Advanced Taxiing Path Guidance using Multi-Agent Reinforcement Learning for Air Traffic Management

Sungjoon Lee†, Gyu Seon Kim†, Soohyun Park‡, and Joongheon Kim†
†Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
‡Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mails: ssungjoon@korea.ac.kr, kingdom0545@korea.ac.kr, soohyun.park@sookmyung.ac.kr, joongheon@korea.ac.kr

***Abstract*—Airports are critical hubs for international travel, managing increasing passenger numbers and dynamic flight schedules. Efficient air traffic management (ATM) is essential for ensuring safety, minimizing delays, and optimizing airport capacity. Traditional methods like Dijkstra's algorithm are limited in dynamic environments due to their static nature and high computational requirements. This paper proposes using multi-agent reinforcement learning (MARL) algorithms, specifically communication network (CommNet), to address these challenges. CommNet enables information sharing among agents and coordinated actions, leading to efficient and safe aircraft routing. Our study leverages CommNet's centralized training and decentralized execution (CTDE) to demonstrate MARL flexibility, adaptability, and efficiency. Evaluated in a simulated environment modeled after Incheon International Airport (ICN), CommNet's performance is compared with other reinforcement learning (RL) algorithms lacking inter-agent communication. Results show CommNet significantly reduces aircraft delay times, optimizes taxiing energy consumption, and maintains safety standards, using approximately 10.1% less energy than independent MARL (I-MARL). These findings highlight CommNet's potential to enhance next-generation ATM systems through improved coordination and decision-making.**

***Index Terms*—Aircraft taxi routing, Multi-agent reinforcement learning (MARL), Communication network (CommNet), Centralized training and decentralized execution (CTDE), Airport scheduling, air traffic management (ATM)**

I. Introduction

A. Background and Motivation

In the contemporary world, airports serve as crucial hubs for international travel and global interaction. They facilitate the seamless crossing of national boundaries and foster connections across the globe. As the demand for air travel continues to grow dynamically, airports are tasked with accommodating an increasing number of passengers and adjusting to frequently changing flight schedules [1]. Uncertain factors such as changes in weather conditions, sudden increases in air traffic, and problems with technical defects in aircraft bring about continuous changes in the airport operating environment [2]. This necessitates the efficient management of aircraft takeoffs and landings under constantly evolving conditions. Given the critical nature of airport operations, safety remains a paramount concern. The potential for large-scale disruptions necessitates the implementation of robust safety measures to ensure the well-being of passengers, crew, and airport personnel. Efficient air traffic management systems are essential to navigate these challenges, aiming to maximize airport capacity while minimizing delays and maintaining the highest safety standards.

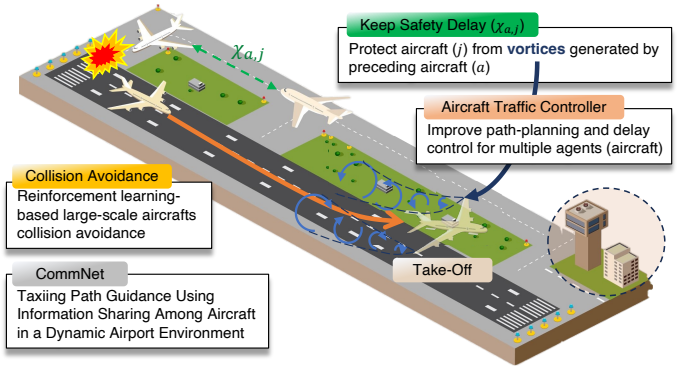

Fig. 1. RL-based airport scheduling concept for collision avoidance and energy minimization.

Air traffic management encompasses a wide range of activities aimed at ensuring the safe and efficient movement of aircraft within the airspace and on the ground, as illustrated in Fig. 1. These activities include air traffic control, airspace management, and flow control. The primary objective of air traffic management (ATM) is to maximize the utilization of airspace and airport infrastructure while minimizing delays and maintaining high safety standards. Traditionally, ATM systems have relied on deterministic algorithms like Dijkstra's algorithm to compute the shortest paths for aircraft taxiing and routing [3], [4]. However, these methods are inherently limited by their static nature and high computational requirements. The complex and dynamic nature of modern airports, characterized by frequent changes in weather conditions, varying traffic densities, and unforeseen operational disruptions, demands a more flexible and adaptive approach to ATM. The need for real-time decision-making and the ability to handle multiple aircraft simultaneously has driven the exploration of advanced techniques such as reinforcement learning (RL) and multi-agent systems [5]–[8]. The limitations of traditional ATM

Corresponding authors: Soohyun Park and Joongheon Kim
This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation); and also by IITP grant funded by MSIT (RS-2024-00439803, SW Star Lab) for Quantum AI Empowered Second-Life Platform Technology.

ISBN 978-3-903176-65-2 © 2024 IFIP

algorithms in dynamic environments have motivated the search for alternative solutions that can provide greater flexibility and adaptability. The RL, a subset of machine learning, offers a promising approach to address these challenges. Flight delays can lead to problems such as direct costs related to additional crew and aircraft time, compensation to passengers, potential fines, and the knock-on effect of delays on subsequent flights. Additionally, if an efficient ATM system is not designed, the number of aircraft that can take off and land within an hour is reduced, reducing airport capacity and causing economic losses of airports. In other words, research on efficient ATM is essential to reduce *flight delay times*, increase aircraft departure rate (ADR).

