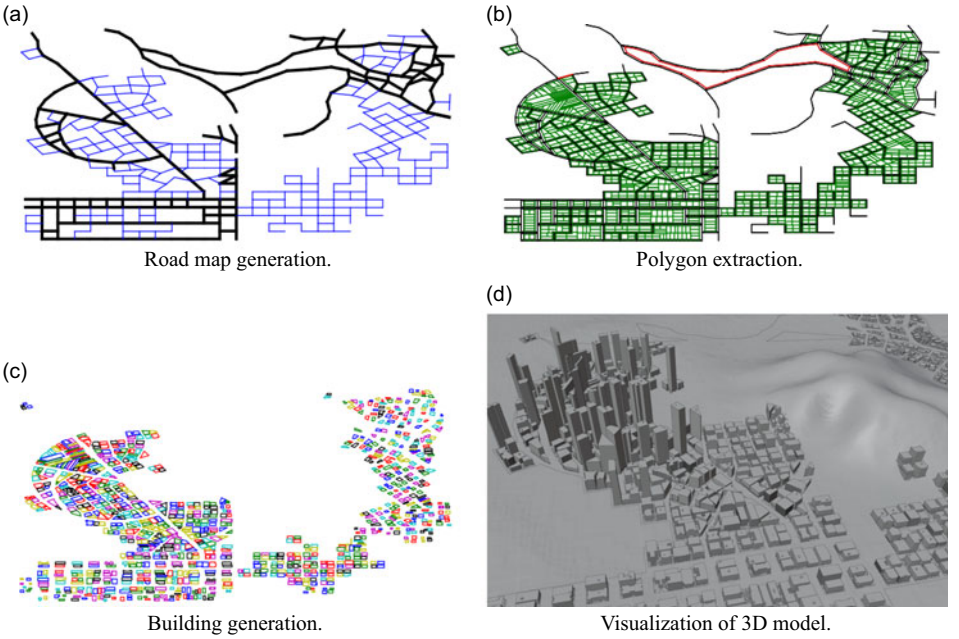



***Figure 2.*** *Random city layout and model generation.*

Let $R\left(x, y, z\right)$ represent the 3D model, where $x$, $y$ and $z$ are the coordinates in the 3D space $\mathbb{R}^3$ of the model. The obstacle map $M_L\left(x, y\right)$ at flight level $L$ can be defined as:

$$M_L\left(x, y\right) = \begin{cases} 1 & \text{if } \exists z \geq L \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $R\left(x, y, z\right)$ is a binary function that indicates the presence ($R\left(, y, z\right) = 1$) or absence ($R\left(x, y, z\right) = 0$) of an obstacle at a given point $(x, y, z)$ in the 3D model. In other words, the $M_L\left(x, y\right)$ map saves the points in the 2D plane above altitude $L$ from the 3D model $R\left(x, y, z\right)$. If there exists a point $(x, y, z)$ in the 3D space with altitude $z$ greater than or equal to $L$, the corresponding value in the altitude map $M_L\left(x, y\right)$ is set to 1, indicating the presence of an obstacle. Otherwise, the value in $M_L\left(x, y\right)$ is set to 0, representing the availability of space at the specified flight level.

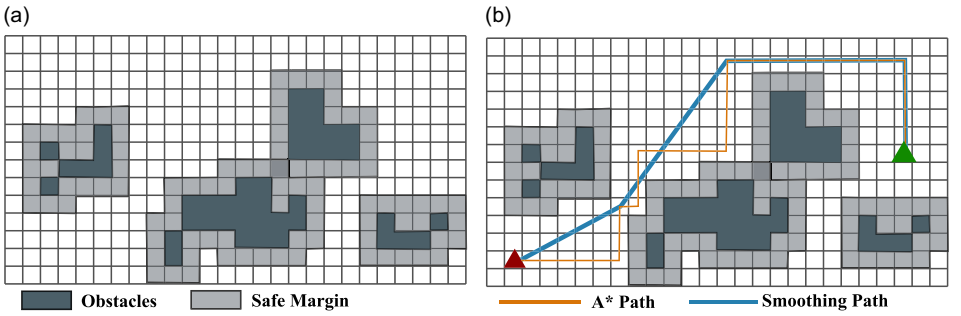
To account for environmental uncertainties and enhance safety during UAM operations, each 2D map $M_L$ at flight level $L$ is extended with a safe margin $a$. This safe margin is added near obstacles to create a buffer zone as depicted in Fig. 3a, ensuring that UAM vehicles maintain a safe distance from potential conflicts with static obstacles. The extended 2D map $M_L^{ext}$ is defined as:

$$M_L^{ext}\left(x', y'\right) = \begin{cases} 1 & \text{if } \exists z \geq L, \text{where } x' \in \left[x - a, x + a\right], y' \in \left[y - a, y + a\right] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $R\left(x', y', z\right)$ is the 3D model at point $(x', y', z)$; $x'$ and $y'$ are coordinates within the safe margin $a$ around $(x, y)$. The safe margin $a$ extends the 2D altitude map $M_L^{ext}\left(x, y\right)$ around each obstacle, ensuring that potential conflicts are detected early and allowing sufficient space for UAM vehicles to fly safely.

The next step is to create flight routes using path-finding algorithms, such as $A^*$, based on the extended 2D obstacle map $M_L^{ext}$. Especially, $A^*$ is an efficient and widely used algorithm for finding the shortest path between two points on a map. The first step is to select collision-free waypoints at specific locations within the map. Let $P_L = \{\left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_n, y_n\right)\}$ be the set of selected collision-free points at level $L$. Next, the shortest route between each pairwise point in $P_L$ is computed using $A^*$. The resulting routes are represented as $R_{ij}$ from $(x_i, y_i)$ to $\left(x_j, y_j\right)$, where $i, j \in [1, n]$. However, the generated routes $R_{ij}$ from $A^*$ are not always straight and may contain unnecessary zig-zag paths, which can be inefficient for

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press




Safe margin near obstacles. Path smoothing.

***Figure 3.*** *Flight corridor generation.*

UAM flights. Hence, the need for path smoothing arises. Path smoothing is essential to optimise the flight trajectory and improve the efficiency of UAM operations. As depicted in Fig. 3b, by applying the path smoothing, the generated routes are modified to create smoother and more direct trajectories, reducing energy consumption and travel time for UAM vehicles. Algorithm 3.1.2 describes the path smoothing approach in detail. It operates using two functions: *LineOfSight*, which checks for a clear line of sight between two points on the map using Bresenham's line algorithm [26], and *SmoothPath*, which iterates through consecutive pairs of points in the original path. The *SmoothPath* function retains points with a clear line of sight between them, effectively removing intermediate waypoints and generating a smoother path. By reducing unnecessary zig-zagging, the algorithm creates a more efficient trajectory $\hat{R}_{ij}$, leading to improved performance and safer navigation in complex urban environments.

The processed routes $\hat{R}_{ij}$ are regarded as prescribed flight corridors. For each flight level $L$, the prescribed corridors $C_L$ are given as:

$$C_L = \bigcup_{(x_i,y_i),(x_j,y_j)\in P_L, i\neq j} \hat{R}_{ij} \tag{3}$$

Finally, the flight corridors from all flight levels are combined to create the comprehensive flight path for the UAM system. The overall flight corridor $C_{\text{UAM}}$ is obtained as the union of all individual flight corridors at different levels:

$$C_{\text{UAM}} = \bigcup_{L} C_L \tag{4}$$

### *3.2 Problem formulation*

#### *3.2.1 Graph generation*

With the designed flight corridor structure $C_{\text{UAM}}$, waypoints $W$ and vertiports $V$ sets are integrated to build a graph $G = (N, E)$ , $N = V \cup W \cup I$, where $V$ represents vertiports, $W$ represents waypoints and $I$ represents the intersections of flight corridors. $E$ represents the set of edges connecting these nodes in the graph. An example of corridor structure is provided in Fig. 4a.

