

Multi-agent Estimated Time of Arrival prediction and dynamic arrival sequencing by Emulating Air Traffic Controllers[☆]

Hong-Cheol Choi^a,^{*} Chuhao Deng^a, Hyunsang Park^a, Jaeyoung Ryu^b, Hak-Tae Lee^b, Inseok Hwang^a

^a School of Aeronautics and Astronautics, Purdue University, West Lafayette, 47907, IN, USA

^b Department of Aerospace Engineering and the Program in Aerospace Systems Convergence, Inha University, Incheon, 21999, Republic of Korea

ARTICLE INFO

Keywords:

ETA prediction
Arrival sequencing
Multi-agent system
Attention mechanism
Machine learning
Air traffic management

ABSTRACT

Accurate Estimated Time of Arrival (ETA) prediction is critical to the air traffic management system including aircraft sequencing for which Air Traffic Controllers (ATCs) are responsible. Although significant advancements have been achieved in both ETA prediction and arrival sequencing, the development of decision support tools can be further improved by learning the expertise of ATCs and reflecting on their practical considerations. To fill the research gap, in this paper, we propose a multi-agent model for both ETA prediction and arrival sequencing based on the attention mechanism that can account for the current air traffic situation and capture the decisions made by ATCs. The proposed model is demonstrated with real air traffic surveillance data recorded at Incheon International Airport in South Korea and compared with existing models in terms of ETA prediction, sequence similarity, and arrival sequencing performance. The experimental results show that, in a real-time manner, the proposed model can provide landing sequences more acceptable to ATCs as well as more accurate ETAs than those of comparison models. Specifically, sequence similarity is measured by two rank correlation coefficients, which shows the superiority of the proposed model in emulating ATC decisions. Furthermore, important considerations in arrival sequencing are discussed based on actual ATC feedback.

1. Introduction

The Air Traffic Management (ATM) system encompasses complex and safety-critical operations which are mainly managed by Air Traffic Controllers (ATCs) and pilots to ensure safety and efficiency. This air traffic operation becomes more complex and challenging as demands continue to increase. Indeed, the demand for air transport is expected to increase by an average of 4.3% annually over the next 20 years, and the projected number of flights is expected to reach around 90 million by 2040 (ICAO, 2019). This continuous growth of demands can lead to an excessive workload for both ATCs and pilots, thereby resulting in the degradation of the ATM system. To effectively respond to this problem, a lot of effort has been put into developing decision support tools.

One of the tasks where such a decision support tool is most needed is arrival sequencing which has a significant impact on the efficiency of the overall ATM system in minimizing delays. This task is a crucial decision-making process to determine a landing sequence for multiple incoming aircraft, facilitating the timely and orderly arrival of aircraft

to their designated destinations. The primary objectives of arrival sequencing are to optimize traffic flow, alleviate congestion, and improve airspace safety. To achieve these objectives, Arrival Manager (AMAN) was developed, which assists ground-based ATCs in establishing safe and efficient arrival sequences to a designated airport (Eurocontrol, 2010). The tool, AMAN, works in two main steps. Initially, Estimated Time of Arrival (ETA) prediction is performed for each aircraft based on current states. Subsequently, the optimal sequence of arrivals builds on the ETAs of the arriving aircraft, aiming to minimize overall flight delays and/or maximize airport throughput, while maintaining safety.

To develop arrival sequencing models for AMAN, various approaches have been taken in the existing literature. Beasley et al. (2000) formulated aircraft sequencing and scheduling for landings as a mixed integer linear programming problem. The study aimed to identify the optimal sequence of arrivals and individual landing times for various runway configurations. Due to the complexity of finding the optimal solution, arrival sequencing is known as an NP (Nondeterministic Polynomial time)-hard problem. For practical applications, heuristic

[☆] This paper is an extension of work originally presented in DASC 2023 (Choi et al., 2023b).

^{*} Corresponding author.

E-mail addresses: choi642@purdue.edu (H.-C. Choi), deng113@purdue.edu (C. Deng), park1375@purdue.edu (H. Park), jaeyoungryu@inha.edu (J. Ryu), haktae.lee@inha.ac.kr (H.-T. Lee), ihwang@purdue.edu (I. Hwang).

<https://doi.org/10.1016/j.jairtraman.2025.102828>

Received 27 March 2024; Received in revised form 5 April 2025; Accepted 20 May 2025

Available online 11 June 2025

0969-6997/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

approaches have instead been adopted to find a near-optimal sequence of arrivals. To efficiently find feasible arrival sequences, Ernst et al. (1999) proposed a specialized simplex algorithm with a metaheuristic. In addition, the Genetic Algorithm (GA) was widely used to generate nearly optimal solutions within a reasonable time (Beasley et al., 2001; Salvatore and Ignaccolo, 2004). Furthermore, Hancerliogullari et al. (2013) proposed metaheuristics using greedy algorithms as initial solutions, which results in better performance for the real-world application of the aircraft sequencing problem.

Recognizing the dynamic nature of the ATM environment and the practical issues that affect actual operations, various new approaches to arrival sequencing have been proposed. Dear (1976) and Balakrishnan and Chandran (2006) proposed Constrained Position Shifting (CPS) based on a practical consideration. CPS is initialized with First-Come First-Served (FCFS) sequences of arrivals and only allows a predefined number of position shifts from FCFS orders, which is determined by sorting ETAs. To address the issues that arise in the dynamic environment near terminal airspace, dynamic arrival sequencing has been solved based on the displacement problem (Beasley et al., 2004) or the Receding Horizon Control (RHC) algorithm (Hu and Chen, 2005). Ben-nell et al. (2017) proposed to use dynamic programming and local search heuristics to solve the dynamic problem, which requires periodic updates to the previous arrival sequence with the newly available aircraft.

Although previous studies have shown great progress and their optimization tools have helped ATCs, the actual arrival sequence adopted by ATCs often deviates from that presented by optimization algorithms. This can be explained by the limitations of existing algorithms that do not fully accommodate the decision-making processes of ATCs and/or the dynamic nature of the environment (Tang and Abbass, 2014; Jung et al., 2018). To address this practical issue, data-driven approaches have been adopted. Tang and Abbass (2014) employed a probabilistic finite-state machine and GA to derive ATC heuristics for aircraft sequencing with simulated aircraft data. In the recent study (Jung et al., 2018), Pairwise Preference Learning (PPL) was proposed to directly accommodate the cognitive processes of actual ATCs, which are important for practical application to actual operations. However, PPL has insufficient capabilities to capture the dependence that the landing order of each aircraft is affected by the ordering decisions of other aircraft, and its computational burden to consider all possible pairs increases quadratically as the number of aircraft increases.

Moreover, existing studies perform ETA prediction and arrival sequencing separately, as AMAN works. Consequently, the performance of arrival sequencing is greatly dependent on the accuracy of ETA prediction models. Du et al. (2023) showed that a more accurate ETA prediction can contribute to reducing the average delay and dwell time as well as the deviation from the actual landing sequence. In this regard, various data-driven models have been used to improve the accuracy of the ETA prediction. Gui et al. (2021) utilized the conventional machine learning algorithm (i.e., extreme gradient boosting), and Wang et al. (2020b) proposed the automated method that stacks multiple existing models. In addition, the neural network-based approach has been widely used, such as recurrent neural network (Ayhan et al., 2018), bidirectional long short-term memory (Wang et al., 2020a), spatiotemporal neural network (Ma et al., 2022), and clustering-based deep neural network (Deng et al., 2023). However, despite these advancements, existing works have focused only on the single-agent ETA prediction model that assumes each aircraft to be independent and neglects the effects of neighboring aircraft and air traffic control in congested terminal airspace.

Therefore, to address the aforementioned limitations, this paper proposes a multi-agent model as a decision support system for two intertwined problems: ETA prediction and arrival sequencing. Firstly, we employ the attention mechanism to better understand the interdependencies of multiple aircraft and emulate ATC's decisions in arrival sequencing. By incorporating the separation constraint into the loss

function and utilizing a novel mixed training strategy, a single model can accommodate both ETA prediction and arrival sequencing, resulting in better performance in each task. Moreover, this multi-agent model helps avoid the generation of unrealistic arrival sequences that result from the combination of errors in the ETA prediction model and inaccurately allowed deviations in the sequencing model. The performance of the proposed model is demonstrated with extensive experiments on both ETA prediction and arrival sequencing and analyzed in terms of various metrics. Furthermore, blind tests with actual ATCs as subjects and their feedback reveal important considerations that traditional optimization-based algorithms can fail to reflect.

The remainder of this paper is organized as follows. Section 2 describes the specific problem, domain background, and data preparation for this study. Section 3 presents the preliminary of the attention mechanism and then proposes the multi-agent model to emulate ATCs' decisions. In Section 4, the performance of the proposed model is evaluated using the dataset prepared in Section 2 and analyzed in terms of ETA prediction, sequence similarity, and arrival sequencing, respectively. Lastly, the paper concludes with a summary of the findings and outlines potential future research in Section 5.

2. Problem description

In this section, we first clarify the air traffic problems (ETA prediction and arrival sequencing) studied in this paper. Secondly, we describe the data used for our data-driven model and domain knowledge regarding the specific terminal airspace, where the data was collected.

2.1. Problem statement

This paper focuses on providing a realistic landing sequence and corresponding arrival times for multiple aircraft in terminal airspace, designed to function as a decision support system for ATCs. Specifically, two intertwined problems (i.e., ETA prediction and arrival sequencing) are addressed at once. By taking the state and traffic information as input (e.g., position, speed, and travel time), the generated advisories reflect ATC decisions and practical considerations by emulating ATCs from past operations recorded in the historical dataset. In addition, this paper considers dynamic scenarios in which the number of aircraft keeps changing over time, requiring iterative updates in real time.

This problem necessitates a new approach as shown in Fig. 1, while the existing approach solves the ETA prediction and arrival sequencing problems sequentially. In the existing approach, the track points of each aircraft are fed into an ETA prediction model. Once all the ETAs are computed and collected, they are then utilized for a sequencing algorithm to determine the optimal sequence of arrivals for the current air traffic situation. In contrast, on the right side, all the track points of multiple aircraft are directly fed into one multi-agent model trained for implicitly learning ATC intentions and decisions through agent interactions.