B. Algorithm Concept

By enabling agents to learn optimal policies through continuous interaction with the environment, RL algorithms can adapt to real-time changes and make decisions that enhance overall efficiency and safety. The multi-agent reinforcement learning (MARL) extends the capabilities of RL by facilitating cooperation and communication among multiple agents, thereby improving their collective decision-making abilities [9]–[11]. RL models learn from experience in a given dynamic airport environment to make optimal decisions in response to changing conditions. This is particularly relevant in ATM, where numerous aircraft must be managed simultaneously, and their interactions significantly impact the overall system performance [12]. MARL algorithms, such as communication network (CommNet), facilitate information sharing among agents and enable coordinated actions, leading to more efficient and safer aircraft routing and taxiing. The motivation behind this research is to explore the potential of MARL in enhancing ATM systems [13]. By leveraging the capabilities of CommNet, which combines centralized training with decentralized execution, this study aims to demonstrate the advantages of MARL in terms of flexibility, adaptability, and efficiency. Rather than comparing against traditional algorithms like Dijkstra's, this study focuses on the impact of information sharing among agents in MARL algorithms. We hypothesize that such information sharing will enhance environmental adaptability and ATM stability. Therefore, we implement the CommNet algorithm, which features inter-agent communication and information sharing, and compare its performance with other RL algorithms that lack these features. The primary objectives are to reduce aircraft delay times, optimize energy consumption during taxiing, and maintain safety standards by ensuring adherence to required departure intervals. This paper evaluates the performance of a MARL-based ATM system within a simulation environment modeled after Incheon International Airport (ICN). By comparing the results with those obtained using other RL algorithms that do not incorporate inter-agent communication, we aim to highlight the advantages of CommNet in terms of efficiency, stability, and adaptability. The findings are expected to contribute to the development of next-generation ATM systems capable of handling the increasing demands of modern air traffic through improved coordination and decision-making facilitated by advanced MARL techniques.

Fig. 1 shows the concept of the ATM at an airport. ATM means controlling the traffic of large numbers of aircraft at an airport. There are numerous routes for an aircraft to land at an airport and get from the runway to the gate to disembark passengers. Conversely, after passengers board the aircraft at the gate, there are numerous routes until the aircraft reaches the runway threshold for take-off. At this time, the aircraft passes numerous taxiways when going from the gate to the runway, and must reach the runway without interfering with the movement of other aircraft. The direction of aircraft movement must be determined to minimize delay time, and incorrect routing causes flight delays and reduced ADR. In conclusion, to manage traffic smoothly at the airport, aircraft must quickly navigate to their destination without colliding with other aircraft or interfering with the paths of different aircraft.

C. Contributions

- *Optimized Aircraft Departure Rate Using RL.* This study utilizes an RL technique that requires minimal computational effort and can adapt to dynamically changing environments to manage aircraft traffic and optimize the aircraft departure rate.
- *CommNet-based Centralized Critic Algorithm.* By employing a CommNet-based Centralized Critic algorithm that considers multiple agents, this research enables learning through the sharing of feature information between agents. This approach enhances the coordination and efficiency of the agents in the environment.
- *Real-World Simulation Environment Based on ICN.* To implement a realistic aviation simulation, a full-scale environment based on ICN is developed. This environment incorporates real satellite images and applies time intervals based on aircraft weight to ensure agent safety.

D. Organization

The rest of this paper is organized as follows. Sec. II introduces the preliminary information for our proposed algorithm. Sec. III elucidates the specifics of the algorithm and the formulation of RL, while also presenting a detailed description of the experimental environment. Sec. IV evaluates the performance of the proposed algorithm compared to MARL and Independent MARL. Lastly, Sec. V concludes this paper.

II. Preliminaries

This section provides the foundational context for our proposed algorithm, including a review of related works, an overview of RL and MARL basics, and specific details of our modeling for flight time intervals between landings.

A. RL and MARL Basics

1) RL and DRL: RL is a robust framework for training agents to make sequential decisions by interacting with their environment. Fundamentally, RL encompasses an agent, states, actions, and rewards. The agent observes the current state s, selects an action a, and receives a reward R based on the

resultant state, which informs its future decisions. This iterative process aims to maximize the cumulative reward over time. Dynamic programming is generally considered the optimal solution approach to this conventional RL formulation. However, dynamic programming cannot be used for problems with large-scale states due to its pseudo-polynomial computational complexity. Therefore, this conventional RL algorithm cannot solve complex and sophisticated problems with a massive number of states, such as AlphaGo [14]. To address these limitations in conventional RL, Deep Reinforcement Learning (DRL) was proposed, which reformulates RL problems using deep neural networks (DNN) [15]. In DRL, DNN's parameters are trained where the inputs and outputs are the states and actions/rewards of RL formulations. Using this DRL formulation enhances computation speed because the DNN training is done before execution, Making DRL computation primarily for DNN interference with given input states. Algorithms such as deep Q-networks (DQN) employ deep learning techniques to approximate the Q-value function, thereby enhancing the scalability and efficiency of RL in complex environments. DRL's capacity to generalize across various scenarios makes it particularly effective for applications in ATM, where dynamic and unpredictable conditions are common. In summary, while RL provides a strong foundation for sequential decision-making, its limitations in handling large-scale state spaces are overcome by DRL. By incorporating deep neural networks, DRL significantly improves the scalability and efficiency of RL, making it suitable for complex applications such as ATM

2) MARL Concepts and Algorithms: MARL addresses scenarios where multiple agents operate simultaneously in a shared environment. MARL algorithms must consider the interactions and cooperation between agents, which adds complexity compared to single-agent RL. In MARL, agents learn policies not only based on their observations but also by communicating and sharing information with other agents. MARL utilizes DRL techniques to handle the increased complexity and high-dimensional state and action spaces inherent in multi-agent environments. The CTDE framework is a prominent approach in MARL, where agents are trained centrally with access to global state information but execute their policies independently [16]. CommNet is an example of a CTDE-based algorithm, which utilizes a shared centralized neural network for multiple agents. This architecture facilitates efficient information sharing and coordination, crucial for optimizing collective behaviors in ATM.