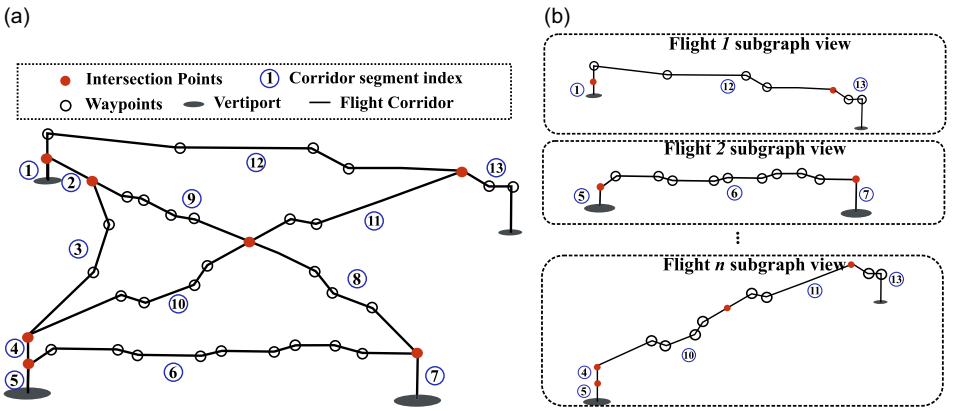
#### *3.2.2 Flight generation*

We set the flight number $f_N$ in the scenario and then for each flight $f_i$, the starting vertiport $v_{\text{start},f_i} \in V$ and the destination vertiport $v_{\text{dest},fi} \in V$ (start $\neq$ dest) are randomly selected. For each flight, we apply a pathfinding algorithm to find the shortest path $P_{f_i}$ as the scheduled path from the constructed graph $G$. And each flight is randomly assigned with flight velocity $v_{f_i}$, operation time $t_{f_i}$ and appropriate flight level.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

**Algorithm 1:** Path Smoothing Algorithm

**Function** `LineOfSight(`$M_L^{ext}$, $(x_i, y_i)$, $(x_j, y_j)$`):`
  `// Check if there is a clear line of sight between points` $(x_i, y_i)$ `and` $(x_j, y_j)$ `on the map` $M_L^{ext}$ `with Bresenham's line algorithm`
  **if** *there is an obstacle between* $(x_i, y_i)$ *and* $(x_j, y_j)$ *in map* $M_L^{ext}$ **then**
    **return** False;
  **end**
  **return** True;
**Function** `SmoothPath(`$M_L^{ext}$, $R_{ij}$`):`
  `// Smooth the given path` $R_{ij}$ `by removing unnecessary waypoints`
  Initialize an empty list $\hat{R}_{ij}$;
  Add the first point $(x_i, y_i)$ of the path $R_{ij}$ to $\hat{R}_{ij}$;
  **for** *each consecutive pair* $(x_k, y_k)$ *and* $(x_{k+1}, y_{k+1})$ *in* $R_{ij}$ **do**
    **if** *there is a clear line of sight between* $(x_k, y_k)$ *and* $(x_{k+1}, y_{k+1})$ *in map* $M_L^{ext}$ **then**
      Skip to the next pair of points;
    **end**
    Add the last point $(x_k, y_k)$ with a clear line of sight to $\hat{R}_{ij}$;
  **end**
  **return** $\hat{R}_{ij}$;
`// Main program`
Call `LineOfSight(`$M$, $(x_i, y_i)$, $(x_j, y_j)$`)` to check for a clear line of sight between $(x_i, y_i)$ and $(x_j, y_j)$ on the map $M_L^{ext}$;
Call `SmoothPath(`$M$, $R_{ij}$`)` to smooth the given path $R_{ij}$ and obtain the processed path $\hat{R}_{ij}$;




Example of corridor structure. Selective subgraph views of flights.

***Figure 4.*** *Flight corridor generation.*

The flight scheduling process involves constructing multi-view graphs, where each flight schedule is represented as a subgraph $G_{f_i} = \left(V_{f_i}, E_{f_i}\right)$ extracted from $G$ as depicted in Fig. 4b. $V_{f_i}$ is the set of nodes specific to flight $f_i$ and $E_{f_i}$ is the set of edges that connect those nodes and can be denoted as follows:

$$V_{f_i} = \left\{v_{f_i,j} | v_{f_i,j} \in V\right\} \tag{5}$$

$$E_{f_i} = \left\{e_{f_i,k} | e_{f_i,k} \in E\right\} \tag{6}$$

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

where $v_{fi,j}$ represents a node in flight $f_i$ with the subscript $j$ to indicate the node index in that subgraph, similarly, $e_{fi,k}$ represents an edge within flight $f_i$ with the subscript $k$ to indicate the edge index in that flight.

Meanwhile, an important distinction needs to be mentioned that the node features are not stored in the global graph $G$. Instead, the node features are specific to each subgraph, even if common nodes appear in different subgraphs.

Let $f\left(v_{fi,j}\right)$ represent the feature vector of node $v_{fi,j}$ in the subgraph for flight $f_i$. It consists of essential flight information, including the flight ID, arrival time, speed and waypoint coordinates $\left[x, y, z\right]$. We can express the feature vector $f\left(v_{fi,j}\right)$ of node $j$ as follows:

$$f\left(v_{fi,j}\right) = \left[f_i, t_j, s_j, \bar{x}_j, \bar{y}_j, \bar{z}_j\right]$$

where:

- $f_i$ represents the flight ID.
- $t_j$ represents the arrival time of flight $f_i$ at the node $j$.
- $s_j$ represents the speed of flight $f_i$ at the specified node.
- $\bar{x}_j, \bar{y}_j, \bar{z}_j$ represent the normalised $x, y, z$ coordinates of the waypoint of flight $f_i$.

The multiple subgraphs $G_{f_1}, G_{f_2}, \ldots, G_{f_n}$ construct the multi-view representation, where each subgraph $G_{fi} = \left(V_{fi}, E_{fi}\right)$ is associated with a specific flight $f_i$. This enables efficient representation and management of flight scheduling data within the multiple graph view.

#### *3.2.3 Conflict detection*

Real-time conflict detection is of utmost importance to ensure safe and efficient operations in the UAM environment. To achieve this, multi-view subgraphs are employed to comprehensively capture the interactions and dependencies among flights.

The entire operation time is discretised into $s$ discrete time slots $\{t_1, t_2, \ldots, t_s\}$ with a specific interval $\Delta t$ seconds. In each time slot $t_i$, the current positions of aircraft are represented as $\mathbf{P}\left(t_i\right) = \{\mathbf{p}_1\left(t_i\right), \mathbf{p}_2\left(t_i\right), \ldots, \mathbf{p}_n\left(t_i\right)\}$, where $n$ is the number of aircraft. The distance between any pairwise aircraft positions is calculated as $d\left(\mathbf{p}_j\left(t_i\right), \mathbf{p}_k\left(t_i\right)\right)$ for all $j, k$. If $d\left(\mathbf{p}_j\left(t_i\right), \mathbf{p}_k\left(t_i\right)\right)$ is less than a predefined safety margin $\delta d$, then a conflict is detected at time slot $t_i$.

Let conf $(t_i)$ be the set of conflict flight index tuples at time slot $t_i$, and denote as:

$$\mathrm{Conf}(t_i) = \left\{(j,k) \mid d\left(\mathbf{p}_j\left(t_i\right), \mathbf{p}_k\left(t_i\right)\right) < \delta d \text{ for } j,k \in [1,n] \text{ and } j \neq k\right\} \tag{7}$$

It returns the set of conflict flight index tuples at each time slot $t_i$, representing the detected conflicts between pairs of aircraft.

Moreover, it is possible to look ahead several time intervals for longer-time conflict detection at a specific time $t_i$, $i \in \{1, 2, \ldots, s\}$, allowing the system to anticipate and address potential conflicts $(\mathrm{conf}\left(t_{i+1}\right), \mathrm{conf}\left(t_{i+2}\right), \ldots)$ that may arise in the future time slots $t_{i+1}, t_{i+2}, \ldots$. This time-based conflict detection mechanism ensures continuous monitoring of aircraft positions and facilitates timely and efficient conflict resolution.

In addition, this conflict detection part is only used for ground truth generation. It is essential to highlight that during the model inference process, conflict detection will not be relied upon. Instead, the trained model will directly use the node features of flight schedules for conflict resolution and decision-making.