2.2. Data preparation

In this paper, we collect and utilize the arrival trajectories at Incheon International Airport (ICN), South Korea, recorded in the Automatic Dependent Surveillance-Broadcast (ADS-B) data between January and May 2019. We also take advantage of Aeronautical Information Publication (AIP) data in conjunction with the ADS-B data. The ADS-B data consists of various aircraft states such as longitude, latitude, altitude, ground speed, vertical speed, and course angle, as well as flight information and time. In order to focus on the terminal airspace operation, we extract regulation and domain knowledge from the AIP data. For instance, entry fixes, Standard Terminal Arrival Routes (STARs), and instrument approach procedures are denoted with the triangle markers and white dotted lines in Fig. 2. Around ICN, there are four

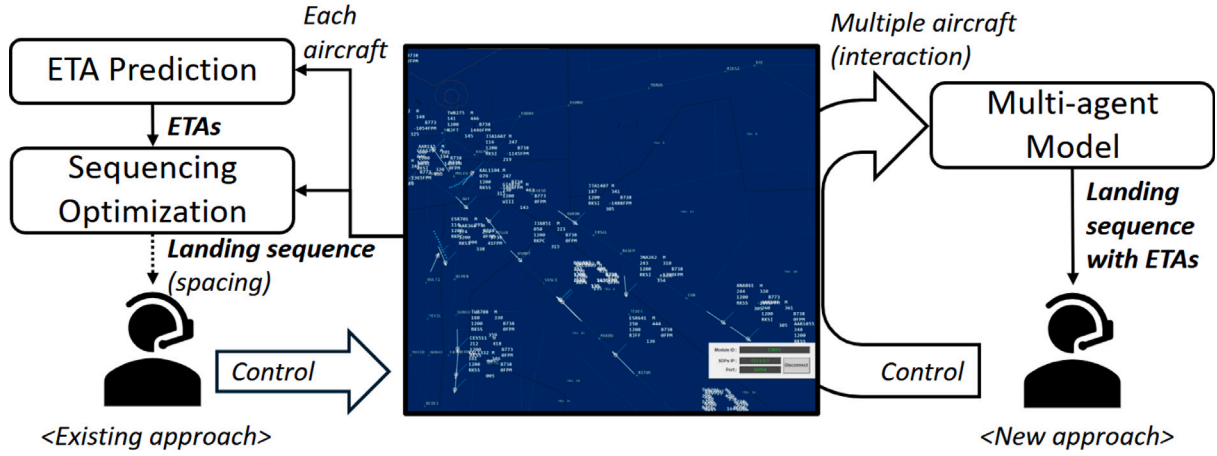


Fig. 1. Comparison between existing and new approaches to the arrival sequencing problem.

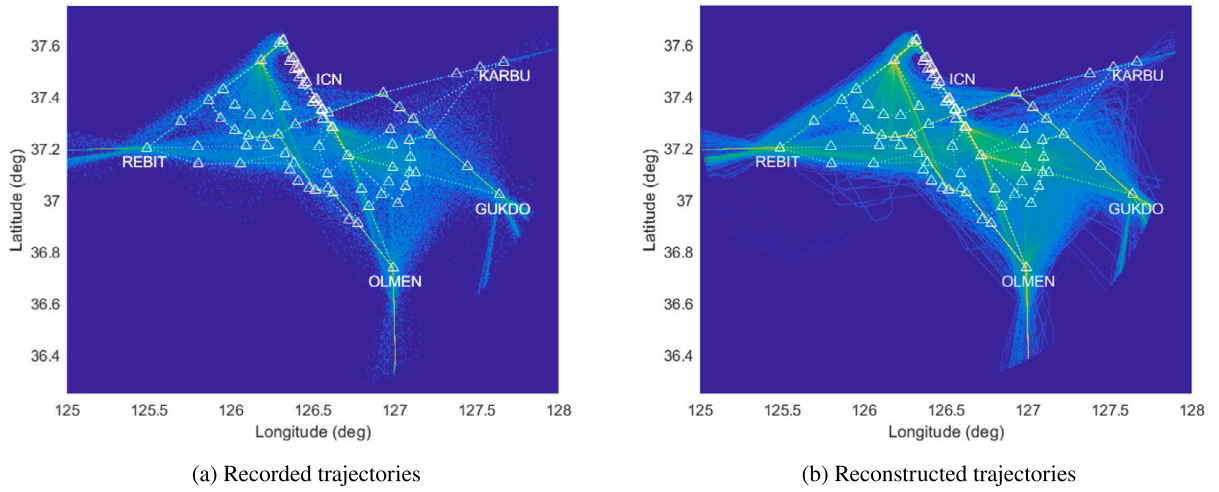


Fig. 2. Illustration of historical aircraft trajectories and entry fixes in ICN terminal airspace.

different entry fixes to enter terminal airspace, i.e., KARBU, GUKDO, OLMEN, and REBIT. Aircraft entering through KARBU are predominantly from North America, whereas those passing through GUKDO generally come from Oceania and Japan. Aircraft passing through REBIT typically originate from Europe and China, while those traversing OLMEN come from Southeast Asia and Jeju. Upon crossing one of four entry fixes, sector controllers hand off each aircraft to approach controllers.

Firstly, we process each aircraft trajectory for ETA prediction and arrival sequencing. In terminal airspace, we limit the length of the trajectories from the Final Approach Fix (FAF) to 70 nmi from the airport. The distance of 70 nmi is determined as the extended distance before entering terminal airspace through the four entry fixes. The data point at the FAF is used as the arrival time instead of landing at the runway due to the lack of recorded data near the airport, which makes the actual landing time unavailable. This is because most of the last data points are recorded before the actual aircraft lands on the runway, whereas few of them have data points on the runway. Therefore, to unify them to the same point, we enforce the last data point of all data to end at the FAF and set the time as the arrival time. Additionally, although the ADS-B data provides flight information which includes aircraft types and call signs, there are so many different kinds, which makes it difficult for data-driven models to understand the relationship between neighboring aircraft. Instead, using the Wake Turbulence Category (WTC), which prescribes the minimum separation requirement between aircraft, the aircraft can be categorized and labeled with one

of five categories (super, heavy, medium, light, and unknown case). In addition, to account for the aircraft's travel time in terminal airspace, we compute the travel time as the time flown in terminal airspace, starting at zero when the aircraft is 70 nmi away from the airport. Therefore, along with the state information (position, speed, and course angle), the travel time and the WTC information are also used for model training to improve the accuracy of the model. Therefore, each aircraft trajectory includes a total of 8 features with a 10-s interval.

Additionally, the recorded ADS-B data need to be preprocessed to resolve irregular sampling rates, which typically range from 20 to 60 s. The remaining trajectories are then reconstructed using a regularized least-squares optimization (Barratt et al., 2018; Deng et al., 2024b), which is equivalent to solving the optimization problem:

$$\underset{P}{\text{minimize}} \quad \|HP - \hat{P}\|_F^2 + \lambda_1 \|D_2 P\|_F^2 + \lambda_2 \|D_3 P\|_F^2 \quad (1)$$

where the optimization variable $P \in \mathbb{R}^{L \times 3}$ is the reconstructed trajectory of length L , and $\hat{P}$ is the measurement matrix. H is a diagonal matrix to indicate whether the data is measured or not at a given time. D_2 and D_3 are second-order and third-order difference matrices, respectively. λ_1 and λ_2 are regularization hyper-parameters. As a result, it is observed that the reconstructed trajectories in Fig. 2(b) are smooth and have a constant sampling rate compared to the originally recorded trajectories in Fig. 2(a).

Lastly, for a multi-agent system, we need to collect multiple aircraft trajectories at each time instance in the form of a traffic scene (i.e., a

sample). Herein, to account for dynamically changing traffic, we set the time window for observation to 2 min. In other words, we utilize each aircraft's trajectory from two minutes ago to the present to predict its ETA at the current time. From the collected dataset, we exclude traffic scenes that contain three cases: (i) go-around, (ii) diversion, and (iii) separation violation. These cases could significantly deteriorate the performance of the ETA prediction due to abnormal flight times caused by multiple approach and landing attempts or landing at different nearby airports. Furthermore, we consider traffic situations where only one runway (runway 15L or runway 33R) is open for arrival flights. The dataset is split into two subsets based on the landing direction: flights heading southbound and northbound to ICN. A total of 11,398 and 17,738 samples are collected for the northwest and southeast datasets. In each case, 80% are randomly selected for training, while 10% each is used for validation and testing, respectively.

3. Methodology

In this section, we first summarize the attention mechanism that is the basis for the proposed approach. Secondly, the proposed multi-agent model (in Fig. 3) is explained in detail, especially focusing on agent-aware attention and mixed training strategy.

3.1. Attention mechanism

In natural language processing or neural machine translation, the attention mechanism is a powerful technique that can address the issue when considering the entire context of a long input sequence. This mechanism can give more importance to specific and relevant parts of the input sequence while making predictions. For example, the standard attention mechanism selectively attends to specific words in a sentence or parts of an image by mimicking the human attention process.

Especially, for self-attention, the scaled dot-product attention function is introduced in the attention layer within the Transformer architecture (Vaswani et al., 2017). The self-attention function works by creating three vectors from the encoded input sequence: queries, keys, and values. In the ETA prediction application, queries work like questions about the relevance between a specific track point and other observations (position, speed, and course angle) that can help predict the ETA. Keys represent all other points being compared to the current point, and values contain the actual information associated with each point in the input sequence for ETA prediction. An attention function takes a query and a set of key-value pairs as input and computes the dot product between the query and all keys, followed by scaling the result down by $\sqrt{d_k}$, where d_k represents the square root of the dimension of the key vectors. This scaling is crucial to prevent numerical instability caused by large dot products. Subsequently, a softmax function is applied to the compatibility function (the dot product between the query and the key vectors) to derive the relevance weights on the values. In practice, as an attention function operates on a set of queries concurrently, the queries, keys, and values are arranged into matrices denoted as Q , K , and V , respectively. The scaled dot-product attention can be expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

Finally, the mechanism generates an output that summarizes the most important information from the input sequence for the current prediction.

The multi-head attention function extends the capabilities of the scaled dot-product attention by projecting the queries, keys, and values linearly into multiple subspaces. This technique allows the deep learning model to attend to and integrate information from various representation subspaces and positions. In the case of ETA prediction, using multiple attention heads allows the model to focus on various

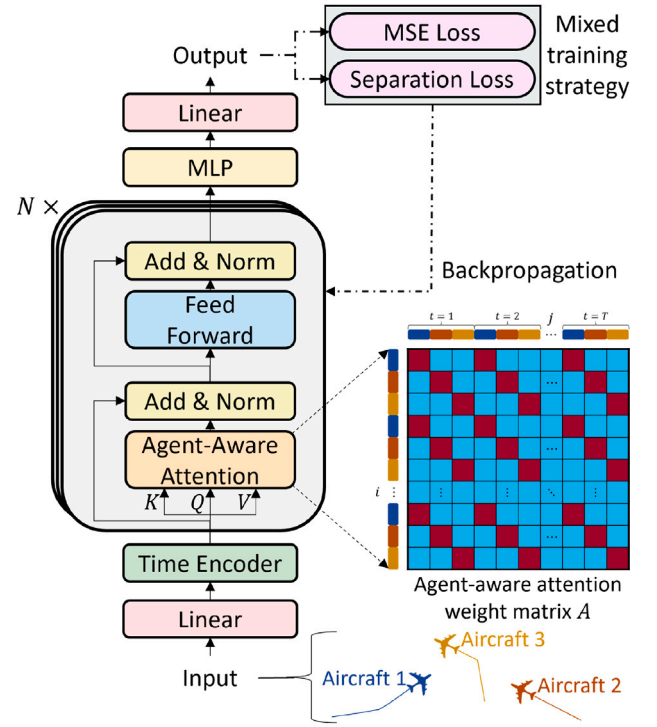


Fig. 3. The proposed neural network structure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

aspects such as speed changes, altitude fluctuations, and trajectory patterns, which can enhance ETA accuracy by providing a more detailed representation of the input sequence. When the attention function is executed in parallel, the results are concatenated and subsequently projected again, providing the final values.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_m)W^O \quad (3)$$

where $\text{head}_h = \text{Attention}(QW_h^Q, KW_h^K, VW_h^V)$, $h \in \{1, 2, \dots, m\}$, and m is the total number of heads. W^O , W_h^Q , W_h^K , W_h^V are weight matrices.