III. Algorithm Design

A. Objectives of ATM

1) Smooth Take-Off and Landing of Aircraft: Efficient air traffic management ensures that aircraft can take off and land smoothly without conflicts. This involves coordinating movements on limited airside infrastructures, including runways, taxiways, and gate areas. An optimized ATM system reduces congestion and improves the overall throughput of the airport.

2) Minimize Aircraft Delay Time and Maximize ADR: Minimizing delays is critical for maintaining the operational efficiency of airports and airlines. Delays can lead to significant economic costs, including increased operational expenses and passenger dissatisfaction. By optimizing the scheduling and routing of aircraft, ATM systems can reduce delays, thereby increasing the ADR.

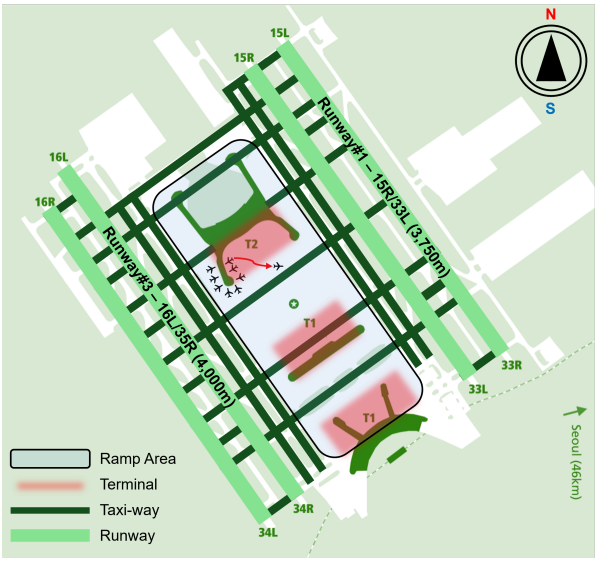

Fig. 2. Runway map of the ICN used in the performance evaluation

3) Safety Guaranteed: Safety is paramount in ATM. Inefficient management can result in collisions between aircraft or ground vehicles, leading to severe consequences. A robust ATM system ensures safe taxiing and routing, preventing accidents and maintaining operational integrity. This ensures that operational safety is upheld despite potential inaccuracies in data transmission.

B. Description of ICN's Airport Configuration

Fig. 2 is ICN's runway map adapted from our proposed MARL-based ATM system. Efficient routing within ICN's infrastructure is essential for maintaining high throughput and minimizing delays. This airport has a total of five parallel runways ($15L-33R$, $15R-33L$, $16L-34R$, $16R-34L$), and the runway sizes are indicated below the runways. Each runway is connected by several taxiways, which act as a road for the aircraft to the gate from the runway. Taxiway routes for aircraft are planned to minimize overlapping travel routes between aircraft and facilitate the flow of aircraft traffic, thereby reducing taxi routing time and potential delay time. Reduced flight delay time leads to higher ADR. An aircraft must travel from the gate to a designated runway threshold to take off and pick up passengers, or from a runway threshold to a ramp area to land and disembark passengers. In this process, the aircraft must select numerous travel routes, ensure stability without collisions, and taxi routes efficiently.

C. Departure Time Interval between Aircraft

The interval between successive takeoffs on a runway is influenced by several factors, including the size and weight of the aircraft. Larger aircraft generate stronger wake turbulence,

TABLE I
CLASSIFICATION OF AIRCRAFT BY MTOW

Manufacturer	Heavy-Body	Large-Body
Airbus	A330, A380	A320, A220
Boeing	B777, B787	B737, B757

necessitating greater separation from the following aircraft to ensure safety. In other words, if a small aircraft takes off first followed by a heavy aircraft, the required time interval between takeoffs is relatively small due to the smaller vortices [17] generated by the small aircraft. The takeoff time interval χ_{ab} between two aircraft a and b can be expressed as follows,

$$\chi_{ab} \triangleq \begin{cases} \max[\frac{\nu+\eth_{ab}}{\nu_b} - \frac{\Re}{\nu_a}, \mho_a], & \text{if } \nu_a > \nu_b \text{ (i.e., } \zeta_a > \zeta_b), \\ \max[\frac{\eth_{ab}}{\nu_b}, \mho_a], & \text{otherwise,} \end{cases} \tag{1}$$

