2024 IEEE International Conference on Communications Workshops (ICC Workshops) | 979-8-3503-0405-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCWORKSHOPS59551.2024.10615924

# Transformer-based Multi-agent Reinforcement Learning for Multiple Unmanned Aerial Vehicle Coordination in Air Corridors

Liangkun Yu*, Zhirun Li*, Jingjing Yao†, and Xiang Sun*
*Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA.
†Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA.

***Abstract*—Advanced Air Mobility (AAM) envisions a world where urban mobility is seamlessly integrated with the third dimension, above the ground. In AAM, different types of Unmanned Aerial Vehicle (UAVs) will be used to carry people and goods off the ground, on demand, and into the sky. To provide safe and efficient AAM services, defining the structure of air corridors is critical but has not yet been explored. This paper proposes the detailed design of air corridors and provides mathematical models of different types of air corridors. Based on the proposed air corridor models, the multi-UAV control problem in the context of air corridors is formulated to minimize the overall travel time among all UAVs, while avoiding collisions and air corridor boundary crossings. To solve the problem, the paper proposes transformer-based multi-agent reinforcement learning for multiple UAV coordination (TransRL), which incorporates a transformer to handle the dynamic dimension of each UAV's observations, and curriculum learning to improve the training efficiency. The test results show that TransRL is capable of achieving a successful arrival rate of over $90\%$ under different test settings. The code of the air corridors model and TransRL is at https://github.com/fzvincent/air-corridor.**

***Index Terms*—Reinforcement learning, transformer, air corridor**

## I. Introduction

With the increasing adoption of Unmanned Aerial Vehicles (UAVs) across various sectors and industries, NASA and the FAA have unveiled their intentions and aspirations to create an air transportation system capable of safely navigating UAVs to efficiently transport cargo and passengers. Air corridors are designated and structured 3D highways in the airspace to be used by aircraft to navigate among different vertiports. UAVs are expected to operate within these specified air corridors and adhere to suggested flight regulations, thus facilitating efficient and controlled movement of air traffic, preventing conflicts, and improving overall aviation safety. Although FAA provides very general definitions of air corridors in Class B, C, or D airspace [1], but without specifying detailed configurations, such as sizes and shapes of air corridors.

In addition, how to control a group of UAVs to efficiently fly towards their destinations via designed air corridors while avoiding collisions, remains a challenging task. One popular solution to coordinate multiple UAVs is the centralized control method, which requires UAVs to send their states (e.g., current locations and velocities) and observations (e.g., states of other observed UAVs) to the central controller to have a global view of airspace. The centralized method, however, is unscalable to large and crowded airspace, vulnerable to communication failures, and fails to avoid collisions owing to communication delay. To overcome the drawbacks, multi-agent reinforcement learning (MARL) has been applied to enable each UAV to optimize its behaviors based on its states and observations, where two neural networks, i.e., actor and critic networks, are executed in each UAV. The actor network generates the UAV's action in terms of acceleration based on its states and observations, and the critic network evaluates the action from the actor network. A common limitation hindering the application of MARL to coordinate multiple UAVs is its fixed input dimension, making it incapable of handling dynamic observation dimensions. For example, five pairs of inputs for an MARL can only handle a maximum of five pairs of observations from a UAV. If the UAV observes more than five other UAVs, some observations need to be ignored to fit into the input lines, thus jeopardizing the generated action. On the other hand, the transformer architecture has been widely used in natural language processing tasks to handle different input lengths by assigning different attention weights to different parts of the sequence. Inspired by the transformer architecture, we propose the transformer-based MARL framework to address the limitations of traditional MARL. The paper's key contributions are outlined below.