3.2. Proposed multi-agent model

To address the air traffic problem in Section 2.1, we propose to emulate ATC's decision-making process under various airspace conditions by adopting a multi-agent system based on the attention mechanism. Given air traffic situations, ATCs monitor aircraft operations and decide their landing orders by comparing neighboring aircraft based on several factors, such as priority, relative position, aircraft type, and arrival time. In this regard, imitating ATC requires a multi-agent system that can understand the air traffic situation and let each aircraft perceive itself differently from other nearby aircraft. To achieve this goal, we utilize the agent-aware attention function proposed in Yuan et al. (2021).

While the attention mechanism mentioned in the previous subsection does not have any notion of temporal and social dimensions which are important in multi-agent systems, agent-aware attention takes both dimensions into account simultaneously. Agent-aware attention is designed to maintain agent identities and recognize the properties of other agents by distinguishing between the elements of the same agent and the elements of other agents. Similar to the scaled dot-product attention, the agent-aware attention mechanism takes keys K , queries Q , and values V as input. The output of the agent-aware attention function is computed as

$$\text{AgentAwareAttention}(Q, K, V) = \text{softmax}\left(\frac{A}{\sqrt{d_k}}\right)V \quad (4)$$

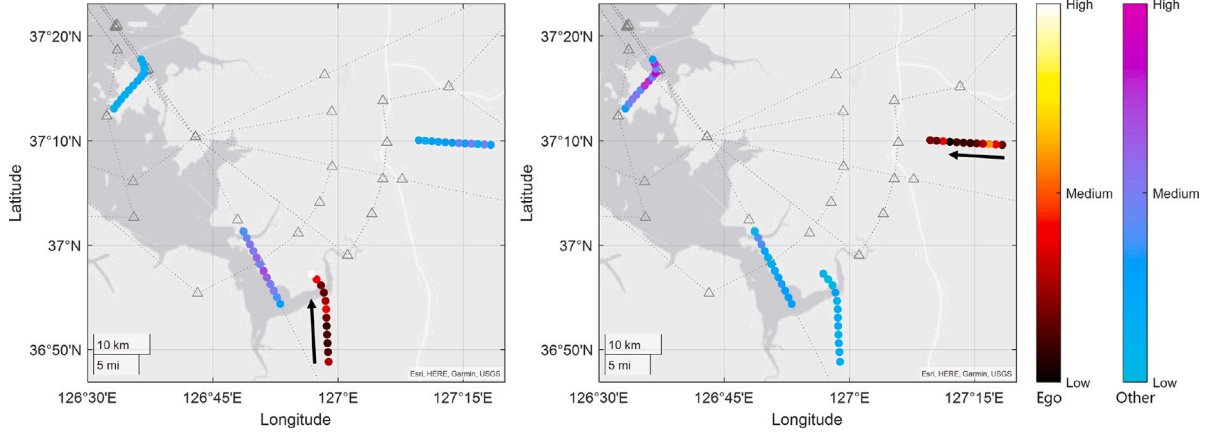


Fig. 4. Illustration of the agent-aware attention mechanism in an air traffic scene by attending to the ego agent (in red) and other agents (in sky blue) differently. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$A = M \odot (Q_{self} K_{self}^T) + (1 - M) \odot (Q_{other} K_{other}^T) \quad (5)$$

$$Q_{self} = QW_{self}^Q, \quad Q_{other} = QW_{other}^Q \quad (6)$$

$$K_{self} = KW_{self}^K, \quad K_{other} = KW_{other}^K \quad (7)$$

where M is a masking matrix in which M_{ij} equals to one if the i th query q_i and j th key k_j belong to the same agent, and M_{ij} equals to zero otherwise. In other words, this masking technique enables the computation of the attention weight matrix (A in Fig. 3) to be calculated differently depending on whether the i th query and the j th key belong to the same agent. $\odot$ denotes element-wise product and $W_{self}^Q, W_{self}^K, W_{other}^Q, W_{other}^K \in \mathbb{R}^{n \times d_k}$ are trainable parameters. n is the number of agents.

Fig. 4 illustrates how agent-aware attention works to preserve agent identities. In the traffic scene, four aircraft are heading northwest for landing and the observed trajectories are depicted as a series of dots ($t = 1, \dots, T$). As shown in Fig. 3, in the matrix A , the elements (red) whose i th query and j th key belong to the same agent are represented by different colors than those (blue sky) whose i th query and j th key belong to different agents. This indicates that the ego agent can pay attention to its own trajectory and the trajectories of other agents in another way for its prediction. On the left in Fig. 4, the ego agent with the arrow is more attending to its current track point ($t = T$) among its entire track points and more attending to the closest neighboring agent among the other agents. On the other hand, in the right plot, the ego agent in the east pays attention to the preceding aircraft during the approach, rather than the closest neighboring agent. This implies that the level of attention is closely related to the landing order in determining arrival time. Therefore, based on agent-aware attention, each agent is able to perceive its own identity and selectively attend to the more relevant information from its neighbors, capturing multi-agent interactions for the arrival sequencing problem.

For the arrival sequencing problem, the proposed multi-agent model first takes multiple trajectories in the traffic scene as an input sequence:

$$X = (x_1^1, \dots, x_1^n, \dots, x_T^1, \dots, x_T^n) \quad (8)$$

where $x_t^l \in \mathbb{R}^{d_f}$ is observed features of agent l at timestep t and d_f is the number of features. Specifically, in this paper, X is represented as multi-agent trajectories with n agents, 12 timesteps, and 8 features per timestep. Subsequently, agent-aware attention can help the proposed model practically assign each aircraft an individual arrival time by maintaining agent identities and understanding multi-agent interactions. Therefore, an output sequence is given as:

$$Y = (\hat{y}_1, \dots, \hat{y}_n) \quad (9)$$

where $\hat{y}_l$ is an ETA of aircraft l at the current timestep T .

To ensure that assigned times satisfy the appropriate separation between multiple aircraft for arrival sequencing, we propose to use the combined loss function by adding separation loss to Mean Squared Error (MSE) loss during training. The separation loss acts as a soft constraint to impose the separation requirement between leading and trailing aircraft. To train the model, we propose to use the combined loss function by adding separation loss to Mean Squared Error (MSE) loss. The separation loss acts as a soft constraint to impose the separation requirement between leading and trailing aircraft.

$$\mathcal{L}_{combined} = \omega \mathcal{L}_{MSE} + (1 - \omega) \mathcal{L}_{sep} \quad (10)$$

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{l=1}^n (y_l - \hat{y}_l)^2 \quad (11)$$

$$\mathcal{L}_{sep} = \sum_{p,q} g(\hat{y}_p, \hat{y}_q) \quad (12)$$

$$g(\hat{y}_p, \hat{y}_q) = \begin{cases} T_{sep} - |\hat{y}_p - \hat{y}_q|, & \text{if } |\hat{y}_p - \hat{y}_q| < T_{sep} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where ω is the weight between two loss functions. $y_l, \hat{y}_l$ are the actual arrival time recorded in historical data and the predicted arrival time of agent l , respectively. T_{sep} is the runway separation requirement based on Table 1 (Park and Lee, 2023). Herein, we take a mixed training strategy that utilizes the combined loss and MSE loss for better performance than sticking to a single loss function. For example, the model is trained based on the MSE loss function for the first 40 epochs and then trained after switching to the combined loss function for the remaining 60 epochs. Performance analysis over different mixed ratios of loss functions is covered in the following section.

The neural network structure of the proposed model is summarized in Fig. 3. The original Transformer (Vaswani et al., 2017) adopted the encoder-decoder architecture. In practice, a decoder that uses previously generated outputs is not essential for our task that requires a single prediction. Additionally, it is noted that there is no significant performance difference with or without a decoder. In this regard, the proposed multi-agent model in this paper adopts only a Transformer encoder with the agent-aware attention mechanism. The encoder has multiple identical layers, each composed of two sublayers: a sublayer of multi-headed agent-aware attention and a sublayer of feedforward neural network. Residual connections and layer normalization are implemented after each sublayer. By taking an input sequence, the time encoder generates the timestamped sequence of the observed trajectories to provide the timestep corresponding to each element in the given input sequence. From the time-encoded sequence of observed trajectories, the queries, keys, and values are obtained and fed into the agent-aware attention sublayer. Lastly, by taking the encoder's output, a Multi-Layer Perceptron (MLP) is used to provide a final prediction,

Table 1

Runway separation on a single runway (arrival after arrival in the same direction).

Lead/Trail	Light	Medium	Heavy	Super
Light	120	120	120	120
Medium	180	120	120	120
Heavy	180	120	120	120
Super	180	180	120	120

a list of arrival times for all agents. Additionally, an arrival sequence is then determined by sorting these ETAs in ascending order, assigning earlier ETAs to earlier landing slots.

The detailed information for the model implementation is as follows. The number (N) of identical encoder layers is 3. All queries, keys, and values have a dimensionality of 512. The feedforward layers are set to a dimension of 1024, and the hidden layers in the Multi-Layer Perceptron (MLP) use dimensions of (512, 256). The multi-headed attention employs 16 heads ($m = 16$), and the dropout rate is set as 0.1 for regularization. Finally, during the model training process, the backpropagation method calculates the gradients of the error function, and the ADAM optimizer is used to update the internal parameters of neural networks and thus minimize the loss function (Kingma and Ba, 2014).

4. Results and discussion

To test the methodology described in Section 3, this section consists of three experiments. For the first experiment, we conduct comparative experiments to demonstrate the ETA prediction performance of the proposed model compared to existing models. Second, we measure sequence similarity to evaluate how well the proposed model emulates the ATC's decision and compare it to the baseline model, Pairwise Preference Learning (PPL) (Jung et al., 2018). For the last experiment, we prepare the test dataset for arrival sequencing and use it for comparative analysis based on the actual ATC feedback.