where $\Re$, $\eth_{ab}$, ν_a, ν_b, $\mho_a$, and χ_{ab} represent the length of the final approach path, the minimum separation requirement by ATC between aircraft types a and b, the velocity of aircraft type a, b on final approach, runway occupancy time of aircraft type a, and minimum possible time interval between successive takeoffs of type a and b, respectively. The Aircraft type a takes off first, followed by aircraft type b, , and ζ_a and ζ_b denote the maximum takeoff weights (MTOW) of each aircraft. The $\mho_a$ is the runway occupancy time it takes for an aircraft to complete its takeoff roll and exit the runway. Due to the principle that only one aircraft can occupy a runway at any given time when aircraft take off in succession, aircraft type a must promptly complete its takeoff roll and exit the runway via taxiway exits to allow aircraft type b to take off, thereby enhancing the airport's capacity. If the first aircraft to take off is heavier, the formula above in (1) is used. Conversely, if the first aircraft to take off is lighter, the formula below in (1) is applied. It is generally assumed that heavier aircraft have greater velocities than lighter aircraft, which is a common assumption used in ATM. The $\eth_{ab}$ is a value that varies depending on the order and respective sizes of the aircraft. If the first aircraft a is heavy and the second aircraft b is light, this value becomes large and in other cases it becomes small. In the experiment, real aircraft from Airbus and Boeing, which are widely used at airports, are considered, and the classification of aircraft by aircraft size and weight is summarized in Table I. If the MTOW is 116 tons or more, the aircraft is classified as heavy-body; if the MTOW is between 19 and 116 tons, it is classified as large-body; and if the MTOW is 19 tons or fewer, it is classified as small-body. The small-body includes lightweight jets, turboprop aircraft, and piston-engine aircraft, and is not manufactured by Airbus or Boeing. Based on the size and weight classifications shown in Table I, and the order in which the aircraft takes off, the $\eth_{ab}$ value changes accordingly. The larger value between each calculated value and $\mho_a$ in (1) is selected as χ_{ab}. Once χ_{ab} is obtained through (1), the airport's capacity, expressed in terms of ADR can be calculated as: $\frac{3600s}{\mathbb{E}[\chi_{ab}]_s}$, and aircraft lands in order, according to the time interval χ_{ab}.

D. CommNet Algorithm

1) Information Sharing: The CommNet algorithm facilitates communication between multiple agents by sharing hidden states. This is particularly effective in a decentralized partially observable Markov decision process (Dec-POMDP) environment, where agents cannot observe the global state [18], [19]. The algorithm enhances cooperation by allowing agents to share local observations, which are then processed through a shared neural network. First of all, every Agent explores the environment consisting of state and observation in Fig. 3 Their experiences are encoded into hidden variables of the first layer θ_j^1 with encoder function in Fig. 3 as follows,

$$\theta_j^1 = \text{Encoder}(s, o_j), \tag{2}$$

where o_j represents agent j's observation value. When hidden variables are fed into the deeper hidden layers, the communication variables are entered simultaneously. The communication variable of the j-th aircraft in the i-th hidden layer is obtained by averaging the hidden variables of other aircraft, as illustrated in Fig. 3 as follows,

$$c_j^i = \frac{1}{J-1} \sum_{j' \neq j} h_{j'}^i. \tag{3}$$

At the input of every i-th layer except for the first layer, h_j^i and c_j^i are fed forwarded to the next layer using the single-agent module $f^i(\cdot)$, which produces output vector h_j^{i+1} as follows,

$$h_j^{i+1} = f^i(h_j^i, c_j^i), \tag{4}$$

where

$$f^i(\cdot) = \text{Activ}(\text{Concat}(h_j^i, c_j^i)), \tag{5}$$

where Activ$(\cdot)$ and Concat$(\cdot)$ denote a non-linear activation function (e.g., ReLU) and a concatenation function, respectively. This intercommunication among aircraft enhances information sharing, thereby facilitating robust learning even in unstable communication environments. Finally, the action probabilities for the j-th aircraft are derived by decoding the output of the final layer in Fig. 3 as follows,

$$p_{\theta_j}(A_j; o_j) = \text{Decoder}(h_j^K), \tag{6}$$

where $p(\cdot)$ and A_j represent the action distribution and the set of actions available to agent j, respectively. In summary, the sequential feed-forward process can be represented as follows,

$$p_{\theta_j}(A_j; o_j) = Q(o_j, a; [\theta_1, \cdots, \theta_J]), \tag{7}$$

where $Q(o_j, a)$ corresponds to the action-value function in a Dec-POMDP [18]. Notably, the output probabilities for the j-th aircraft are influenced by the network parameters of other aircraft.

2) CTDE-based Parameterized Policy Training: Multiple actors independently acquire experiences by exploring environments, while a centralized critic evaluates the value of the global state, as depicted in Fig. 3. The centralized critic may function as a central server, such as a control tower in fig. 1. This strategy enables all aircraft to learn near-optimal