1) We propose the concrete air corridor design and provide the corresponding models to describe air corridors. These models are designed with the intention of simplifying the complexity of the subsequently proposed solution.
2) We formulate multiple UAV coordination in air corridors as an optimization problem to ensure safe and efficient UAV navigation toward their destinations. We propose transformer-based multi-agent reinforcement learning for multiple UAV coordination (TransRL) to solve the problem. TransRL incorporates a transformer to handle the dynamic dimension of each UAV's observations, and curriculum learning to improve the training efficiency.
3) Extensive simulations are conducted to demonstrate the

This work was supported by the National Science Foundation under Award under grant no. CNS-2323050 and CNS-2148178, where CNS-2148178 is supported in part by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

performance of TransRL.

## II. System model

Assume that a set of UAVs, denoted as $\mathcal{I}$, is currently in an airspace, where $i$ is used to index these UAVs. They need to traverse several air corridors to reach their destinations. The trajectory from the source to the destination for each UAV is predefined, but the real-time actor of a UAV needs to be optimized based on the control algorithm.

### A. Aerodynamic model of a UAV

Consider a 3D Cartesian coordinate system, where $\boldsymbol{c}_i(t) \in \mathbb{R}^3$ (i.e., $\boldsymbol{c}_i(t) = [c_i^x(t), c_i^y(t), c_i^z(t)]$), $\boldsymbol{v}_i(t) \in \mathbb{R}^3$, and $\boldsymbol{a}_i(t) \in \mathbb{R}^3$ represent the current position, velocity, and acceleration of UAV $i$. The controller at each UAV is able to control $\boldsymbol{a}_i(t)$ to adjust $\boldsymbol{v}_i(t)$ and $\boldsymbol{c}_i(t)$. By assuming the acceleration is consistent during each time step, the velocity and position of UAV $i$ at the beginning of the next time step is defined as

$$\boldsymbol{v}_i(t+1) = \boldsymbol{v}_i(t) + \boldsymbol{a}_i(t)\Delta t, \tag{1}$$

$$\boldsymbol{c}_i(t+1) = \boldsymbol{c}_i(t) + \frac{\boldsymbol{v}_i(t) + \boldsymbol{v}_i(t+1)}{2} \times \Delta t, \tag{2}$$

where $\Delta t$ is the duration of a time step.

### B. System models of air corridors

The airspace can be segmented into distinct parallel layers, each containing several air corridors. These air corridors fall into two primary categories: horizontal lanes and on-off ramps. Fig. 1 provides an example, where two horizontal lanes that are connected by an on-off ramp.

*1) System model of a horizontal lane:* A horizontal lane is a one-way air corridor located in a parallel layer and can be modeled as a truncated cylinder, which can be characterized by the following four parameters.

- Anchor point $\boldsymbol{b}^{cyl} \in \mathbb{R}^3$ of a truncated cylinder.
- Orientation $\boldsymbol{d}^{cyl} \in \mathbb{R}^3$ of a truncated cylinder.
- Radius $r^{cyl}$ of a truncated cylinder.
- Length $l^{cyl}$ of a truncated cylinder.

Here, we define a truncated cylinder whose $\boldsymbol{b}^{cyl} = [0, 0, 0]$ and $\boldsymbol{d}^{cyl} = [0, 0, 1]$ as the standard truncated cylinder. Other truncated cylinders can always be transformed into the standard truncated cylinder, i.e.,

$$\boldsymbol{M}_i^{cyl} \left[\boldsymbol{b}_i^{cyl}, \boldsymbol{d}_i^{cyl}\right]^T = \left[\boldsymbol{b}^{cyl}, \boldsymbol{d}^{cyl}\right]^T, \tag{3}$$