4.1. ETA prediction

For the ETA prediction experiment, we first implement comparison models. For single-agent prediction models, Gradient Boosting Machine (GBM) is chosen as a conventional machine learning model, while Transformer is selected as a deep learning model. Based on the literature on data-driven ETA prediction (Wang et al., 2020b; Choi et al., 2023b), GBM typically outperforms other conventional models such as multiple linear regression, random forests, and k -nearest neighbors. Similarly, Transformer-based models show the best performance over other deep learning models such as long short-term memory and a generative adversarial network (Giuliari et al., 2021; Deng et al., 2024a). Therefore, we select GBM and Transformer as our comparison models over other models. GBM is a widely used ensemble learning model that combines several weak models (Friedman, 2001). The decision tree model is commonly employed as a foundational model (i.e., a weak model) in GBM. The main idea behind GBM involves the incremental inclusion of decision trees in the ensemble model, with a focus on correcting the errors made by the preceding trees. Hence, in the application of ETA prediction, GBM calculates the discrepancy between the predicted ETA and the Actual Time of Arrival (ATA) from the previous iteration and constructs a new decision tree based on this residual to enhance the accuracy of the predictions. The Transformer-based model highly relies on the attention mechanism (described in Section 3.1) which can prioritize the most relevant parts of the data sequence for accurate prediction. Lastly, for multi-agent prediction, we choose the agent-aware attention mechanism using only MSE loss as a comparison.

We perform the ETA prediction experiment using two datasets (southeast and northwest) described in Section 2.2. Given that each sample typically comprises multiple aircraft, the single-agent models,

which are capable of handling only one aircraft at a time, need to generate predictions individually for each aircraft in the sample. In contrast, the multi-agent model produces ETAs for all aircraft in the sample with a single prediction. For comparative analysis, all models are trained and tested on the same dataset split for training, validation, and testing, and the hyperparameters of the comparison models are carefully selected to ensure a fair comparison. The deep learning models using self-attention and agent-aware attention basically adopt equivalent parameters to the proposed model, such as the dimension of the model, the number of encoder layers, the number of heads, and the number of epochs for training. Subsequently, by adjusting the learning rate within the range of 10^{-6} to 10^{-3} , we evaluate whether the loss values for the three models (self-attention, agent-aware attention with MSE loss, and the proposed model) reach a specific level where further training yielded negligible performance improvement (i.e., the loss curve flattens out). This convergence indicates that the model had fully learned from the given data. Through these experiments, we identify 10^{-5} as the learning rate that leads to convergence for three neural network-based models.

However, it is noted that GBM is a conventional machine learning model with distinct hyperparameters such as the bagging fraction, feature fraction, and maximum number of leaves. In this regard, we tune these hyperparameters separately from the neural network-based models. To align the experimental setup with the previous three models, both the bagging fraction and feature fraction are set to 1, representing all available features and data are considered during training. For the maximum number of leaves, which controls the complexity of the GBM model, we compare MSE loss values by varying this parameter between 30 and 300. Based on these experiments, we set the maximum number of leaves to 250.

For the proposed model, since we employ the combined loss function and the mixed training strategy, there exist more hyperparameters to be tuned. Firstly, we conduct a sensitivity analysis by changing the weight (ω) of the MSE loss in Eq. (10). Although we select a weight of 0.5 based on the smallest loss value in this paper, it is important to note that the impact of weight selection is not significantly meaningful, especially when considering the fluctuations in loss values after convergence. In addition, we conduct a performance analysis on different mixed ratios of loss functions to determine the best ratio. The performance of the model is evaluated in terms of both the Mean Absolute Error (MAE) and the separation accuracy. The separation accuracy is calculated by dividing the number of samples in which every pair of leading and trailing aircraft in a given traffic scene satisfies the separation requirement by the total number of samples. As an example, the performance analysis result using the southeast dataset is summarized in Fig. 5. The worst performance is observed when using the combined loss function alone, and the best performance is obtained when the ratio is 3:7, which is significantly better than the only MSE loss function used. Therefore, it is found that by taking the mixed training strategy, the model can achieve better performance even if it is trained for the same 100 epochs.

The ETA prediction results are evaluated using four different metrics. In addition to the MAE and separation accuracy mentioned above, the Root Mean Squared Error (RMSE) and the average violation time are also considered. The primary difference between MAE and RMSE lies in the fact that RMSE is more sensitive to outliers and imposes greater penalties on larger errors, whereas MAE assigns equal treatment to all errors. Hence, MAE and RMSE are useful for assessing overall performance and for reducing significant errors, respectively. Secondly, while the separation accuracy only determines whether each sample violates the separation requirement, the average violation time measures how much the violation deviates from the separation minimum based on Eq. (12).

Table 2 summarizes the ETA prediction performance of the comparison and proposed models. The bold entries are used to highlight the superior performance in comparison to other models, primarily indicating the proposed model. Compared to the original model with the

Table 2
Overall performance of ETA prediction.

Model		Single-agent		Multi-agent (Agent-aware attention)	
Landing direction	Metrics	GBM	Self-attention	MSE loss only	Proposed
Southeast	MAE (s)	37.1550	34.0288	20.5228	17.7091
	RMSE (s)	57.5425	56.5080	26.4195	24.4290
	Sep. acc.	0.8101	0.8108	0.8936	0.9212
	Avg. violation (s)	41.9096	41.4721	16.4680	16.6773
Northwest	MAE (s)	50.2030	46.2987	18.0253	14.8618
	RMSE (s)	74.3045	72.2432	22.3079	18.6480
	Sep. acc.	0.7770	0.7742	0.9125	0.9373
	Avg. violation (s)	44.1212	42.1675	14.1415	13.3767

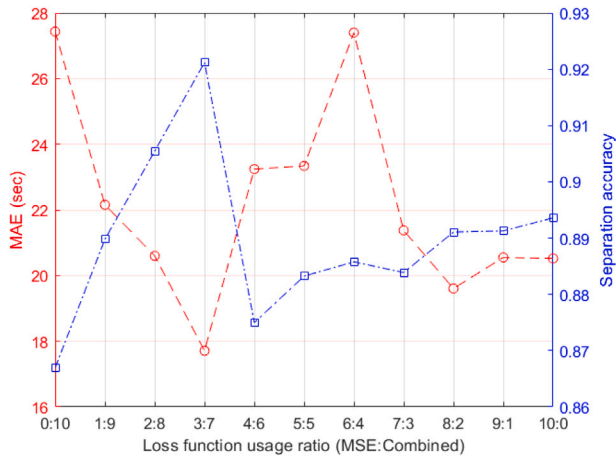


Fig. 5. Performance difference with respect to the usage of different loss functions.

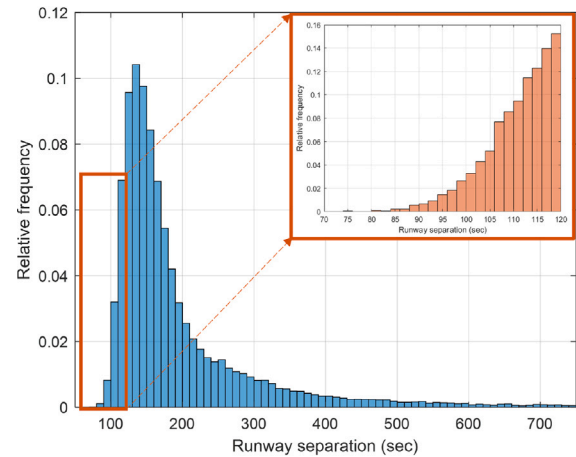


Fig. 6. Histogram of runway separation between leading and trailing aircraft on the same runway (minimum separation time: 120 s). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

MSE loss function, the proposed model employing the combined loss function and the mixed training strategy demonstrates better performance in both datasets. Furthermore, the multi-agent models perform much better than the single-agent models, which can be evidence of the single-agent model's inability to capture varying ETAs in different traffic conditions accurately. In other words, the single-agent ETA prediction models focus only on individual aircraft independently, whereas the multi-agent models consider multiple aircraft simultaneously within a given traffic scenario and try to understand their relationship. Moreover, based on the series of ETAs and separation constraints, the proposed model can closely emulate the ATC's decision made for the given traffic situation, leading to the capability to effectively handle varying ETAs caused by different traffic situations.

Although not as good as the multi-agent models, the separation accuracy of the single-agent models is higher than expected. This is attributed to the fact that, in many cases where the traffic density is not heavy, there exists a time margin between aircraft (as shown in the blue histogram in Fig. 6), and therefore prediction errors do not cause the separation to be violated. However, in terms of average violation time, significantly large errors are observed in the single-agent models, while the degree of violation time in the multi-agent models is within an acceptable range based on the actual operation results (the orange histogram in Fig. 6). It is important to note that the results are subject to errors inherent in the ADS-B system and processing.

One interesting observation is that the single-agent models perform better on the southeast data, while the multi-agent models show better performance on the northwest data. One of the main differences between the two datasets is that the average flight times for the northwest and southeast datasets are 1,437 and 1,182 s, respectively. In this regard, longer flight times in the northwest dataset can have an adverse impact on the prediction accuracy of single-agent models because there could be greater variation in ETA. However, despite longer flight times, the multi-agent models perform better in this

dataset. To understand the performance difference, it is important to understand the operational complexity in the two cases. When landing in the northbound direction, aircraft coming from up to four entry fixes simultaneously merge at the Initial Approach Fix (IAF). Conversely, in the southbound case, aircraft from three entry points (KARBU, GUKDO, and OLMEN) align first before merging with aircraft from REBIT at the IAF. Therefore, in the northwest dataset with lower operational complexity, the multi-agent models can capture traffic situations better, leading to more accurate ETA predictions. This is another evidence to show the importance of multi-agent models in ETA predictions.

The proposed model is also demonstrated with dynamically changing traffic scenarios, where the aircraft state information and the air traffic situation, including the number of aircraft operating in terminal airspace, are continuously updated. Through dynamic scenarios, it can be investigated how accurately the predicted ETAs and landing sequence are updated. Note that the landing sequence is obtained by sorting ETAs produced by the proposed model. Fig. 7 illustrates six consecutive air traffic scenes in 90-s intervals, showing the different colored aircraft with their ETA, ATA, and landing order. The predicted and actual landing orders are indicated in square brackets next to ETA and ATA, respectively.