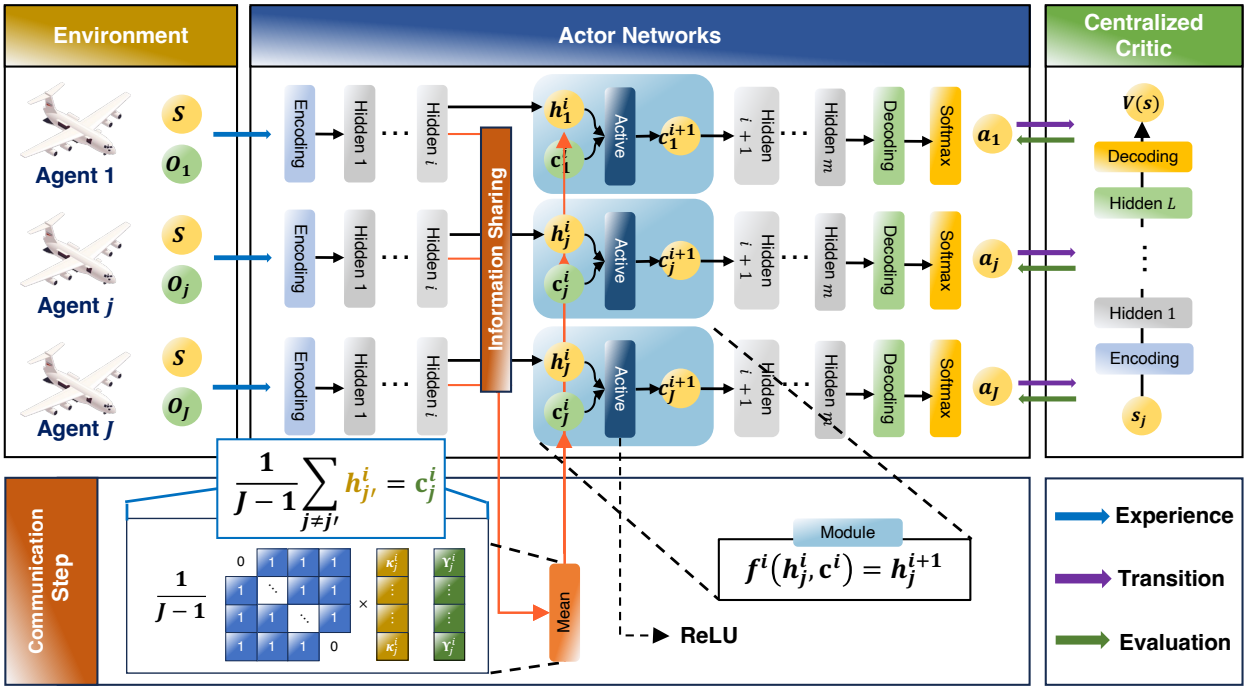

Fig. 3. Multi-Agent reinforcement learning based centralized critic CommNet network architectures with information sharing

policies symmetrically. Additionally, the multi-agent policy gradient (MAPG) method, based on the temporal difference (TD) actor-critic approach and the Bellman optimality equation, is employed to reduce high variance [20].

3) Centralized Critic: To assess the value of parameterized policies for decentralized actors, the centralized critic trains its network parameters ϕ to approximate the optimal joint state-value function, formulated as follows,

$$V_\phi(s) = \mathbb{E}_{s_t, E_t \sim \pi_\theta}\left[\sum_{u=t}^{T} \gamma^{u-t} \cdot \mathcal{R}(s^u, a^u, s^{u+1})\right], \quad (8)$$

where u is the index variable for time steps within the expectation. The centralized critic minimizes the loss function, as presented as follows,

$$\nabla_\phi \mathcal{L}(\phi) = \sum_{t=1}^{T} \nabla_\phi \left|\delta_t^\phi\right|^2, \quad (9)$$

where

$$\delta_t^\phi = \mathcal{R}(s^t, a^t, s^{t+1}) + \gamma V_\phi(s^{t+1}) - V_\phi(s^t), \quad (10)$$

where δ_t^ϕ and γ denote the TD error based on the Bellman optimality equation at time step t and the discount factor used in the reward calculation, respectively. The TD target consists of the sum of current and future reward values. The centralized critic updates its network parameters by minimizing the loss function through gradient descent, i.e.,

$$\phi^{t+1} \approx \phi^t + \alpha_{critic} \times \left[\delta_t^\phi \cdot \nabla_\phi V_\phi(s^t)\right], \quad (11)$$

where α_{critic} is the learning rate of the centralized critic network.

E. Multiple Actors

Actors, corresponding to aircraft, provide air transportation services within environments. They learn policy parameters θ to approximate an optimal policy for efficient taxi routing. At each time step t, actors make sequential decisions based on their parameterized strategy function as follows,

$$a_j^t = \arg\max_{a^t} \pi_{\theta_j}(a^t | o_j^t). \quad (12)$$

Each aircraft selects an action with the highest probability among all possible actions. The parameterized policy π_{θ_j} is defined as follows,

$$\pi_{\theta_j}(a^t | o_j^t) \triangleq \text{softmax}(p_{\theta_j}(A_j; o_j^t)), \quad (13)$$

where

$$\text{softmax}(x) \triangleq \left[\frac{e^{x_1}}{\sum_{i=1}^{N} e^{x_i}}, \cdots, \frac{e^{x_N}}{\sum_{i=1}^{N} e^{x_i}}\right]. \quad (14)$$

The objective function for dispersed actors to maximize is,

$$\nabla_{\theta_j} J(\theta_j) = \mathbb{E}_{o_j \sim \mathbb{E}}\left[\sum_{t=1}^{T} \delta_t^\phi \cdot \nabla_{\theta_j} \log \pi_{\theta_j}(a_j^t | o_j^t)\right]. \quad (15)$$

Accordingly, actors update their parameters to maximize the objective function by gradient ascent, i.e.,

$$\theta_j^{t+1} \approx \theta_j^t + \alpha_{actor} \times \left[\delta_t^\phi \cdot \nabla_{\theta_j} \log \pi_{\theta_j}(a_j^t | o_j^t)\right], \quad (16)$$

where α_{actor} is the learning rate of actor network.

F. MARL MDP for Aircraft Taxi Routing

State. The aircraft j's state is expressed as, $s_j \triangleq \{s_{j,\varphi}, s_{j,\zeta}, s_{j,\eta}, s_{j,\delta}\}$, where $s_{j,\varphi}$ represents the coordinates of the current aircraft, $s_{j,\zeta}$ is a boolean state value indicating locations where the aircraft cannot pass (areas that are not runways or taxiways, non-road areas as shown in Fig. 2), and $s_{j,\eta}$ is current step value, and $s_{j,\delta}$ is aircraft type shown in Table I, respectively. The current aircraft j's coordinates denote as, $s_{j,\varphi} \triangleq \{\mathcal{X}_{j,\varphi}, \mathcal{Y}_{j,\varphi}\}$, in 2D Cartesian coordinate system. The boolean state value indicating locations where the aircraft cannot pass are denoted as, $s_{j,\zeta} \triangleq \{\mathcal{B}_{j,Left}, \mathcal{B}_{j,Right}, \mathcal{B}_{j,Up}, \mathcal{B}_{j,Down}\}$ $s_{j,\eta}$ is the normalized state value in the range of [0, 1] up to the current step based on the maximum step. The variable $s_{j,\delta}$ represents the aircraft type associated with the agent. It is categorized as follows: Heavy is denoted by 0, Large by 1, and Small by 2.