### *3.3 Muti-view representation learning*

After the conflicts are detected, we propose a multi-view representation learning scheme based on conflict and non-conflict views of flight trajectories and their interactions for conflict resolution. Each view

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

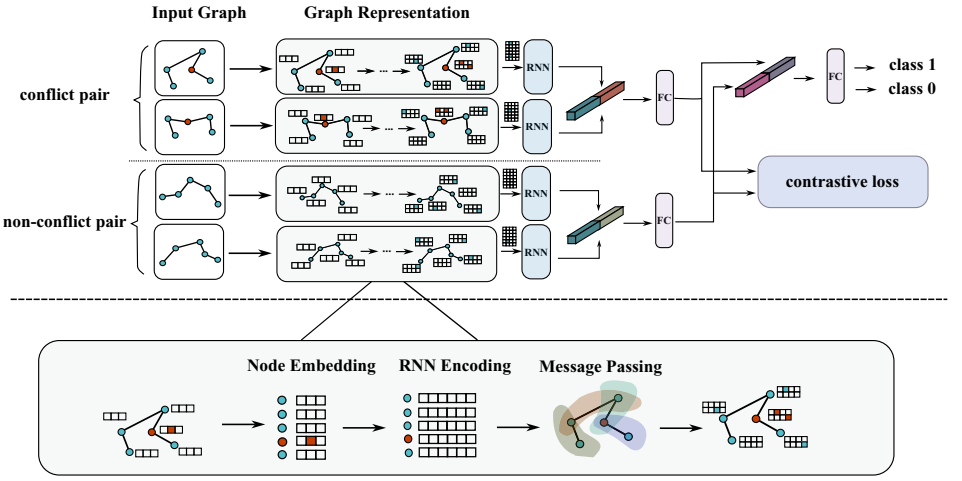



***Figure 5.*** *The architecture of multi-view graph representation learning.*

represents a unique perspective of flight schedules: conflict views capture scenarios where flights have conflicting nodes, while non-conflict views represent scenarios without conflicts. The architecture of the multi-view graph representation learning is demonstrated in Fig. 5. The input consists of two graphs, one for a conflict pair and another for a non-conflict pair. The flight schedules in both graphs have conflicting nodes in the conflicting pair, while in the non-conflict pair, the flights have no conflicts. The graph representation is designed to capture important features from the flight schedules, and the RNN is used to encode node features with GNN efficiently. The pairwise graph representations are concatenated and fed into a classifier for class prediction. Additionally, contrastive loss is employed, where the conflict and non-conflict pairs are used to encourage the GNN to learn discriminative features for distinguishing between conflicting and non-conflicting flights.

*3.3.1 Graph representation*

Graph representation is the critical module in the architecture. It takes the graph $\mathcal{G}$ as input, and the node feature matrix is embedded into high-dimensional features $\mathbf{X} \in \mathbb{R}^{N\times d}$, where $N$ is the number of nodes and $d$ is the original feature dimension. The RNN is then employed to encode the node features and can be expressed as follows:

$$\mathbf{h}_i = \mathrm{RNN}\left(\mathbf{h}_i, \mathbf{X}\left[i\right]\right)$$

where $\mathbf{h}_i$ is the hidden state of the RNN for node $i$, $\mathbf{X}\left[i\right]$ is the input feature of node $i$. The RNN updates the node features to $\mathbf{H} \in \mathbb{R}^{N\times D}$ based on the graph structure and node interactions.

The updated node features are then used in the message-passing process of the GNN. The information in node-wise message passing is propagated and exchanged between neighboring nodes in a graph in a sequential manner. At each step $t$, each node $i$ aggregates information from its neighboring nodes, forming a set of messages $\mathcal{M}_i^{(t)}$:

$$\mathcal{M}_i = \psi\left(\left\{\mathbf{H}_j \text{ for each neighbor } j \text{ of node } i\right\}\right) \tag{8}$$

Here, $\psi$ is a function that combines the neighboring node features in a way that summarises the information for each node $i$. The node then updates its own feature using the aggregated messages and its current feature:

$$\mathbf{H}'_i = \varphi\left(\mathbf{H}_i, \mathcal{M}_i\right) \tag{9}$$

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

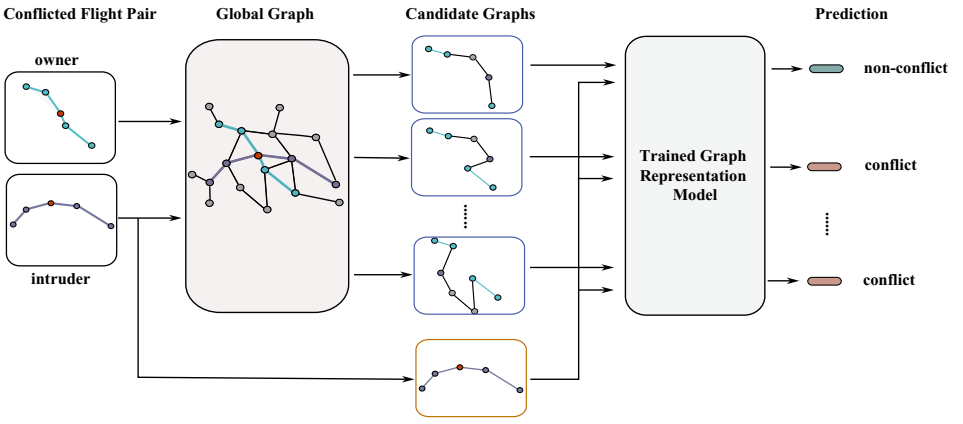



***Figure 6.*** *The workflow of conflict resolution with multi-view graph representation.*

where $\varphi$ is a function that updates the node features based on the aggregated messages and the current node features. The message-passing process is repeated for multiple time steps to allow the nodes to gather and exchange information with their neighbors iteratively to obtain final node features $\mathbf{H}^{(F)}$.

### *3.3.2 Loss function*

In the training process, we utilise a combination of two loss functions: cross-entropy $\mathcal{L}_{CE}$ and contrastive $\mathcal{L}_{CR}$ loss.

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{CR} \tag{10}$$

The cross-entropy loss is employed for predicting whether a given pair represents a conflicted or a non-conflicted pair of flights. This loss function allows us to effectively classify the two classes based on their input features.

$$\mathcal{L}_{\text{CE}}(p, q) = -\sum_{i} p_i \log(q_i) \tag{11}$$

where $p$ is ground-truth labels obtained from conflict detection, $q$ is the predicted probability distribution from the model, and log is the natural logarithm.

On the other hand, contrastive loss plays a crucial role in enhancing discrimination between the different classes. By enforcing a margin between the embeddings of conflict and non-conflict pairs, the contrastive loss helps the model to learn a more robust representation, effectively distinguishing between conflicting and non-conflicting schedules. The incorporation of both loss functions enables our model to not only accurately predict the conflict class but also to learn meaningful embeddings that facilitate better conflict evaluation.

$$\mathcal{L}_{\text{CR}}(x, y) = \frac{1}{2}\left(y \cdot d(x_1, x_2)^2 + (1 - y) \cdot (\max(0, \sigma - d(x_1, x_2)))^2\right) \tag{12}$$

where $x_1$ and $x_2$ are the embeddings of the two inputs, $d(x_1, x_2)$ is the Euclidean distance between $x_1$ and $x_2$, $\sigma$ is a margin hyperparameter, and $y$ is the binary label indicating whether the inputs are from the same class ($y = 1$) or different classes ($y = 0$).

### *3.3.3 Conflict resolution*

After the training process, the model cannot be directly used for conflict resolution. In this way, we build the conflict resolution workflow with the trained model (Fig. 6).

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

The conflict resolution process starts by detecting potential conflicts between flights. If a conflict is detected between two flights $f_i$ and $f_j$, the process determines the owner and intruder flights based on their priorities. The flight with lower priority is considered the owner, and the other is considered the intruder.

Next, all reachable routes $\left\{r_1^{f_i}, r_2^{f_i}, \ldots, r_m^{f_i}\right\}$ from the current node to the destination node for the owner flight are searched from the global graph. Each candidate route of flight combines with the intruder route $r^{f_j}$ to check for potential conflicts. All combined pairs $\left\{\left(r_1^{f_i}, r^{f_j}\right), \left(r_2^{f_i}, r^{f_j}\right), \ldots, \left(r_m^{f_i}, r^{f_j}\right)\right\}$ are then input into the trained model to obtain the conflict status.

If the input pair is predicted as a non-conflict, the route $r_m^{f_i}$ in the pair $\left\{\left(r_m^{f_i}, r^{f_j}\right)\right\}$ is considered a valid solution to the tactical conflict, and the flight can proceed with the selected candidate route without conflict.

The incorporation of uncertainties in the flight environment, such as wind turbulence, is of paramount importance in the design and operation of eVTOL vehicles. The micro weather conditions in urban environments are highly complex due to factors like building morphology, which significantly affects the normal flight of eVTOL vehicles. Ignoring these uncertainties can lead to unsafe flight operations and potential accidents.

The 3D wind profile $W$ is defined as a function $W(x, y, z)$, where $x$, $y$, and $z$ represent the three-dimensional spatial coordinates. At each flight level $z$, the wind turbulence map $W_{2D}(x, y)$ is two-dimensional, representing the wind intensity at that level. The discrete wind maps of all levels combine to form the complete 3D wind profile, which is given as:

$$W(x, y, z) = \{W_{2D}(x, y)\}_{z=0}^{H} \tag{13}$$

where $H$ represents the maximum flight level in the 3D space.