In the first scenario, multiple aircraft from two entry fixes (REBIT and OLMEN) are continuously approaching the airport (ICN). In the first scene, since Aircraft 3 (AC#3) is closely following AC#1, the proposed model predicts the ETA of AC#3 and the landing order for the second place, which is different from the ATA and actual order. This false prediction is immediately corrected in the second scene by observing AC#3 deviates from the same path as AC#1. It seems that the ATC adjusts the landing sequence later due to certain reasons, rather than the prediction being inaccurate, and the proposed model can quickly respond to this dynamic change. Additionally, in the first scene,

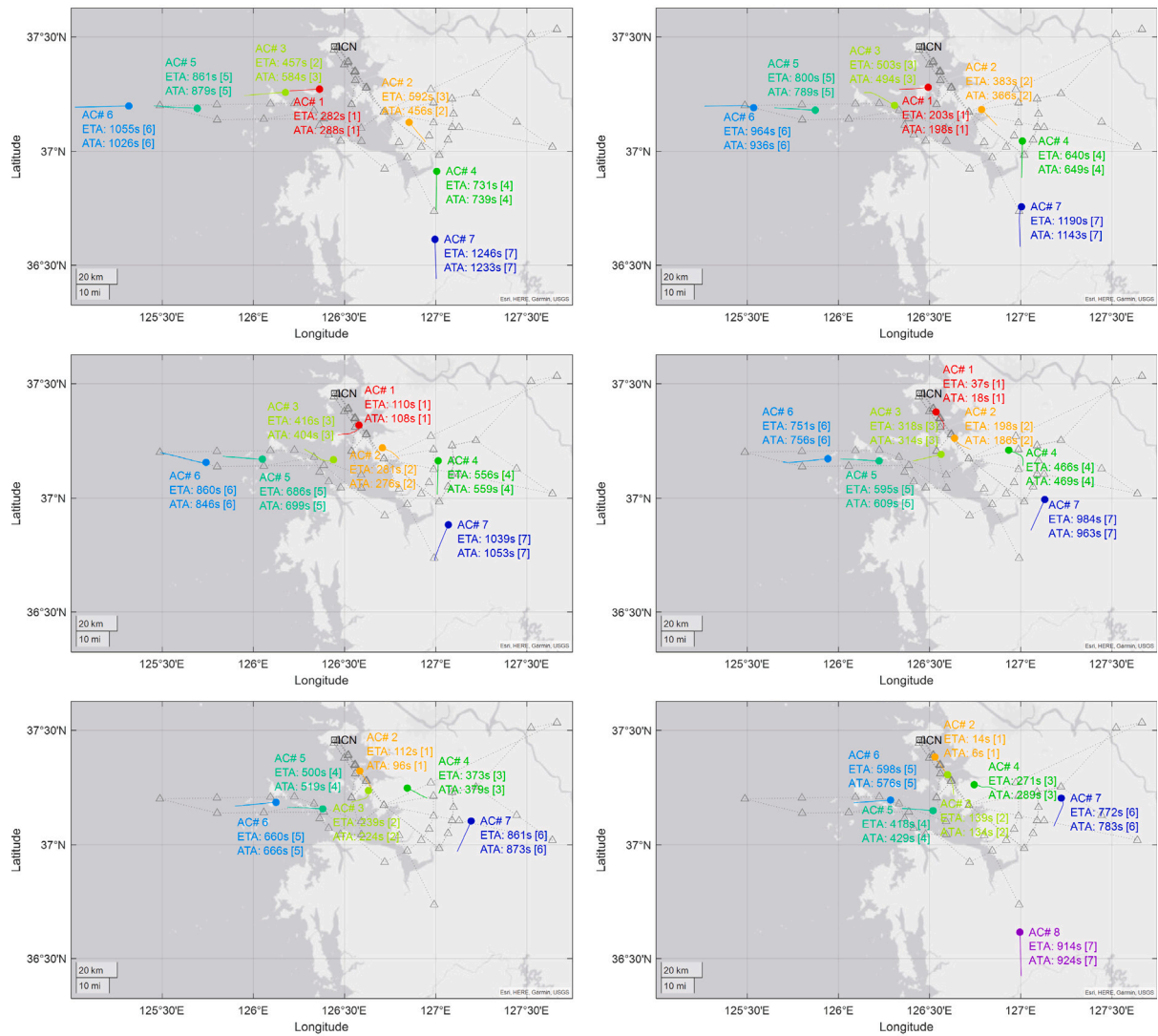


Fig. 7. Dynamic scenario 1 for ETA prediction and dynamic arrival sequencing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

AC#6 is ahead of AC#7 in the landing sequence, which likely reflects the ATC preference to prioritize aircraft from other entry fixes over aircraft from OLMEN in terms of speed and energy management (Choi and Hwang, 2024). Although AC#7 in the first scene is in a similar position to AC#8 in the sixth scene, their estimated arrival times are significantly different, determined based on different traffic conditions. We can see in the 3rd through 5th scenes that AC#7 takes a path-stretch vectoring toward the GUKDO procedure to maintain separation from AC#6. Meanwhile, AC#8 does not need to be vectored off from the assigned route.

In Fig. 8, the second scenario shows consecutive aircraft coming through GUKDO and OLMEN. AC#1 through AC#5 are already aligned in the first scene. AC#3 follows a path similar to AC#2, but their course angles differ slightly. The proposed model seems to be unable to account for this aspect, leading to a large error in the ETA prediction. Subsequently, this error is corrected in the next scene by capturing the difference in the paths of the two aircraft over time. Similarly, in the fourth scene, ETAs of AC#6 and AC#7 are predicted to violate the separation requirement, which is then adjusted based on their positional separation in the subsequent scene. This demonstrates that even if the proposed model makes some errors due to a lack of future information, it can correct the errors over time in a real-time operation. Moreover, scenes 3 to 6 depict a dynamically evolving traffic situation,

with the aircraft (AC#1 and AC#2) arriving at ICN and the aircraft (AC#9, AC#10, and AC#11) newly entering the terminal airspace. The snapshots present that the proposed model can precisely predict ETAs and the resultant landing sequence in consideration of landing aircraft removed from each scene and new incoming aircraft added to each scene. In other words, this indicates that not only real-time ETA prediction but also dynamic arrival sequencing can be done concurrently by the proposed model.

4.2. Sequence similarity

In the previous section, we show that the proposed model can provide sequencing advisories as well as ETAs in real-time operations. However, in terms of performance, we mainly focus on ETA prediction and its accuracy, but resultant landing advisories are not extensively studied. Hence, in this section, we will quantitatively evaluate how accurately the proposed model can emulate ATC's decision relative to the existing model. This quantitative analysis should be performed using metrics that can calculate the degree of similarity between the estimated arrival sequences and the actual arrival sequences (the actual decisions of the ATCs recorded in the ADS-B data). In this regard, for sequence similarity analysis, we employ two evaluation metrics: (i) the Kendall rank correlation coefficient (Kendall, 1938) and (ii) the

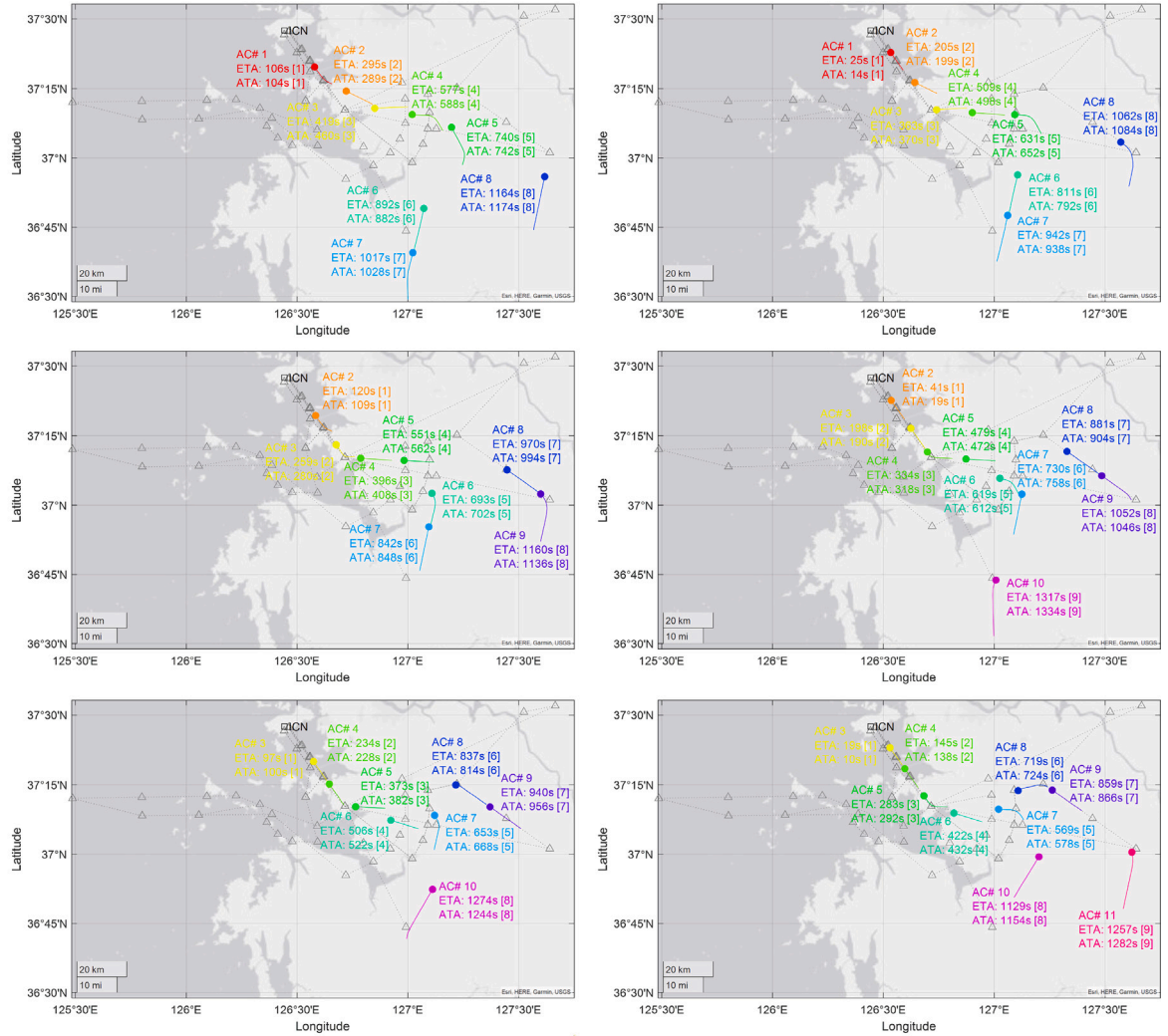


Fig. 8. Dynamic scenario 2 for ETA prediction and dynamic arrival sequencing.