Action. There are a total of 5 actions (a), i.e., *go straight (a_G), go back (a_B), turn left (a_L), turn right (a_R)*, and *just stay (a_S)* in a 2D grid. Then, the action space can be expressed as $a \triangleq \{a_{\mathrm{G}}, a_{\mathrm{B}}, a_{\mathrm{L}}, a_{\mathrm{R}}, a_{\mathrm{S}}\}$, where $a_{\mathrm{G}} = \{\mathcal{X}_\varphi - 1, \mathcal{Y}_\varphi\}$, $a_{\mathrm{B}} = \{\mathcal{X}_\varphi + 1, \mathcal{Y}_\varphi\}$, $a_{\mathrm{L}} = \{\mathcal{X}_\varphi, \mathcal{Y}_\varphi - 1\}$, $a_{\mathrm{R}} = \{\mathcal{X}_\varphi, \mathcal{Y}_\varphi + 1\}$, and $a_{\mathrm{S}} = \{\mathcal{X}_\varphi, \mathcal{Y}_\varphi\}$, respectively. In a given environment, the aircraft selects the optimal action from the action set a to adopt the most efficient and fastest taxi-routing path based on the reward.

Reward. The reward values are designed to encourage the agents to reach their goals efficiently while avoiding collisions and unnecessary movements. The reward is as follows,

$$R_j = R_{j,\text{step}} + R_{j,\text{goal}} + R_{j,\text{distance}} + R_{j,\text{collision}} + R_{j,\text{spacing}}, \quad (17)$$

where R_j is the reward for agent j.

- **Step Cost**: Each step taken by the agent incurs a step cost, i.e.,

$$R_{j,\text{step}} = -1. \quad (18)$$

- **Goal Achievement Reward**: If an agent reaches its goal, it receives a base reward and a bonus proportional to the remaining steps, i.e.,

$$R_{j,\text{goal}} = \begin{cases} 3000 + 0.8 \times (T_{\max} - t) & \text{if goal is reached} \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

- **Distance-Based Reward**: The reward based on the change in Manhattan distance to the goal, i.e.,

$$R_{j,\text{distance}} = \begin{cases} 200 \times (D_{j,\text{prev}} - D_{j,\text{new}}) & \text{if } D_{j,\text{new}} < D_{j,\text{prev}} \\ 100 \times (D_{j,\text{prev}} - D_{j,\text{new}}) & \text{otherwise.} \end{cases} \quad (20)$$

- **Collision Penalty**: If multiple agents occupy the same cell, each agent involved in the collision receives a penalty, i.e.,

$$R_{j,\text{collision}} = \begin{cases} -2000 & \text{if collision occurs} \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

TABLE II
ENVIRONMENTAL HYPERPARAMETERS AND SETUP

Notation	Value
Agents (Aircrafts)	4
The number of nodes in all layers	128
Discount factor	9.8×10^{-1}
Learning rate	2×10^{-4}
Batch size	256
Replay buffer size	1024
Activation function	ReLU
Optimizer	AdaDelta
Size of environment	4km (H) $\times$ 3km (W)
Max number of episodes	3000
Safe tolerance ϵ	2.0

- **Spacing Reward**: Additional reward for maintaining safe spacing time step between agents based on aircraft type, i.e.,

$$R_{j,\text{spacing}} = \begin{cases} -500 & \text{if } T_{\text{arrival},a} < \chi_{aj} \\ 500 & \text{if } |T_{\text{arrival},j} - T_{\text{arrival},a} - \chi_{aj}| \le \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

Here, $T_{\max}$, $D_{j,\text{prev}}$, $D_{j,\text{new}}$, $D_{j,\text{prev}}$, $D_{j,\text{new}}$, J, $T_{\text{arrival},a}$ and $T_{\text{arrival},j}$ are the maximum number of steps allowed in the episode, previous Manhattan distance to the goal for agent j, new Manhattan distance to the goal for agent j, total number of agents, step at which the previously arrived agent a reached its goal, step at which the current agent j reached its goal, respectively. If $T_{\text{arrival},a}$ is less than χ_{aj}, agent j receives a negative reward of -500, indicating that the agent arrived too soon after agent a. If $|T_{\text{arrival},j} - T_{\text{arrival},a} - \chi_{aj}|$ is within the tolerance ϵ, agent j receives a positive reward of 500, indicating that the agent arrived at an appropriately safe interval after agent a. Otherwise, agent j receives no additional reward or penalty for spacing. These reward values are designed to encourage the agents to find the most efficient and collision-free paths to their goals while minimizing unnecessary steps and avoiding collisions with other agents.

IV. PERFORMANCE EVALUATION

This section aims to demonstrate the value of applying MARL with inter-agent information sharing to ATM. Furthermore, MARL algorithms are effective in controlling multiple aircraft within an airport. The simulations were conducted in an environment that realistically depicts the runway map of the ICN in Fig. 2. The hyperparameters applied to the RL environment and algorithms are listed in Table II. Additionally, the number of aircraft that can depart per hour, considering safety delays, is used as a key performance indicator.