where $\boldsymbol{M}_i^{cyl}$ is the transformation matrix to convert the horizontal lane, where UAV $i$ is currently located in the global 3D coordinate system, into the standard truncated cylinder, and $\boldsymbol{b}_i^{cyl}$ and $\boldsymbol{d}_i^{cyl}$ are the anchor point and orientation of horizontal lane $i$. Hence, if $\boldsymbol{b}_i^{cyl}$ and $\boldsymbol{d}_i^{cyl}$ are known, $\boldsymbol{M}_i^{cyl}$ can be obtained based on Eq. (3). Then, we use $\boldsymbol{M}_i^{cyl}$ to convert the position, velocity, and acceleration of UAV $i$ in the global 3D coordinate system into the ones in the standard truncated cylinder, i.e., $\boldsymbol{c}_i^T := \boldsymbol{M}_i^{cyl}\boldsymbol{c}_i^T$, $\boldsymbol{v}_i^T := \boldsymbol{M}_i^{cyl}\boldsymbol{v}_i^T$, and $\boldsymbol{a}_i^T := \boldsymbol{M}_i^{cyl}\boldsymbol{a}_i^T$. *Hence, without any specification,* $\boldsymbol{c}_i$, $\boldsymbol{v}_i$, *and* $\boldsymbol{a}_i$ *are all referred to as the position, velocity, and acceleration of UAV* $i$ *in the standard truncated cylinder for the rest of the paper.* Note that the major reason for implementing this conversion is to reduce the number of parameters in describing a truncated cylinder. We can use only two scalars, i.e., $\boldsymbol{s}_i^{cyl} = [r^{cyl}, l^{cyl}]$, to describe any truncated cylinder in the global 3D coordinate system, thus reducing the complexity of the designed MARL model later on.

The following two inequalities are met if UAV $i$ is currently located within the standard truncated cylinder.

$$\begin{cases} \delta_i^{ver} \leqslant \frac{l^{cyl}}{2}, \\ \delta_i^{hor} \leqslant r^{cyl}, \end{cases} \tag{4}$$

where $\delta_i^{ver}$ and $\delta_i^{ver}$ are the vertical and horizontal distance between UAV $i$ and the anchor point of the standard truncated cylinder, respectively, which can be calculated based on

$$\begin{cases} \delta_i^{ver} = \parallel \boldsymbol{d}^{cyl} \cdot \boldsymbol{c}_i \parallel . \\ \delta_i^{hor} = \sqrt{\|\boldsymbol{c}_i\|^2 - (\delta_i^{ver})^2}. \end{cases} \tag{5}$$

*2) System model of an on-off ramp:* An on-off ramp is a one-way air corridor that connects two horizontal lanes. Any movement between two adjacent horizontal lanes must occur through the respective on-off ramp, regardless of whether these lanes are within the same parallel layer or not. To ensure a seamless shift between two horizontal lanes, an on-off ramp is modeled as two interconnected partial tori as shown in Fig. 1. The following parameters can characterize each partial torus.

- Anchor point $\boldsymbol{b}^{tor} \in \mathbb{R}^3$ of a partial torus.
- Orientation $\boldsymbol{d}^{tor} \in \mathbb{R}^3$ of a partial torus, which is the direction that is perpendicular to the direction of the partial torus (i.e., rule of thumb).
- Tube radius $r^{tor}$ of a partial torus, which is the perpendicular distance between the central path and the edge of the partial torus as shown in Fig. 1.
- Central points of the begin and end planes for a partial torus, denoted as $\boldsymbol{g}^{tor}$ and $\boldsymbol{e}^{tor}$, respectively.

We also define the standard partial torus, whose anchor point $\boldsymbol{b}^{tor} = [0, 0, 0]$, orientation $\boldsymbol{d}^{tor} = [0, 0, 1]$, and the central point of the end plane is at the y axis, i.e., $\boldsymbol{e}^{tor} = [0, R^{tor}, 0]$, where $R^{tor}$ is the distance between the anchor point and the axis of the partial torus, i.e., $R^{tor} = \parallel \boldsymbol{g}^{tor} - \boldsymbol{b}^{tor} \parallel = \parallel \boldsymbol{e}^{tor} - \boldsymbol{b}^{tor} \parallel$. Any other partial tori can always be transformed into the standard partial torus based on