Spearman's rank correlation coefficient (Spearman, 1961). Both are non-parametric rank statistics without specific assumptions about data distribution and assess the monotonic relationship between two sets of ordinal data. Additionally, these metrics are widely used to evaluate arrival sequencing (Jung et al., 2018; Du et al., 2023). The Kendall rank correlation coefficient is given as

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \quad (14)$$

where n_c and n_d correspond to the number of concordant pairs and discordant pairs, respectively. n represents the length of the landing sequence (i.e., the number of arriving aircraft). On the other hand, the Spearman's rank correlation coefficient is defined as

$$\rho = 1 - \frac{6\delta(\sigma', \sigma)}{n(n^2 - 1)} \quad (15)$$

where $\delta(\sigma', \sigma) = \sum_{r=1}^n (\sigma'(r) - \sigma(r))^2$, and σ' and σ are the predicted arrival sequence and the actual arrival sequence, respectively. Both coefficients range between -1 and 1 , and the sign and magnitude of the value are related to the direction and strength of association, respectively. For example, $+1$ indicates that two arrival sequences have exactly the same rankings.

For comparison, we select two baseline models. One is First-Come First-Served (FCFS) sequencing estimated by a single-agent ETA prediction model, and the other is the Pairwise Preference Learning (PPL) model, the foremost data-driven approach to emulate ATC preference

and decision. PPL consists of multiple pairwise preference models based on entry fixes, which are Binomial Logistic Regression (BLR) models. All pairs of aircraft in traffic situations are fed into the corresponding pairwise preference models to compute the preference probabilities. The final score of each aircraft obtained by summing all probabilities is then compared with others to determine the overall sequence of arrivals. The experiment is carried out on the same dataset described in Section 2.2. We calculate two correlation coefficients for all testing samples and illustrate their distribution in two histograms. As shown in Fig. 9, in both correlation coefficients, the distributions of the proposed model are the most right-skewed, followed by FCFS, and finally PPL. Since closer to $+1$ indicates greater similarity between two arrival sequences, PPL is the least accurate in terms of emulating ATC's decisions for arrival sequencing.

To provide a more comprehensive analysis, we examine this result more closely by categorizing it according to traffic density. In this study, we classify traffic density based on the number of aircraft, with light traffic referring to 5 or fewer aircraft, medium traffic referring to 10 or fewer aircraft, and heavy traffic referring to more than 10 aircraft. The results are summarized in Tables 3 and 4. We observe that as traffic density increases, the PPL's values of both coefficients are reduced sharply in comparison to the other two models. In order to ensure that the difference in performance does not originate from the choice of classification models or the way of training, we thoroughly train the multiple models, including GBM and Support Vector

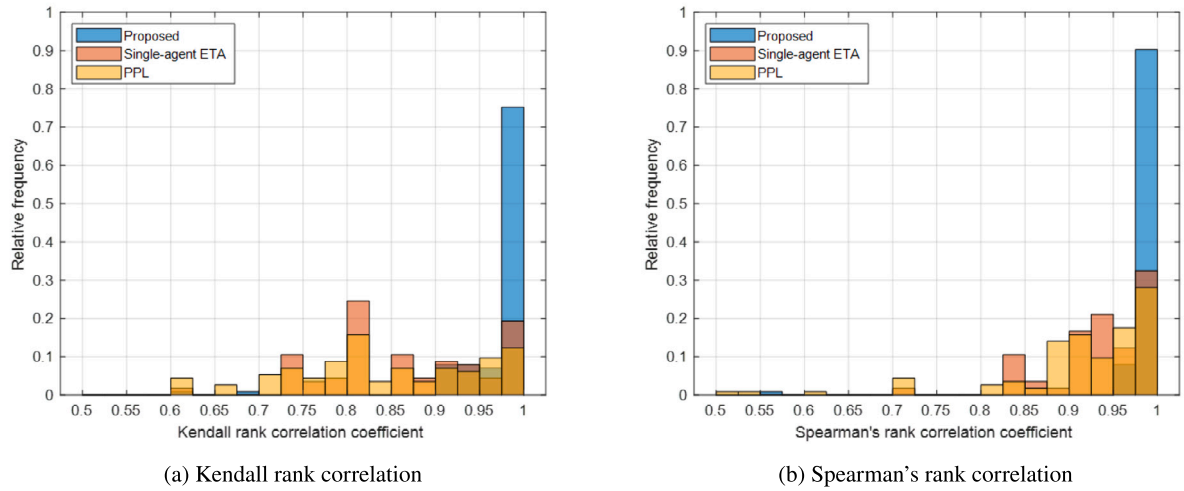


Fig. 9. Sequence similarity histograms for comparison.

Table 3

Kendall rank correlation coefficient (average).

Traffic density	PPL	FCFS	Proposed
Light	0.8790	0.8909	0.9894
Medium	0.8231	0.8601	0.9734
Heavy	0.7643	0.8324	0.9584
Overall	0.8305	0.8649	0.9760

Table 4

Spearman's rank correlation coefficient (average).

Traffic density	PPL	FCFS	Proposed
Light	0.9278	0.9394	0.9984
Medium	0.9137	0.9328	0.9840
Heavy	0.8974	0.9179	0.9686
Overall	0.9154	0.9325	0.9860

Machine (SVM), using a ten-fold cross-validation, and testing results are presented in Table 5. It is noticed that certain cases exhibit the same level of accuracy, and the average accuracy of three different models is almost identical, 97.1%. This implies that the performance of the classification models nearly reaches the Bayes error rate (Fukunaga, 2013). The Bayes error rate is the minimum theoretical error rate (i.e., fundamental limit) achievable for a given dataset due to the inherent overlap between classes in the data, which suggests that the classification models perform effectively.

Since there are no performance issues with the pairwise preference models, we need to focus on how PPL determines the arrival sequence. PPL can accommodate the actual ATC's cognitive process of making pairwise comparisons for arrival sequencing. However, the problem seems to lie in PPL's insufficient capabilities to capture dependence between decisions about the landing order of each aircraft. This means that PPL makes the sequencing decision independently based on the ranking of the scores, although the landing order of one aircraft is heavily influenced by those of other aircraft. This issue is illustrated by the following examples in Fig. 10, where a filled dot indicates the current position of each aircraft, and an asterisk indicates the future position of the aircraft in 5 min.

In Fig. 10(a), the predicted landing order of AC#2 and AC#5 is reversed compared to the actual one. It is noted that AC#4 and AC#5 fly very close together, and hence it is difficult for AC#2 to get between them based on the estimated arrival sequence. Therefore, even though AC#5 has a slightly higher score than AC#2, AC#2 can proceed with direct-to vectoring after AC#1 and AC#3, which are already vectored

off the designated routes. Note that this frequent vectoring based on area navigation (RNAV) and the point merge system is a common operation in terminal airspace (Deng et al., 2022; Choi et al., 2023a), leading to variability in aircraft travel times. In Fig. 10(b), AC#10 and AC#11 are in a similar situation to AC#2 and AC#5 in the previous scene. When comparing AC#10 and AC#11 based on PPL, AC#10 has twice the score of AC#11, supporting that AC#10 should arrive before AC#11. However, looking at the positions (marked with asterisks) 5 min later, AC#8, AC#9, and AC#11 are all aligned in a straight line. When AC#9 takes a slight path-stretch vectoring to maintain separation from AC#8, AC#11 follows closely behind AC#9. Consequently, unlike AC#2, AC#10 does not directly head toward the IAF to overtake AC#11. In this regard, the separation between the multiple leading aircraft and the resulting time slots available to the trailing aircraft should be considered for accurate sequencing predictions.

This analysis highlights the importance of the proposed model in the sense that our attention-based model with the separation constraint can address the observed limitations. This is evidenced by the two coefficients significantly greater than those of the other two models. As a side note, it is noticed that FCFS shows two coefficients greater than those of PPL at medium and high traffic densities, which means that the predicted arrival sequences by FCFS are more similar to the actual arrival sequences than those by PPL. However, this observation raises doubts in the sense that a single-agent ETA prediction model does not consider the priorities or interactions among the incoming aircraft. A possible explanation is found in the existing literature (Du et al., 2023). This study analyzes sequencing performance with respect to different ETA accuracy and claims that an improvement in prediction accuracy results in a landing sequence closer to an actual landing sequence, as well as a reduced average delay. Therefore, it is attributed to the fact that the enhanced accuracy of the ETA prediction model, which utilizes the self-attention mechanism and accommodates more features, can lead to an increased similarity to actual landing sequences.

4.3. Arrival sequencing

In this section, the proposed model is compared with the prominent optimization algorithms in terms of average delay as well as sequence similarity. For this experiment, we first prepare 50 testing samples (25 for the northwest and 25 for the southeast) labeled with a scheduled time of arrival for each aircraft. For comparison, three different algorithms are considered: (i) FCFS, (ii) Receding Horizon Control (RHC), and (iii) Constrained Position Shifting (CPS). The FCFS-based arrival sequence is obtained by sorting ETAs generated by a single-agent ETA prediction model in non-descending order, which is used as an initial

Table 5
Accuracy of pairwise preference models.

Models	KARBU-KARBU	KARBU-GUKDO	KARBU-OLMEN	KARBU-REBIT	GUKDO-GUKDO
BLR	0.9971	0.9806	0.9663	0.9531	0.9918
SVM	0.9971	0.9791	0.9652	0.9544	0.9908
GBM	0.9971	0.9817	0.9652	0.9560	0.9917
Models	GUKDO-OLMEN	GUKDO-REBIT	OLMEN-OLMEN	OLMEN-REBIT	REBIT-REBIT
BLR	0.9587	0.9469	0.9912	0.9302	0.9880
SVM	0.9587	0.9483	0.9916	0.9299	0.9880
GBM	0.9609	0.9468	0.9914	0.9285	0.9873

Table 6
Arrival sequencing performance evaluation.