A. Reward over Training Epochs

Fig. 4 illustrates the reward convergence for each algorithm. We compare CommNet, which features information sharing among agents and a centralized critic, with the DQN-based I-MARL and Single RL algorithms. Unlike the other algorithms, which do not converge and remain unstable, CommNet quickly

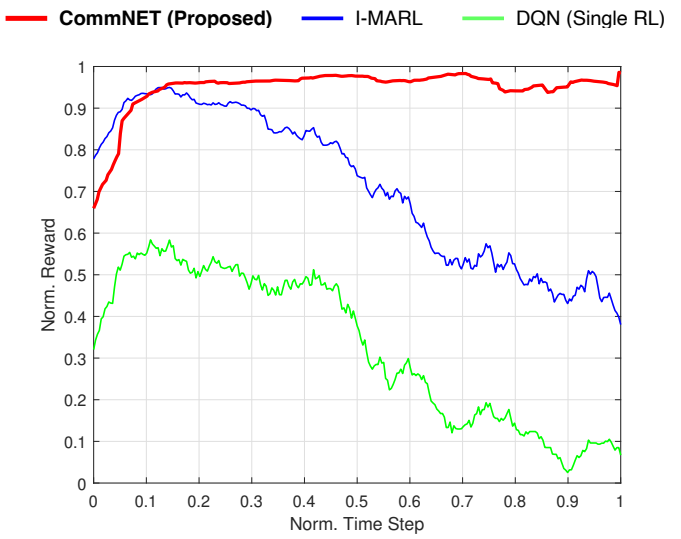

Fig. 4. Normalized rewards trend as episodes progress.

TABLE III
COMPARISON OF REWARD VALUES AND ENERGY CONSUMPTION

Algorithms	Joint Reward	Energy Consumption
CommNet (Proposed)	**0.9853**	**0.0756**
I-MARL	0.3863	0.7468
Single RL	0.0669	0.9640

converges to the optimal value, demonstrating superior performance. CommNet's stable convergence is indicative of its robustness in various operational scenarios. It effectively handles fluctuations and uncertainties in the environment, ensuring consistent performance. This robustness is critical in ATM, where unexpected changes and errors are common. Table III shows that CommNet achieves approximately 39.2% better performance in terms of optimal value compared to the I-MARL algorithm.

Fig. 5 shows each aircraft agent's reward values over normalized episodes. All agents experience a similar trend where reward values initially fluctuate but gradually stabilize as the

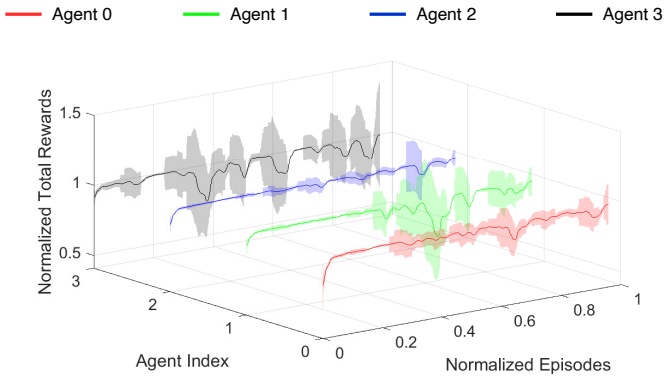

Fig. 5. Normalized reward values for each aircraft over normalized episodes.

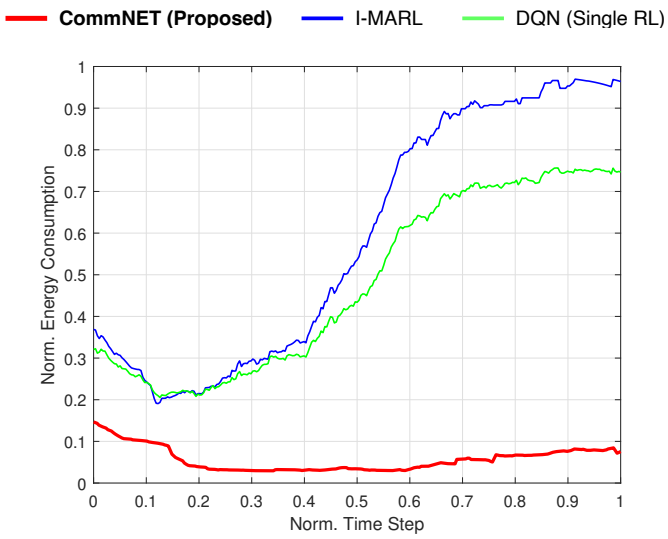

Fig. 6. The energy consumption of each aircraft as the episodes.

TABLE IV
COMPARISON OF NUMBER OF SAFETY DELAY VIOLATIONS

Algorithms	Safety Delay Violation
CommNet (Proposed)	**3**
I-MARL	140

episodes progress. Overall, the reward values suggest that the CommNet algorithm effectively enables agents to learn and optimize their taxi-routing paths, leading to improved performance and reduced energy consumption.

B. Energy Consumption

Fig. 6 shows the energy consumption of aircraft moving from their initial positions to the runway takeoff points within the simulated ICN environment. The results demonstrate that CommNet enables most agents to quickly find the optimal route from the outset, thereby utilizing less energy. In contrast, other algorithms exhibit unstable routing behavior over multiple episodes, leading to higher energy consumption as the aircraft navigates to the runway takeoff points. This highlights the efficiency of CommNet not only in terms of convergence speed but also in minimizing operational energy expenditure. Moreover, reducing energy consumption at the airport is directly linked to maintaining economic efficiency. Lower energy usage translates to reduced operational costs, which is particularly crucial for airlines operating on tight margins. By optimizing taxi-routing paths and ensuring efficient energy use, the CommNet algorithm contributes to significant cost savings and enhanced economic sustainability for airport operations. Additionally, the CommNet algorithm can achieve the optimal route using approximately 10.1% less energy compared to the I-MARL algorithm, further emphasizing its effectiveness and economic advantage in air traffic management.