If the intensity of wind at specific areas, denoted by $w(x_i, y_i, z_i)$, is predicted to exceed the maximum wind resistance speed of the eVTOL vehicles, denoted as $v_{\max}$, certain nodes in the graph representing the flight environment are not reachable for some vehicles based on their flight performance.

The set of unreachable nodes $U$ can be defined as:

$$U = \{v_i | w(x_i, y_i, z_i) > v_{\max}\} \tag{14}$$

where the $v_i \in N$ is the node in the corridor graph $G = (N, E)$, $x_i, y_i, z_i$ are coordinates of the node $v_i$.

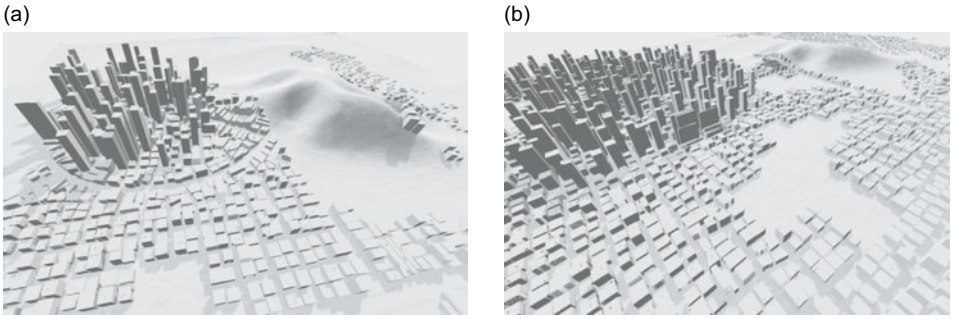
As the flight performance of eVTOL vehicles can vary depending on the specific vehicle's characteristics, such as its size, weight, propulsion system and other factors, the same wind data may result in different reachable nodes for different vehicles due to those attributes. Each vehicle's perspective on the global graph can be represented as $G_{fi} = \left(V, E_{fi}\right)$, where $E_{fi}$ denotes the set of edges accessible to $f_i$ under the influence of wind turbulence. Thus, wind turbulence affects the node connection in the global graph differently from the perspective view of each vehicle, resulting in different graphs $\left\{G_{fi}, G_{fj}, \ldots, G_{fk}\right\}$ for conflict resolution.

We can observe that the proposed framework adopts a centralised architecture where all flight trajectory data and conflict detection/resolution tasks are processed on a central server. This centralised design enables comprehensive conflict detection across all flights in the airspace, ensures consistent decision-making for conflict resolution and allows for global optimisation of flight path adjustments. The central processing unit receives flight schedules and trajectory information from all UAVs, constructs the conflict and non-conflict view pairs, executes the multi-view representation learning model and generates conflict resolution decisions. This architecture facilitates the efficient processing of multiple conflicts simultaneously while maintaining system consistency and enhanced safety through comprehensive conflict monitoring.

## 4.0 Experiments

In this section, we develop a comprehensive simulation environment that allows us to train and evaluate the proposed multi-view representation learning approach effectively. The results obtained from the

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press




Cityscape layout example 1. Cityscape layout example 2.

***Figure 7.*** *Layout samples of random cities.*

training and evaluation processes demonstrate the effectiveness of our approach in resolving conflicts between flights. Moreover, we consider the influence of uncertainties, such as wind turbulence, in our testing phase to assess the robustness and generalisability of the model.

### *4.1 Environment setup*

To ensure the representativeness of our simulated dataset, several key considerations were incorporated into the simulation design. The generated 3D cityscapes feature varying building densities, heights and spatial distributions that reflect typical urban environments. The flight routes are generated with practical operational constraints, including minimum separation requirements between aircraft, realistic flight speed ranges, etc. Each flight schedule is designed to represent common urban air mobility scenarios, such as point-to-point transportation, hub-to-hub routes and emergency service paths. This comprehensive approach ensures that our simulated trajectories capture the essential characteristics and challenges of real-world urban air operations

#### *4.1.1 Data generation*

We design a simulation environment to generate random 3D cityscapes with diverse building and terrain layouts following the procedure in Section 3.1.1. Specifically, two sample layouts are provided in Fig. 7, one with a map size of $3480m \times 3480m \times 277m$ and another with a map region of $5644m \times 5630m \times 287m$. These 3D cityscapes showcase various dimensions and morphologies, which have a significant impact on route generation, corridor structures and selection of vertiport locations. By simulating different urban configurations, we aim to generate a diverse and comprehensive dataset for our study, and how uncertainties, such as wind turbulence, affect the reachable nodes in the global graph from the perspective of different vehicles.

The generated 3D models are utilised to extract 2D map slices by cutting height cross-sections from the 3D space. Subsequently, we employ these 2D map slices to generate flight routes by selecting conflict-free waypoints at each flight level following steps in section 3.1.2.

Flight routes at altitudes of 90 m and 120 m are generated by randomly selecting waypoints as shown in Fig. 8. To ensure safety during flight planning, these routes are generated based on extended binary maps, which include a safe buffer margin around obstacles. The final flight routes are depicted on the original binary map without the buffer margin and heightmap, providing a clear and concise representation of the optimal flight paths. From Fig. 8a and 8b, as the altitude increases, the area occupied by obstacles decreases. This indicates that higher altitudes offer a more obstacle-free path for flight.

Once the routes at each level are determined, we establish connections between adjacent levels to facilitate altitude transfer. According to the generated flight corridor structure, flights are simulated from section 3.2.2. Some selective 3D trajectories are depicted in Fig. 9.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

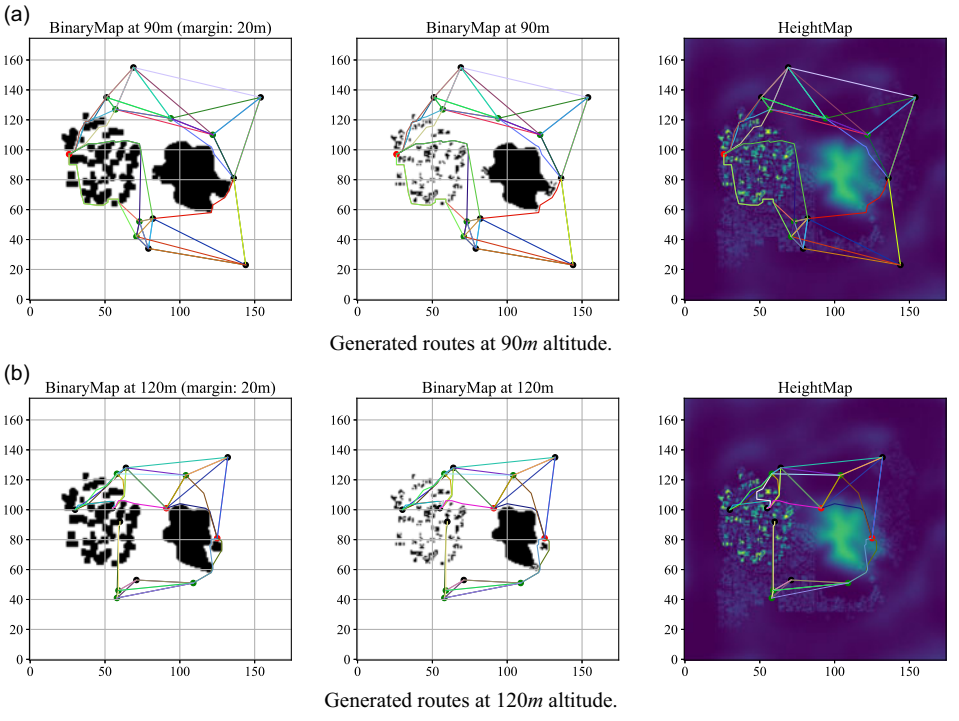



***Figure 8.*** *Generated routes at various altitudes.*

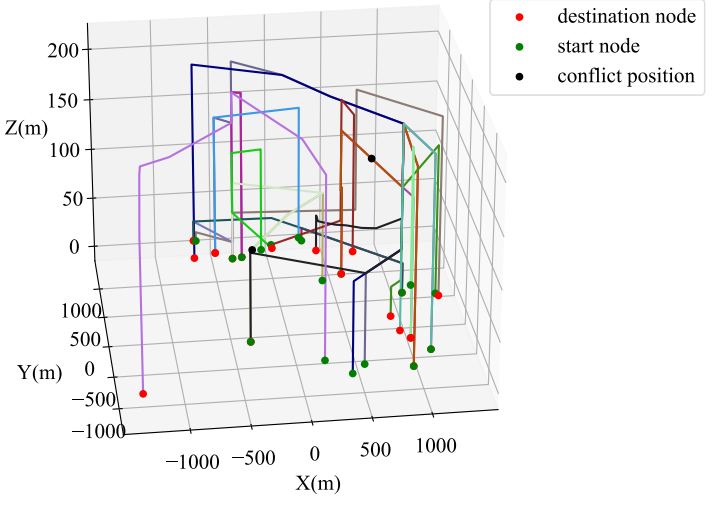



***Figure 9.*** *Selective flight trajectories.*

After schedules are generated, the dataset for training is constructed by collecting flight schedules on the graphs. For schedules with conflicts involving other flights, random segments of the trajectories containing the conflicts are chosen as samples. This selection process ensures that the dataset includes instances of conflict scenarios. On the other hand, for schedules without conflicts, one random segment of the trajectory is randomly chosen as a sample. By including both conflict and non-conflict samples