$$\boldsymbol{M}_i^{tor} \left[\boldsymbol{b}_i^{tor}, \boldsymbol{d}_i^{tor}, \boldsymbol{e}_i^{tor}\right]^T = \left[\boldsymbol{b}^{tor}, \boldsymbol{d}^{tor}, \boldsymbol{e}^{tor}\right]^T, \tag{6}$$

where $\boldsymbol{M}_i^{tor}$ is the transformation matrix, and $\boldsymbol{b}_i^{tor}$, $\boldsymbol{d}_i^{tor}$, and $\boldsymbol{e}_i^{tor}$ are the anchor point, orientation, and central point of the end plane for the partial torus that UAV $i$ is located. Hence, if $\boldsymbol{b}_i^{tor}$, $\boldsymbol{d}_i^{tor}$, and $\boldsymbol{e}_i^{tor}$ are known, $\boldsymbol{M}_i^{tor}$ can be obtained based on Eq. (6), and $\boldsymbol{M}_i^{tor}$ will be used to convert $\boldsymbol{c}_i$, $\boldsymbol{v}_i$, and $\boldsymbol{a}_i$ in the global 3D coordinate system into the ones in the standard partial torus. By implementing this conversion, we can only use three scalars to describe a partial torus in the global 3D coordination, i.e., $\boldsymbol{s}_i^{tor} = [r^{tor}, R^{tor}, \mu^{tor}]$, where $\mu^{tor}$ is the

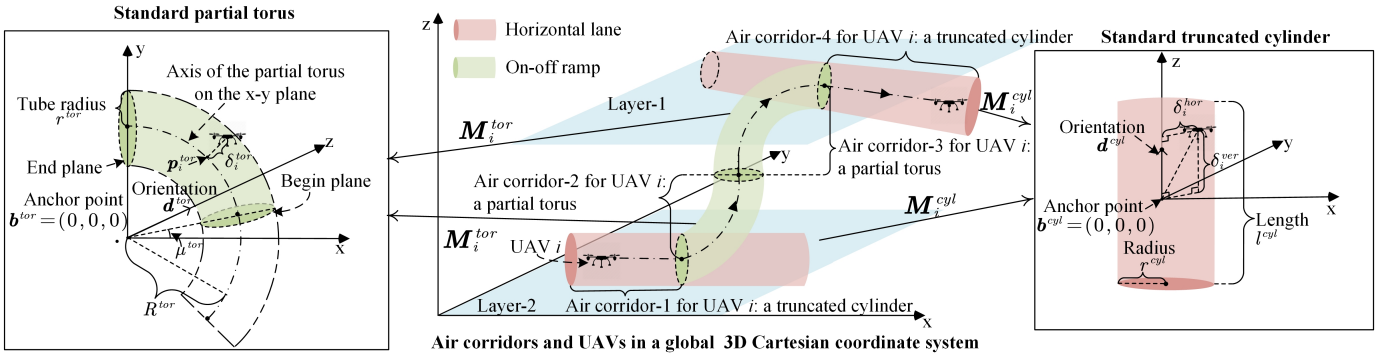


Fig. 1: Air corridor design.

angle between the x-axis and the line that connects the anchor point and end plane's central point in the standard partial torus.

The following two inequalities will be used to evaluate if UAV $i$ is in the standard partial torus.

$$\begin{cases} \| \boldsymbol{p}_i^{tor} - \boldsymbol{c}_i \| \leqslant r^{cyl}, \\ \mu^{tor} \leqslant \arctan(c_i^y / c_i^x) \leqslant \pi/2, \end{cases} \tag{7}$$