C. Evaluation of Safety Departure Time Intervals

Table. IV compares the number of safety delay violations according to different algorithms. To evaluate the performance

of our trained models in adhering to safety departure time intervals between aircraft, we analyzed the departure intervals according to the departure time interval specified in equation 1. The primary objective is to ensure that the actual departure intervals meet the minimum separation requirements specified by ATC regulations. We utilized RL models trained over $3,000$ episodes using both CommNet and I-MARL to assess adherence to the MTOW for ensuring safe delay. In the given environment, the CommNet algorithm, which facilitates information sharing among agents, violated the Safe Delay requirement only three times. In contrast, the model trained with the I-MARL algorithm violated the Safe Delay requirement 140 times under the same conditions. These results highlight the superior performance of the CommNet-based approach in maintaining safe departure intervals, demonstrating its potential effectiveness for improving safety in air traffic management.

V. Concluding Remarks

This paper has demonstrated the potential of RL in significantly enhancing ATM efficiency, especially in the dynamic and complex environments characteristic of modern airports. Traditional methods like Dijkstra's algorithm, while useful in static environments, are insufficient in today's dynamic and complex airport settings, leading to increased flight delays and operational inefficiencies. In contrast, RL algorithms, particularly the MARL algorithm CommNet, collect state information from multiple aircraft in real-time airport environments. This learning-based algorithm offers the advantage of immediate applicability to any situation, considering real-time dynamics. This adaptability enhances the efficiency of ATM by reducing delay times and increasing the number of aircraft take-offs and landings. Furthermore, by optimizing taxiway path planning, this study has demonstrated a reduction in aircraft energy usage and passenger disembarkation delays, thereby ensuring economic benefits. The findings emphasize the potential of CommNet-based RL approaches in transforming ATM systems to better handle the increasing demands of modern air traffic. The superior performance of CommNet in maintaining safe departure intervals further underscores its effectiveness in improving operational safety and efficiency.

References

[1] A. Kazda, B. Badanik, and F. Serrano, "Pandemic vs. post-pandemic airport operations: Hard impact, slow recovery," *Aerospace*, vol. 9, no. 12, December 2022.

[2] M. Schultz, S. Reitmann, and S. Alam, "Classification of weather impacts on airport operations," in *Proc. Winter Simulation Conference (WSC)*, National Harbor, MD, USA, December 2019, pp. 500–511.

[3] S. Sundarraj, R. V. K. Reddy, M. B. Basam, G. H. Lokesh, F. Flammini, and R. Natarajan, "Route planning for an autonomous robotic vehicle employing a weight-controlled particle swarm-optimized Dijkstra algorithm," *IEEE Access*, vol. 11, pp. 92 433–92 442, August 2023.

[4] Y. Lu, X. Huo, and P. Tsiotras, "A beamlet-based graph structure for path planning using multiscale information," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1166–1178, March 2012.

[5] H. Ali, D.-T. Pham, S. Alam, and M. Schultz, "A deep reinforcement learning approach for airport departure metering under spatial–temporal airside interactions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 933–23 950, September 2022.

[6] P. Balakrishna, R. Ganesan, L. Sherry, and B. S. Levy, "Estimating taxi-out times with a reinforcement learning algorithm," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, October 2008, pp. 3.D.3–1–3.D.3–12.

[7] D.-T. Pham, T.-N. Tran, S. Alam, and V. N. Duong, "A generative adversarial imitation learning approach for realistic aircraft taxi-speed modeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2509–2522, October 2022.

[8] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 12 252–12 257, August 2024.

[9] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, March 2020.

[10] A. Gao, Q. Wang, W. Liang, and Z. Ding, "Game combined multi-agent reinforcement learning approach for uav assisted offloading," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12 888–12 901, December 2021.

[11] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9895–9903, 2020.

[12] H. Ali, P. D. Thinh, and S. Alam, "Deep reinforcement learning based airport departure metering," in *Proc. IEEE Int'l Intelligent Transportation Systems Conf. (ITSC)*, September 2021, pp. 366–371.

[13] C. Park, G. S. Kim, S. Park, S. Jung, and J. Kim, "Multi-agent reinforcement learning for cooperative air transportation services in city-wide autonomous urban air mobility," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 4016–4030, August 2023.

[14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, January 2016.

[15] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, December 2017.

[16] G. Chen, "A new framework for multi-agent reinforcement learning centralized training and exploration with decentralized execution via policy distillation," in *Proc. Int'l Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, Auckland, New Zealand, May 2020, p. 1801–1803.

[17] T. Gerz, F. Holzäpfel, and D. Darracq, "Commercial aircraft wake vortices," *Progress in Aerospace Sciences*, vol. 38, no. 3, pp. 181–208, April 2002.

[18] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer, "Decentralized control of partially observable Markov decision processes," in *Proc. IEEE Conf. on Decision and Control*, December 2013, pp. 2398–2405.

[19] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, and J.-H. Kim, "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-UAV control," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7086–7096, October 2022.

[20] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 32, no. 1, New Orleans, LA, USA, April 2018.