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

***Table 1.*** *Dataset summary*

| Dataset Statistics | Value |
|---|---|
| Total flight number | 100 |
| Generated route slices | 10,000 |
| Flight speed range | [4 m/s, 10 m/s] |
| Initial takeoff time | [0 s, 200 s] |
| Flight levels | [30 m, 60 m, 90 m, 120 m, 150 m, 180 m, 210 m] |
| Operation time horizon | [0, 3600 s] |
| Number of vertiports | 10 |

***Table 2.*** *Model configuration for multi-view representation learning*

| Layer | Input Size | Output Size | Num Layers |
|---|---|---|---|
| GraphConv1 | 6 | 20 | – |
| GraphConv2 | 20 | 5 | – |
| RNN | 5 | 2 | 3 |

in the dataset, the model is trained to differentiate between these scenarios, enhancing its performance in resolving conflicts during flight.

The dataset used in this study is summarised in Table 1. It consists of 100 total flight instances and 10 vertiports, resulting in a total of 10,000 randomly generated route slices. The flight speed range spans from 4 m/s to 10 m/s, and the initial takeoff time varies between 0 s and 200 s. Additionally, the dataset includes flights at different altitudes, specifically from 30 m to 210 m. The operation time horizon extends from 0 s to 3600 s.

#### *4.1.2 Model configuration*

The model configuration is a multi-view representation learning architecture that combines GNN and RNN as depicted in Table 2. It takes two graphs along with their corresponding node features as inputs. The GNN comprises two GraphConv layers with hidden sizes 20 and 5, respectively, which process the node features to extract informative representations. In the GraphConv layers, the node aggregation is performed using an RNN layer. The outputs from both branches of the GNN are then concatenated and passed through another RNN layer with a hidden size of 2. The RNN further processes the concatenated features, capturing temporal dependencies and interactions between nodes. Finally, the model computes the mean across the batch dimension to obtain the final output. This configuration enables the model to learn expressive and informative representations from multiple views of the input graphs, facilitating effective conflict resolution and uncertainty handling in the context of urban air traffic management.

### ***4.2 Results analysis***

#### *4.2.1 Model training*

The Fig. 10 depicts the training accuracy and training loss over 500 epochs. As the training progresses, the loss consistently decreases from an initial value of 0.7 to a final value of 0.19, indicating the effective learning process of the model. Simultaneously, the accuracy of classifying conflict and non-conflict pairs steadily increases and reaches an impressive accuracy of 97.4%. These promising training results demonstrate the efficacy of the proposed multi-view representation learning approach, and its ability to effectively differentiate between conflicting and non-conflicting flights.

Figure 11 illustrates the results of two-class clustering obtained during the training process using contrastive loss. Each subplot displays data points in two distinct colors, representing the conflict flight pair and non-conflict flight pair, respectively. The x and y coordinates of the points correspond to the

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

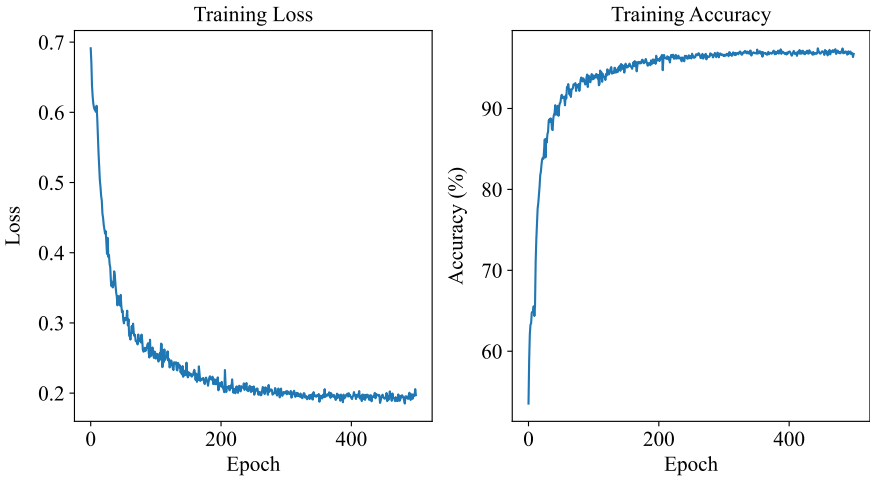



***Figure 10.*** *Training loss and accuracy.*

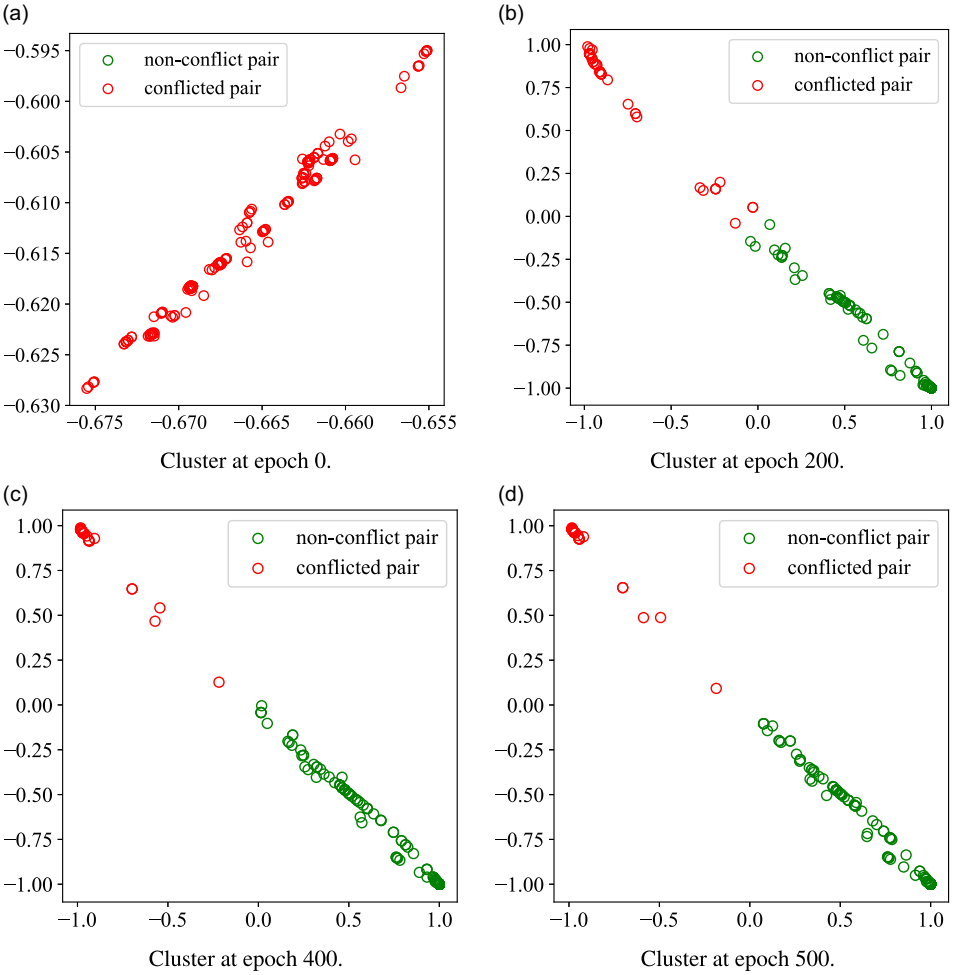



***Figure 11.*** *Clusters separation at each epoch.*

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

embeddings generated by the networks. These embeddings serve as a measure of the relative separation between the different classes. As the training proceeds, the embeddings are continuously adjusted, leading to an increased separation between the two classes. This enhanced separation is visually represented by the growing distance between the clusters of points.