Metrics	FCFS	RHC ($N_{rhc} = 2$)	Proposed	CPS ($\kappa = 3$)
Avg. avg. delays	472.9249 s	425.9315 s	430.1040 s	404.7632 s
Avg. Kendall's τ	0.8766	0.8643	0.9794	0.8364
Avg. Spearman's ρ	0.9366	0.9281	0.9902	0.9115
Response time	29.45 $\pm$ 8.986 ms	0.164 $\pm$ 0.221 s	4.094 $\pm$ 0.383 ms	1.630 $\pm$ 0.817 s

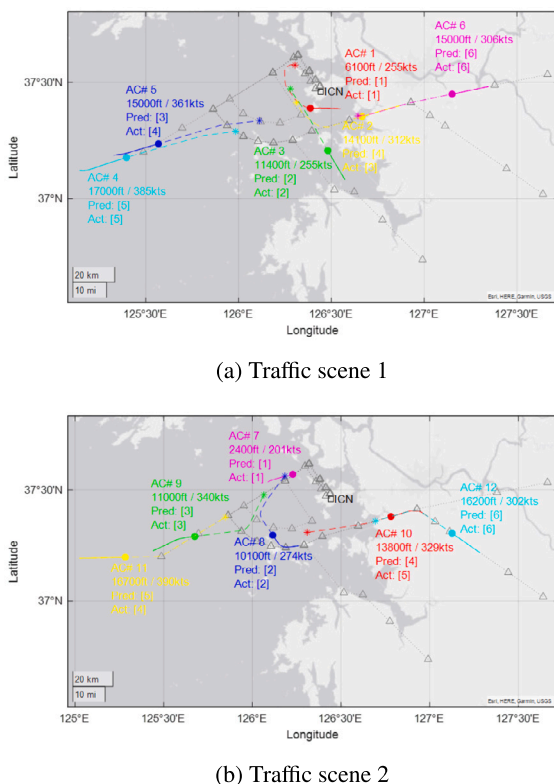


Fig. 10. Illustrative examples of false prediction generated by PPL.

arrival sequence for the following algorithms. The RHC algorithm is not only computationally efficient but also robust in dynamic and uncertain environments by iteratively optimizing the arrival sequence within the dynamic horizon (Hu and Chen, 2005). One of the key parameters in the RHC algorithm is the length of the receding horizon N_{rhc} , which determines the trade-off between performance and computational cost. CPS efficiently creates an optimized arrival sequence by allowing an aircraft to be moved up to a specified maximum number of position shifts ($\times$) from its initial arrival sequence to prevent excessive exploration of the arrival sequence (Balakrishnan and Chandran, 2006).

The four different methods are applied to the testing samples and the resulting average delay is shown as a box plot in Fig. 11. In terms of the interquartile range and median, RHC is slightly worse than CPS, whereas the proposed model and FCFS show significant differences from them. For numerical comparison, the mean of the average

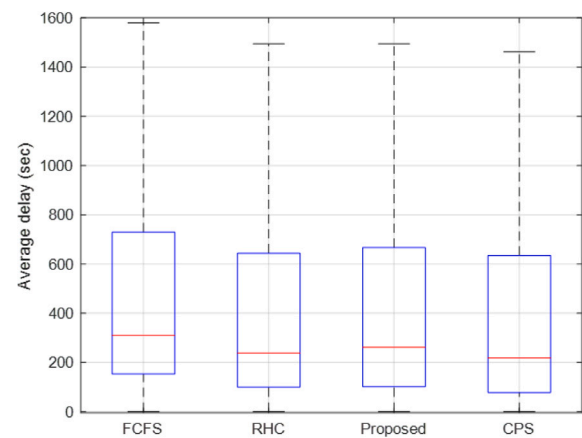


Fig. 11. Performance comparison in terms of average delay.

delays of all samples and the average of the two rank correlation coefficients for sequence similarity are calculated, and the response time is also recorded to evaluate computational efficiency. The overall results are summarized in Table 6. In three comparison algorithms, it is observed that sequence similarity decreases as the average delay is further minimized. This could indicate that excessive rearrangement of the landing sequence to optimize the objective function may be inappropriate or impractical in an aspect of actual air traffic control. On the other hand, although the delay of the proposed model increases by 0.98% compared to that of the RHC algorithm, both coefficients remain significantly higher than the others. The computation results indicate that all methods are generally applicable to real-time systems, and CPS can further reduce the computational burden by carefully tuning κ , with a slight trade-off in sequencing performance (i.e., Avg. delays). It is important to note that the response time includes both the computation of ETA predictions (for n incoming aircraft, the single-agent model iterates n times, while the multi-agent model computes just once) and arrival sequencing optimization. Therefore, it is determined that, by slightly compromising effectiveness, the proposed model can achieve a higher level of similarity and lower computational cost than the other methods.

Relatively small coefficients of other methods seem to originate from their inability to accommodate the ATC's decision process and preferences in given traffic situations. To support this point, the feedback on the samples is obtained from two actual ATCs and two ATC instructors. Participants are asked to choose one of the multiple arrival sequences without knowing which model created which arrival sequence and to provide the main considerations for their decisions. For

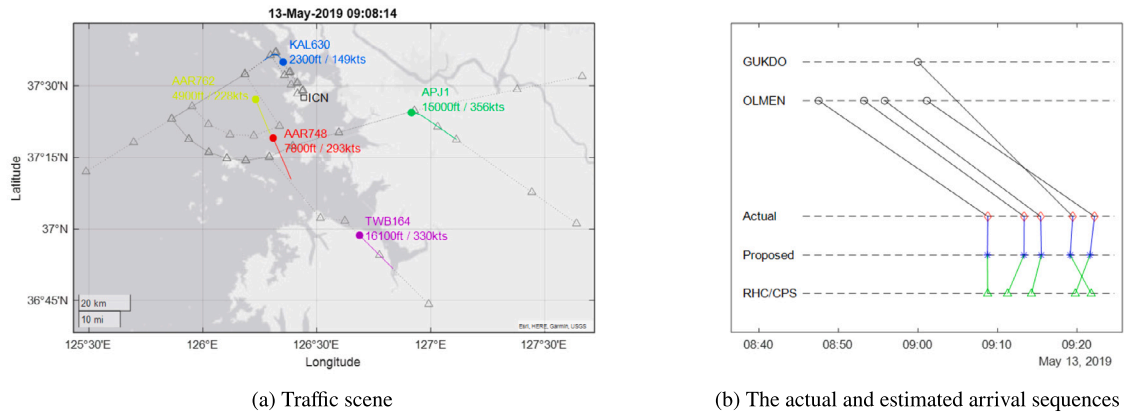


Fig. 12. Case study 1.

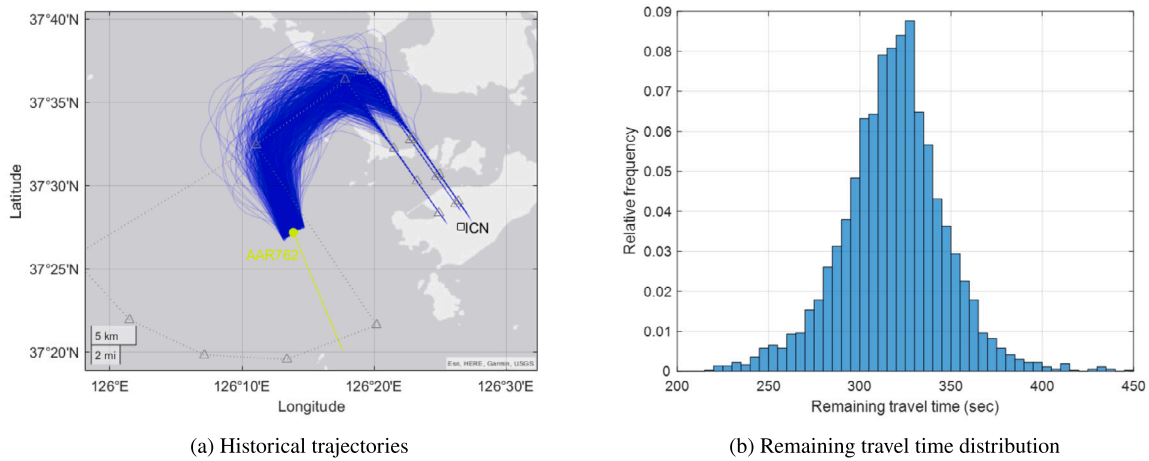


Fig. 13. Operationally infeasible assigned time for AAR762.

illustration, we first analyze two cases based on ATC feedback. Fig. 12 displays the traffic scene with five aircraft and the actual and predicted arrival sequences for the current traffic situation. Among TWB164 coming from OLMEN and APJ1 coming from GUKDO, both RHC and CPS suggest that TWB164 land first, which is different from the actual arrival sequence. However, depending on the ATC's preference and the complexity of control, the ATC can decide to vector APJ1 directly to the IAF like AAR762 and AAR748, thereby improving efficiency in air traffic management. In this case, the arrival sequence estimated by RHC and CPS is calculated to have a Kendall rank correlation coefficient of 0.8 and a Spearman's rank correlation coefficient of 0.9. Furthermore, this case can disclose potential issues that may arise when utilizing an ETA prediction model and an optimization algorithm, respectively. The optimized arrival sequence requires AAR762 to land in 3 min. However, as shown in Fig. 13, it is practically infeasible for AAR762 to land in 3 min based on the travel time distribution of historical trajectories passing through AAR762's current position. This is caused by a large prediction error in the ETA prediction model plus the predetermined time window (i.e., the earliest possible arrival times of the aircraft) allowed by the sequencing algorithms. Note that this issue is not observed in our proposed model which can do both ETA prediction and arrival sequencing simultaneously.

Similarly, in Fig. 14, seven aircraft come from OLMEN and GUKDO. When comparing KAL1402 and VJC878, KAL1402 follows the leading aircraft closely, and its speed is higher than VJC878. Furthermore, the aircraft (VJC878) coming from the south is unable to descend due to the aircraft departing the airport, while KAL1402 is able to descend first. Therefore, it is determined that KAL1402 is positioned ahead

of VJC878. When comparing KAL320 and AAR8532, VJC878 can be followed by KAL320 without any gaps along the STAR. However, if AAR8532 is required to arrive first, many control actions are required to maintain the safe separation between aircraft, increasing the workload and complexity. In this case, the arrival sequence estimated by CPS is calculated to have a Kendall rank correlation coefficient of 0.8095 and a Spearman's rank correlation coefficient of 0.9286.

Lastly, we analyze the frequency of arrival sequences chosen by ATCs and their key considerations. It is observed that the proposed model is chosen 4.19% more often than FCFS, 5.24% more often than RHC, and 28.27% more often than CPS. This difference can be interpreted as significant, considering that the models frequently share the same arrival sequence. Subsequently, all considerations for sequencing decisions in each sample are collected and summarized in Fig. 15. This pie chart reveals that the considerations related to air traffic control (ATC preference for specific traffic patterns, communication clarity, and complexity/workload of control) account for a significant portion (43%) of the total. In addition, the consideration (i.e., feasibility) to prevent impractical situations (e.g., Fig. 13) also accounts for 14%. The key considerations of ATCs are essential for effective and efficient real-world air traffic control and thus influence ATCs' decisions, which can result in a slight degradation in performance but instead lead to smoother operations. In conclusion, optimization-based algorithms cannot effectively account for ATCs' considerations, leading to a discrepancy between the actual landing sequence made by ATCs and the landing sequence optimized by the existing algorithms, while the proposed model can learn from the data and emulate the actual operation.