where $\arctan(c_i^y/c_i^x)$ indicates the angle between the x-axis and the line that connects UAV $i$'s location $\boldsymbol{c}_i$ and the anchor point in the x-y plane, $\boldsymbol{p}_i^{tor}$ is the perpendicular point at the axis of the partial torus for $\boldsymbol{c}_i$, and so $\| \boldsymbol{p}_i^{tor} - \boldsymbol{c}_i \|$ indicates the shortest distance between UAV $i$ and the axis of the partial torus. Here, $\boldsymbol{p}_i^{tor} = R^{tor} \times \frac{\bar{\boldsymbol{c}}_i}{\|\bar{\boldsymbol{c}}_i\|}$, where $\bar{\boldsymbol{c}}_i$ is $\boldsymbol{c}_i$ projected on the x-y plane, i.e., $\bar{\boldsymbol{c}}_i = \boldsymbol{c}_i[1,1,0]^T$.

### C. Problem formulation

Different UAVs are flying from their sources to destinations via predefined air corridors. The system is to minimize the overall travel time for all the UAVs to their destinations while avoiding collisions and ensuring UAVs are located within their air corridors. Hence, we formulate the multiple UAV coordination problem as follows.

$$\boldsymbol{P0}: \arg\min_{\boldsymbol{a}} \sum_{\forall i \in \boldsymbol{\mathcal{I}}} t_i^{travel}, \tag{8}$$

$$\text{s.t.} \quad \forall t, \forall i \in \boldsymbol{\mathcal{I}}, 0 \leq \|\boldsymbol{v}_i(t)\| \leq v^{max}, \tag{9}$$

$$\forall t, \forall i \in \boldsymbol{\mathcal{I}}, 0 \leq \|\boldsymbol{a}_i(t)\| \leq a^{max}, \tag{10}$$

$$\forall t, \forall i, i' \in \boldsymbol{\mathcal{I}}, i \neq i', \|\boldsymbol{c}_i(t) - \boldsymbol{c}_{i'}(t)\| > d^{\text{safe}}, \tag{11}$$

$$\forall t, \forall i \in \boldsymbol{\mathcal{I}}, Eq.\ (4) \text{ or } (7), \tag{12}$$

where $t_i^{travel}$ is the travel time of UAV $i$ to its destination; Eqs. (9) and (10) define the feasible velocity and acceleration of UAVs (here $v^{max}$ and $a^{max}$ are the maximum velocity and acceleration), respectively; Eq. (11) implies collision avoidance to ensure the Euclidean distance between any two UAVs less than the safe distance $d^{\text{safe}}$; Eq. (12) indicates UAV $i$ should always locate within its current air corridor. Note that whether to meet Eqs. (4) or (7) depends on the current air corridor is a truncated cylinder or partial torus.

## III. Transformer-based MARL for Multiple UAV Coordination

$\boldsymbol{P0}$ is difficult to solve by using traditional optimization methods since $t_i^{travel}$ is difficult to estimate. MARL is a machine learning method that enables each agent to train a policy by interacting with an environment modeled as a Markov Decision Process (MDP). A well-trained policy can generate optimal actions that maximize the cumulative reward for an agent. Hence, in this paper, we propose the Transformer-based Multi-agent Reinforcement Learning for Multiple UAV Coordination (TransRL) to solve $\boldsymbol{P0}$. Specifically, $\boldsymbol{P0}$ is first converted into an MDP, where each agent is to control its UAV's acceleration based on its states and observations. Here,

*1) State of UAV $i$:* $\boldsymbol{s}_i(t)$ comprises 1) self-state includes UAV $i$'s current position $\boldsymbol{c}_i(t)$ and velocity $\boldsymbol{v}_i(t)$, and 2) the characteristics of the air corridor that UAV $i$ is residing, i.e., $\boldsymbol{s}_i^{cyl}(t) = [r^{cyl}, l^{cyl}]$ or $\boldsymbol{s}_i^{tor}(t) = [r^{tor}, R^{tor}, \mu^{tor}]$, depending on the air corridor is a truncated cylinder or partial torus. Also, we add another parameter $n_i(t)$ to imply the current air corridor is the first $n = 1$, last $n = 3$, or middle $n = 2$ in its trajectory.