For evaluating contrastive learning quantitatively, common metrics, e.g. silhouette coefficient, Calinski-Harabasz index and Davies-Bouldin index are used for clustering evaluation. The silhouette coefficient is a metric used to assess the quality of clustering results and measures how well-separated the clusters are, with values closer to 1 indicating well-separated clusters. Given a set of data points $X = \{x_1, x_2, \ldots, x_n\}$ and their corresponding cluster assignments $C = \{c_1, c_2, \ldots, c_n\}$, the silhouette coefficient is calculated as follows:

$$\mathrm{Si}(X, C) = \frac{1}{n} \sum_{i=1}^{n} \frac{b\,(x_i, C) - a\,(x_i, C)}{\max(a\,(x_i, C)\,, b\,(x_i, C))} \tag{15}$$

where:

- $a\,(x_i, C)$ is the average distance between data point $x_i$ and all other points in the same cluster as $x_i$.
- $b\,(x_i, C)$ is the average distance between data point $x_i$ and all points in the nearest cluster to which $x_i$ does not belong.

The silhouette coefficient ranges from $-1$ to 1, where a higher value indicates better-defined and well-separated clusters.

The Calinski-Harabasz index is another clustering evaluation metric that measures the ratio of between-cluster dispersion to within-cluster dispersion. It is given by:

$$\mathrm{CHI}(X, C) = \frac{\mathrm{Tr}(B\,(C))}{\mathrm{Tr}(W\,(C))} \times \frac{n-k}{k-1} \tag{16}$$

where:

- $n$ is the total number of data points.
- $k$ is the number of clusters.
- $B\,(C)$ is the between-cluster dispersion matrix.
- $W\,(C)$ is the within-cluster dispersion matrix.
- $Tr$ is the trace of matrix.

A higher Calinski-Harabasz index value indicates better-defined and more separated clusters.

The Davies-Bouldin index is yet another clustering evaluation metric that measures the average similarity between each cluster and its most similar cluster. It is given by:

$$\mathrm{DBI}(X, C) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left( \frac{S_i + S_j}{M_{ij}} \right) \tag{17}$$

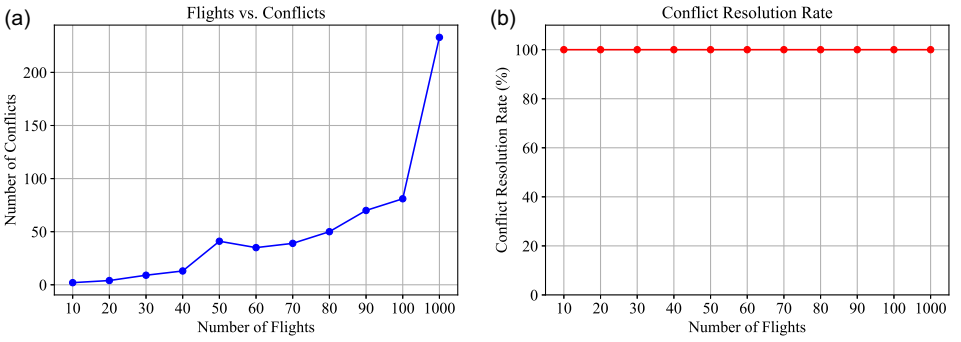
where:

- $k$ is the number of clusters.
- $S_i$ is the average distance between data points within cluster $i$.
- $M_{ij}$ is the distance between cluster centres $i$ and $j$.

The Davies-Bouldin index measures the average similarity between each cluster and its most similar cluster, with lower values indicating better-defined clusters.

The metrics comparison for data in Fig. 12 is demonstrated in Table 3. Each row represents the metrics' values at a specific epoch. At epoch 0, the metrics are not valid since the model's initial predictions only output one class, which results in incorrect clustering and classification. As the training progresses

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press




The conflict numbers vs. flight numbers. Conflict resolution rate.

***Figure 12.*** *Conflict resolution.*

***Table 3.*** *Metrics comparison for contrastive learning*

| Epoch | Silhouette Coefficient | Calinski-Harabasz Index | Davies-Bouldin Index |
|---|---|---|---|
| 0 | None | None | None |
| 200 | 0.791 | 997.087 | 0.317 |
| 400 | 0.816 | 939.86 | 0.222 |
| 500 | 0.824 | 1012.026 | 0.219 |

to epoch 200, the model starts to learn more meaningful representations, and the metrics show improved values. For example, the silhouette coefficient increases from 0.791 at epoch 200 to 0.824 at epoch 400, while the Calinski-Harabasz index shows a similar trend, increasing from 997.087 to 1012.026 during the same period. Additionally, the Davies-Bouldin index decreases from 0.317 to 0.219, reflecting the enhancement in clustering quality. These improvements in the evaluation metrics demonstrate the effectiveness of the contrastive learning approach in generating meaningful and separable representations of the data.

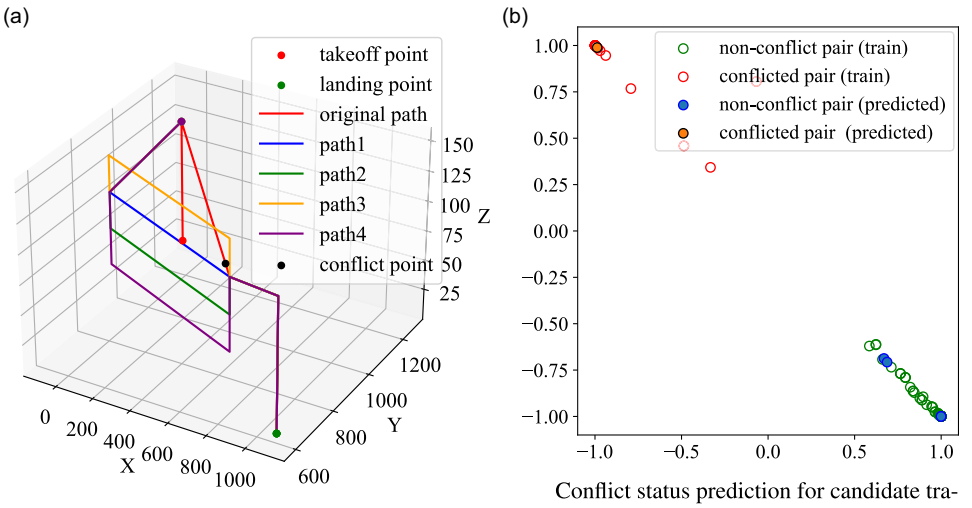
*4.2.2 Model test*

Once the model is trained and validated, it becomes capable of intelligent conflict resolution. By utilising the learned representations, the model can efficiently analyse and resolve conflicts between different flight schedules in a dynamic and real-time manner.

Following the workflow in section 3.3.3, we can propose several alternative routes and let the trained model select the best route for conflict resolution. The line graph in Fig.12a illustrates the relationship between the number of flights and the corresponding number of conflicts in different simulated scenarios. The x-axis represents the total number of flights, ranging from 10 to 100 with increments of 10, while the y-axis represents the number of conflicts observed for each specific number of flights. As depicted in the graph, there is a positive correlation between the number of flights and the number of conflicts. The number of conflicts gradually increases as the number of flights increases. Specifically, with 10 flights, there are only two conflicts, and as the number of flights increases, the number of conflicts progressively rises.

The graph in Fig. 12b reveals that the conflict resolution rate remains at a remarkable 100% for all flight numbers. This indicates the effectiveness of the multi-view representation learning technique in effectively resolving conflicts among flights. With this approach, regardless of the increasing complexity and number of flights in the airspace, the model successfully identifies and resolves conflicts, ensuring the safety and efficiency of flight operations.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press




The candidate trajectories for a sample flight.

Conflict status prediction for candidate trajectories.

***Figure 13.*** *Conflict resolution example.*

***Table 4.*** *Original path nodes and candidate path nodes*

| Original Path Nodes | Candidate Path Nodes |
|---|---|
| [158,161,162,288] | Path 1: [158,159,161,162,288] |
| | Path 2: [93,95,158,159,161,162,288] |
| | Path 3: [158,159,161,162,175,177,288] |
| | Path 4: [58,60,93,95,158,159,161,162,288] |

In addition, a test scenario with 1000 aircraft is included, which serves to validate the performance of our model when the number of aircraft within a specific area increases dramatically. As shown in Fig. 12a, it can be observed that the number of tactical conflicts rapidly increases to 233 as the number of aircraft reaches 1000. However, as depicted in Fig. 12b, the model remains capable of resolving these tactical conflicts effectively.