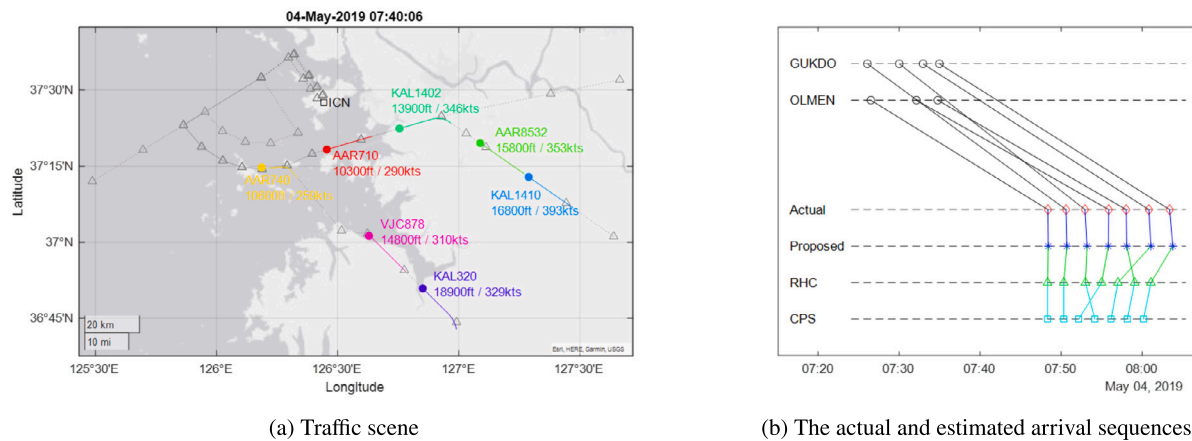


Fig. 14. Case study 2.

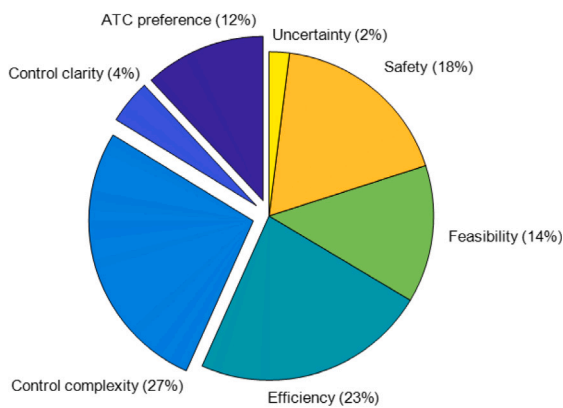


Fig. 15. Main considerations for arrival sequencing decision by ATCs.

5. Conclusion

In this paper, we proposed a multi-agent model for Estimated Time of Arrival (ETA) prediction and dynamic arrival sequencing by emulating Air Traffic Controllers (ATCs). To properly understand the interaction between the air traffic management system and human operators in complex traffic situations, we proposed an attention mechanism-based approach with a mixed training strategy that utilizes both the mean squared error loss and the separation loss functions. To demonstrate the performance of the proposed model, we conducted extensive experiments for ETA prediction and arrival sequencing with real air traffic data. The experimental results demonstrated that the proposed model can provide more accurate ETAs and more realistic landing sequences in real time than existing algorithms. The sequence similarity has been measured by two well-known rank correlation coefficients, which shows the superiority of the proposed model in emulating ATC decisions. Furthermore, the result of experiments with subject matter experts for landing sequence selection in the given traffic samples showed that ATCs choose landing sequences generated by the proposed model more frequently due to actual operation-related considerations, such as the complexity of air traffic control. The feedback from actual ATCs showed that there are some key considerations that cannot be directly incorporated into conventional optimization-based sequencing algorithms, while the proposed model can effectively account for them.

In future work, the current model will be further elaborated by incorporating meteorological information, such as wind and other weather components, and operational information, including runway occupancy and flight schedules. In addition, although departure aircraft

have different routes and designated flight levels from arrival aircraft, it could still constrain the operations of arriving aircraft or affect their landing sequence. Hence, the position and altitude of the departure flights will be considered as an additional feature to predict the landing sequence more accurately. To further improve efficiency and effectiveness, recently developed variants of Transformer (the state-of-the-art models) will be incorporated into the proposed multi-agent model. From the ATCs' feedback and analysis, it is found that there exists the ATC's preference for certain control patterns. Integrating air traffic control patterns into this study can lead to further advancement of the air traffic management system. In this regard, the identification and classification of air traffic control patterns in terminal airspace will be a worthwhile study. Lastly, the successful emulation of an air traffic controller in this study can have the potential to be extended to delay prediction and propagation where the impact of air traffic control is critical.

CRediT authorship contribution statement

Hong-Cheol Choi: Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Chuhao Deng:** Writing – review & editing. **Hyunsang Park:** Software. **Jaeyoung Ryu:** Resources. **Hak-Tae Lee:** Writing – review & editing, Funding acquisition. **Inseok Hwang:** Writing – review & editing, Supervision, Project administration.

Acknowledgments

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2020-KA158275). The authors are grateful to air traffic controllers for their valuable comments.

References

- Ayhan, S., Costas, P., Samet, H., 2018. Predicting estimated time of arrival for commercial flights. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 33–42.
- Balakrishnan, H., Chandran, B., 2006. Scheduling aircraft landings under constrained position shifting. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. p. 6320.
- Barratt, S.T., Kochenderfer, M.J., Boyd, S.P., 2018. Learning probabilistic trajectory models of aircraft in terminal airspace from position data. *IEEE Trans. Intell. Transp. Syst.* 20 (9), 3536–3545.
- Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D., 2000. Scheduling aircraft landings—the static case. *Transp. Sci.* 34 (2), 180–197.
- Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D., 2004. Displacement problem and dynamically scheduling aircraft landings. *J. Oper. Res. Soc.* 55 (1), 54–64.

- Beasley, J.E., Sonander, J., Havelock, P., 2001. Scheduling aircraft landings at London heathrow using a population heuristic. *J. Oper. Res. Soc.* 52 (5), 483–493.
- Bennell, J.A., Mesgarpour, M., Potts, C.N., 2017. Dynamic scheduling of aircraft landings. *European J. Oper. Res.* 258 (1), 315–327.
- Choi, H.-C., Deng, C., Park, H., Hwang, I., 2023a. Gaussian mixture model-based online anomaly detection for vectored area navigation arrivals. *J. Aerosp. Inf. Syst.* 20 (1), 37–52.
- Choi, H.-C., Hwang, I., 2024. Data-driven trajectory-based consensus approach to traffic management for manned and unmanned aviation. In: *AIAA SCITECH 2024 Forum*. p. 0537.
- Choi, H.-C., Park, H., Deng, C., Hwang, I., 2023b. Multi-agent aircraft estimated time of arrival prediction in terminal airspace. In: *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference. DASC, IEEE*, pp. 1–9.
- Dear, R.G., 1976. The dynamic scheduling of aircraft in the near terminal area. Cambridge, Mass.: Flight Transportation Laboratory, Massachusetts Institute
- Deng, C., Choi, H.-C., Park, H., Hwang, I., 2022. Trajectory pattern identification and classification for real-time air traffic applications in area navigation terminal airspace. *Transp. Res. Part C: Emerg. Technol.* 142, 103765.
- Deng, C., Choi, H.-C., Park, H., Hwang, I., 2024a. Multi-agent based transfer learning for data-driven air traffic applications. *arXiv preprint arXiv:2401.14421*.
- Deng, C., Choi, H.-C., Park, H., Hwang, I., 2024b. Temporal precursor discovery using long short-term memory with feature attention. *J. Aerosp. Inf. Syst.* 21 (2), 178–191.
- Deng, W., Li, K., Zhao, H., 2023. A flight arrival time prediction method based on cluster clustering-based modular with deep neural network. *IEEE Trans. Intell. Transp. Syst.*
- Du, Z., Zhang, J., Kang, B., 2023. A data-driven method for arrival sequencing and scheduling problem. *Aerospace* 10 (1), 62.
- Ernst, A.T., Krishnamoorthy, M., Storer, R.H., 1999. Heuristic and exact algorithms for scheduling aircraft landings. *Networks: An Int. J.* 34 (3), 229–241.
- Eurocontrol, 2010. AMAN status review.
- Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232.
- Fukunaga, K., 2013. *Introduction to Statistical Pattern Recognition*. Elsevier.
- Giuliani, F., Hasan, I., Cristani, M., Galasso, F., 2021. Transformer networks for trajectory forecasting. In: *2020 25th International Conference on Pattern Recognition. ICPR, IEEE*, pp. 10335–10342.
- Gui, X., Zhang, J., Peng, Z., Yang, C., 2021. Data-driven method for the prediction of estimated time of arrival. *Transp. Res. Rec.* 2675 (12), 1291–1305.
- Hancerliogullari, G., Rabadi, G., Al-Salem, A.H., Kharbeche, M., 2013. Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. *J. Air Transp. Manag.* 32, 39–48.
- Hu, X.-B., Chen, W.-H., 2005. Receding horizon control for aircraft arrival sequencing and scheduling. *IEEE Trans. Intell. Transp. Syst.* 6 (2), 189–197.
- ICAO, 2019. *ICAO annual report: The world of air transport in 2019*.
- Jung, S., Hong, S., Lee, K., 2018. A data-driven air traffic sequencing model based on pairwise preference learning. *IEEE Trans. Intell. Transp. Syst.* 20 (3), 803–816.
- Kendall, M.G., 1938. A new measure of rank correlation. *Biometrika* 30 (1/2), 81–93.
- Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Ma, Y., Du, W., Chen, J., Zhang, Y., Lv, Y., Cao, X., 2022. A spatiotemporal neural network model for estimated-time-of-arrival prediction of flights in a terminal maneuvering area. *IEEE Intell. Transp. Syst. Mag.* 15 (1), 285–299.
- Park, B.-S., Lee, H.-T., 2023. Airport surface arrival and departure scheduling using extended first-come, first-served scheduler. *Aerospace* 11 (1), 24.
- Salvatore, C., Ignaccolo, M., 2004. Genetic algorithms for solving the aircraft-sequencing problem: the introduction of departures into the dynamic model. *J. Air Transp. Manag.* 10 (5), 345–351.
- Spearman, C., 1961. The proof and measurement of association between two things..
- Tang, J., Abbass, H.A., 2014. Behavioral learning of aircraft landing sequencing using a society of probabilistic finite state machines. In: *2014 IEEE Congress on Evolutionary Computation. CEC, IEEE*, pp. 610–617.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30.
- Wang, L., Li, X., Mao, J., 2020a. Integrating ARIMA and bidirectional LSTM to predict ETA in multi-airport systems. In: *2020 Integrated Communications Navigation and Surveillance Conference. ICNS, IEEE*, 3F2–1.
- Wang, Z., Liang, M., Delahaye, D., 2020b. Automated data-driven prediction on aircraft estimated time of arrival. *J. Air Transp. Manag.* 88, 101840.
- Yuan, Y., Weng, X., Ou, Y., Kitani, K.M., 2021. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9813–9823.