We show one example of conflict resolution. The candidate flight trajectories are shown in Fig. 13a. Table 4 illustrates the variation in traversed nodes for the original path and four candidate paths generated during the search process. Each row in the table represents a candidate path, and it originates and terminates at the same beginning and end nodes. The prediction results in Fig. 13b reveal that the first candidate trajectory, of which node indices [158,159,161,162,288] are predicted to still have a conflict because the traversed nodes are too similar to the original routes. However, three others serve as appropriate candidate solutions to resolve this conflict.

To evaluate the impact of the candidate solutions on the flight, we analyse the increased flight time and distance. These metrics provide valuable insights into how each candidate's trajectory affects the overall flight operations. As depicted in Fig. 14, the increased distance for each candidate trajectory is [1484.68 $m$, 1544.68 $m$, 1544.68 $m$, 1604.68 $m$], with an average of 1544.68 $m$. The increased flight time is [46.58 $s$, 61.58 $s$, 61.58 $s$, 76.58 $s$], with an average of 61.58 $s$. By quantifying the additional time and distance required for each candidate route compared to the original path, we can assess the trade-offs and benefits of potential conflict resolution strategies. Among all candidate routes, path 2 has the smallest increase in both flight time (61.58 $s$) and distance (1544.68 $m$) compared to the original path among the remaining candidate solutions. Therefore, it is the recommended choice for conflict resolution as it minimises the impact on flight time and distance while effectively resolving the conflict.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

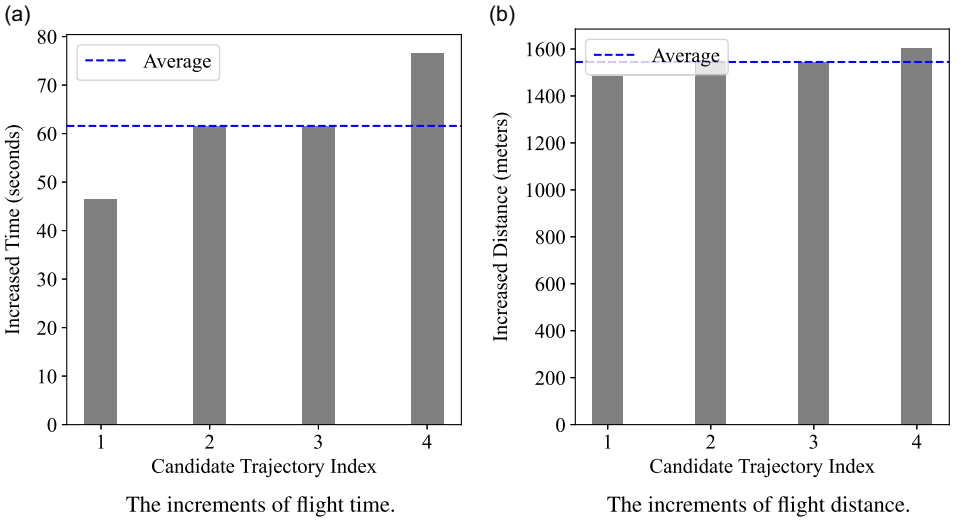



***Figure 14.*** *The increment parameters of candidate trajectories.*

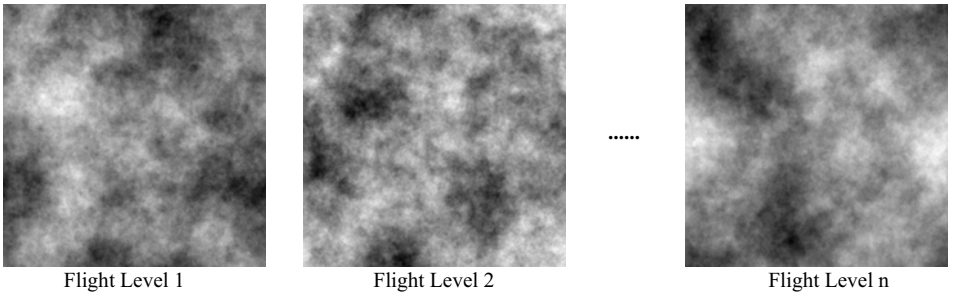



***Figure 15.*** *Random wind turbulence at different flight levels.*

The presented example showcases the process of resolving a specific conflict by selecting the most suitable candidate trajectory while minimising the impact on flight time and distance. However, it is essential to note that this approach is not limited to just one conflict but can be applied iteratively to handle multiple conflicts simultaneously. The multi-view representation learning technique efficiently processes and analyses various conflicts in the environment, providing intelligent and automated solutions for conflict resolution.

*4.2.3 Model test with uncertainties involved*

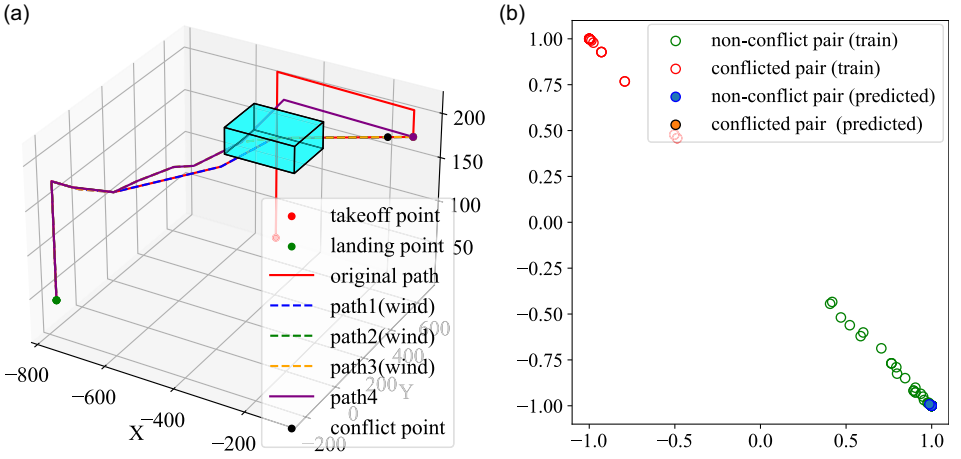
This work also investigates the importance of uncertainties, such as wind turbulence, in the flight environment. The 2D wind turbulence map is generated using a Gaussian random field approach by computing the 2D inverse Fourier transform of a noise signal modulated by the amplitude calculated from the power spectrum. This spectral method captures both the spatial correlations and the multi-scale nature of atmospheric turbulence, providing a realistic wind environment for testing our conflict resolution approach.

Through an iterative process, a series of 2D wind maps are generated to represent wind conditions at various flight levels. For instance, some wind maps are demonstrated in Fig. 15. To create a seamless and comprehensive 3D wind model, linear interpolation is utilised to estimate wind turbulence at positions lying between discrete flight levels.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

***Table 5.*** *Original path nodes and candidate path nodes in the presence of wind*

| Original Path Nodes | Candidate Path Nodes |
|---|---|
| [209,210,223,231,237–239,313] | Path 1: [209,210,223,230,231,237,238,313]<br>Path 2: [209,210,223,231–235,239,313]<br>Path 3:[209,210,223,230–235,313]<br>Path 4:[209,211,223,231–235,239,313] |




The candidate trajectories for a sample flight in the presence of wind.

Conflict status prediction for candidate trajectories in the presence of wind.

***Figure 16.*** *Conflict resolution example in the presence of wind.*

In accordance with the details presented in section 3.3.3, the influence of wind turbulence on flight operations is thoroughly considered. Wind turbulence not only impacts flight performance but also affects the accessibility of certain nodes in different regions. Specifically, in the view of different flights, some node connections within specific areas will be invalidated due to the intensity of wind turbulence. Consequently, this aspect of the model accounts for the variations in reachability across different flight paths and enables a more comprehensive evaluation of potential conflicts in the airspace.

We examine a specific conflicting pair of flight schedules, and the original flight path for the ownership is provided in Table 5. The flight schedule node indices [209,210,223,231,237–239,313] represent the sequence of nodes in the original flight path. However, when considering wind turbulence, it is predicted that the flight performance will be affected within a specific time range [140.38 $s$, 160 $s$] and near the coordinate location of [−410 $m$, 410 $m$, 180 $m$] within a 100-meter region.

To resolve this conflict, we search for candidate routes that avoid the problematic wind area. As shown in Fig. 16a, among the candidate routes, only route 4 is deemed acceptable as it allows the flight to traverse the wind area at a time (154 $s$) that is within acceptable flight performance limits. The other routes (1, 2 and 3) are deemed unsuitable as they would lead to the flight encountering adverse wind conditions at an unacceptable time. This process demonstrates the importance of considering wind turbulence and its impact on flight paths during conflict resolution.

The multi-view representation learning approach is further applied to evaluate the suitability of path 4 as a resolution to the tactical conflict. The results shown in Fig. 16b confirm that path 4 indeed serves as a favourable solution for resolving the conflict. Additionally, the increment in the distance (1281.57 $m$) and increased flight time (34.84 $s$) resulting from this solution are both deemed acceptable performance measures for the involved flights. The successful application of multi-view representation

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

learning demonstrates the effectiveness of this approach in assessing and resolving tactical conflicts in flight schedules while considering various conditions.

In conclusion, the multi-view representation learning approach has proven to be an efficient method for tactical conflict resolution. Through the generation of wind models, it accurately assesses the impact of wind conditions on flight paths. The multi-view representation learning also enables the identification of suitable candidate routes for resolving conflicts. The successful validation of this approach with multiple conflict scenarios demonstrates its robustness and effectiveness in enhancing safety and efficiency in urban airspace management.

## 5.0 Conclusion

In this work, we propose a novel multi-view representation learning framework designed specifically for tactical conflict resolution in diverse city regions. To capture the complexities of 3D cities, we incorporated random city layouts, including buildings and terrains, to create a comprehensive 3D environment. By generating flight corridors based on 2D height maps and constructing a graph structure, we created a dynamic representation of the airspace. Based on this graph structure, we design the multi-view representation learning approach, which combines GNNs and RNNs to learn represented features. In addition, contrastive learning allows it to effectively learn intricate relationships between flight paths and improve resolution performance.

The promising results obtained from diverse scenarios, including uncertain environments affected by wind turbulence, highlight the effectiveness of our method in various scenarios and its contribution to advancing the field of conflict resolution in the context of urban air mobility.

However, it is important to acknowledge the limitations of our study, primarily the focus on simulation rather than real-world tests with hundreds of flights, and the simplified wind model used in the urban city environment. As a future direction, we aim to consider more realistic environment configurations, which will involve more comprehensive wind models and real-world data to validate the performance of our approach. This will lead to more robust and practical solutions for enhancing safety and efficiency in urban air mobility applications.

**Acknowledgements.** This research was partially supported by grants from the Funds of China Scholarship Council (202008420248).

**Competing interests.** The authors declare none.

Cheng Huang contributed to the architecture design, algorithm proposal and implementation and writing of this paper; Ivan Petrunin and Antonios Tsourdos helped with the result analysis and revision of the manuscript.

## References

[1] Urban Air Mobility (UAM) – Concept of Operations v1.0, Federal Aviation Administration, 1–37, June 2020.

[2] Brown, A. and Harris, W.L. Vehicle design and optimization model for urban air mobility, *J. Aircraft*, 2020, **57**, (6), pp 1003–1013.

[3] Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K.-D., Kaiser, J., and Plötner, K.O. An overview of current research and developments in urban air mobility–setting the scene for UAM introduction, *J. Air Transport Manag.*, 2020, **87**, p 101852.

[4] Comer, A.M and Chakraborty, I. Flight control system architecture for urban air mobility simplified vehicle operations, AIAA SCITECH 2023 *Forum*, 2023, p 0399.

[5] Bauranov, A. and Rakas, J. Designing airspace for urban air mobility: a review of concepts and approaches, *Prog. Aerosp. Sci.*, 2021, **125**, p 100726.

[6] Organisation de l'aviation civile internationale. Global Air Traffic Management Operational Concept, ICAO, Montréal, *Canada*, 2005.

[7] Pham, D.-T., Tran, N.P., Goh, S.K., Alam, S., and Duong, V. 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, Danang, Vietnam, 2019, pp 1–6.

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press

[8] Hill, B.P., DeCarme, D., Metcalfe, M., Griffin, C., Wiggins, S., Metts, C., Bastedo, B., Patterson, M.D. and Mendonca, N.L. UAM vision concept of operations (ConOps) UAM maturity level (UML) **4**, NASA Technical Management, Washington, DC, 2020.
[9] Mohamed Salleh, M.F.B., Wanchao, C., Wang, Z., Huang, S., Tan, D.Y., Huang, T. and Low, K.H. Preliminary concept of adaptive urban airspace management for unmanned aircraft operations, in 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, Kissimmee, Florida, 2018, p 2260.
[10] Bankole, I. Urban air mobility (UAM) flight path: literature review and conceptual design of UAM corridor virtual lane system using "Tracks", Graduate project, Embry-Riddle Aeronautical University, Daytona Beach, FL, February 1, 2023, pp 1–31.
[11] Badea, C.A., Morfin Veytia, A., Patrinopoulou, N., Daramouskas, I., Ellerbroek, J., Lappas, V., Kostopoulos, V., and Hoekstra, J. Unifying tactical conflict prevention, detection, and resolution methods in non-orthogonal constrained urban airspace, *Aerospace*, 2023, **10**, (5), p 423.
[12] Bae, S., Shin, H.-S. and Tsourdos, A. Structured urban airspace capacity analysis: four drone delivery cases, *Appl. Sci.*, 2023, **13**, (6), p 3833.
[13] Brittain, M.W. and Wei, P. One to any: distributed conflict resolution with deep multi-agent reinforcement learning and long short-term memory, in AIAA Scitech 2021 Forum, American Institute of Aeronautics and Astronautics, 2021, p 1952.
[14] Zhao, P., Erzberger, H. and Liu, Y. Multiple-aircraft-conflict resolution under uncertainties, *J. Guid. Control Dyn.*, 2021, **44**, (11), pp 2031–2049.
[15] Pelegrín, M., d'Ambrosio, C., Delmas, R. and Hamadi, Y. Urban air mobility: from complex tactical conflict resolution to network design and fairness insights. Optimization Methods and Software, **38**, (6), pp 1311–1343, 2023.
[16] Tang, H., Zhang, Y. and Post, J. Predeparture flight planning to minimize operating cost for urban air mobility, *J. Air Transport.*, 2023, **31**, (4), pp 1–11.
[17] Jover, J., Bermúdez, A. and Casado, R. A tactical conflict resolution proposal for U-space Zu airspace volumes, *Sensors*, 2021, **21**, (16), p 5649.
[18] Causa, F., Franzone, A. and Fasano, G. Strategic and tactical path planning for urban air mobility: Overview and application to real-world use cases, *Drones*, 2022, **7**, (1), p 11.
[19] Zhang, M., Chao, Y., Dai, W., Xiang, X. and Low, K.H. Tactical conflict resolution in urban airspace for unmanned aerial vehicles operations using attention-based deep reinforcement learning, *Green Energy Intell. Transport*, 2023, **2**, (4), p 100107.
[20] Yang, X., Deng, L. and Wei, P. Multi-agent autonomous on-demand free flight operations in urban air mobility, in *AIAA Aviation 2019 Forum*, Dallas, TX, 2019, p 3520.
[21] Chen, S., Evans, A., Brittain, M. and Wei, P. Integrated conflict management for UAM with strategic demand capacity balancing and learning-based tactical deconfliction, *arXiv preprint arXiv:2305.10556*, 2023.
[22] Jiang, W. and Luo, J. Graph neural network for traffic forecasting: a survey, *Exp. Syst. Appl.*, 2022, **207**, p 117921.
[23] Wu, N., Zhao, X.W., Wang, J. and Pan, D. Learning effective road network representation with hierarchical graph neural networks, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, 2020, pp 6–14.
[24] Jin, G., Yan, H., Li, F., Huang, J. and Li, Y. Spatio-temporal dual graph neural networks for travel time estimation, *arXiv preprint arXiv:2105.13591*, 2021.
[25] Parish, Y.I.H. and Müller, P., Procedural modeling of cities, in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 2001, pp 301–308.
[26] Wright, W.E. Parallelization of Bresenham's line and circle algorithms, *IEEE Comput. Graph. Appl.*, 1990, **10**, (5), pp 60–67.

**Cite this article:** Huang C., Petrunin I. and Tsourdos A. (2025). Multi-view representation learning for performance-based tactical conflict resolution in urban air mobility. *The Aeronautical Journal*, **129**, 2240–2262. https://doi.org/10.1017/aer.2025.30

https://doi.org/10.1017/aer.2025.30 Published online by Cambridge University Press