*2) Observations of UAV $i$:* $\boldsymbol{o}_i(t)$ comprises the positions and velocities of other UAVs observed by UAV $i$, i.e., $\boldsymbol{o}_i = \{\boldsymbol{c}_{i'}, \boldsymbol{v}_{i'} | i' \in \boldsymbol{\mathcal{I}}_i\}$, where $\boldsymbol{\mathcal{I}}_i$ is the set of UAVs observed by UAV $i$.

*3) Action of UAV $i$:* , $\boldsymbol{\alpha}_i(t)$ is its acceleration. To facilitate the calculation, we use the Spherical coordinate to define UAV $i$'s acceleration, i.e., $\boldsymbol{\alpha}_i(t) = [\rho_i(t), \theta_i(t), \phi_i(t)]$, which are the size, azimuthal angle, polar angle of UAV $i$'s acceleration, and the conversion between $\boldsymbol{a}_i(t) = [a_i^x(t), a_i^y(t), a_i^z(t)]$ in the Cartesian coordinate and $(\rho_i(t), \theta_i(t), \phi_i(t))$ in Spherical coordinate is

$$\boldsymbol{a}_i(t) = \begin{bmatrix} a_i^x(t) \\ a_i^y(t) \\ a_i^z(t) \end{bmatrix} = \begin{bmatrix} \rho_i(t)\sin(\theta_i(t))\cos(\phi_i(t)) \\ \rho_i(t)\sin(\theta_i(t))\sin(\phi_i(t)) \\ \rho_i(t)\cos(\theta_i(t)) \end{bmatrix}. \tag{13}$$

*4) Reward of UAV $i$:* $r_i(t)$ guides the agent's learning process and reflects the desired behavior. $r_i(t)$ is determined according to the following set of rules. 1) The agent receives a +160 reward if it arrives at the destination; 2) The agent

receives a +40 reward if it arrives at the end plane of its current air corridor; 3) The agent receives a -140 penalty if it breaches an air corridor's boundary; 4) The agent receives a -80 penalty if it collides with any other agent; 5) The agent receives a -0.2 penalty in each time step if it does not arrive at the destination. Having a time penalty encourages the agent to reduce its travel time [2]; 6) The agent receives a liability penalty of -10 if any other two agents have a collision. Having a liability penalty is to discourage the agent from being selfish, e.g., reducing its travel time but leading to other agents' collisions. Note that all rewards, except for the time penalty, are sparse, which poses a challenge for training efficiency [3]. TransRL will incorporate curriculum learning to address the challenge.

Existing RL solutions, such as the advantage actor critic [4], deep deterministic policy gradient [5], and the proximal policy optimization [6], can efficiently train policies to derive optimal actions. In general, these solutions comprise two neural networks, i.e., actor and critic networks, where the actor network generates the action of UAV $i$ $\boldsymbol{\alpha}_i(t)$ to maximize the cumulative reward, i.e., $\sum_t \gamma^t r_i(t)$ (where $\gamma$ is a discount factor), based on the inputs, including the states $\boldsymbol{s}_i(t)$ and observations $\boldsymbol{o}_i(t)$ of UAV $i$. The critic network also inputs $\boldsymbol{s}_i(t)$ and $\boldsymbol{o}_i(t)$, but outputs the state value $V(\boldsymbol{s}_i(t))$ to evaluate the quality of the action generated by the actor network. Two major challenges hinder the existing RL solutions to solve the proposed MDP problem. First, the dimension of UAV $i$'s observation $\boldsymbol{o}_i(t)$ may change, depending on how many other UAVs are observed by UAV $i$. Yet, the input dimension of the actor and critic networks cannot change. It is not clear how to fit the dynamic dimension of $\boldsymbol{o}_i(t)$ into the fixed dimension of the actor and critic networks. Second, as mentioned before, the rewards are sparse, which reduces learning efficiency. How to improve learning efficiency is critical.

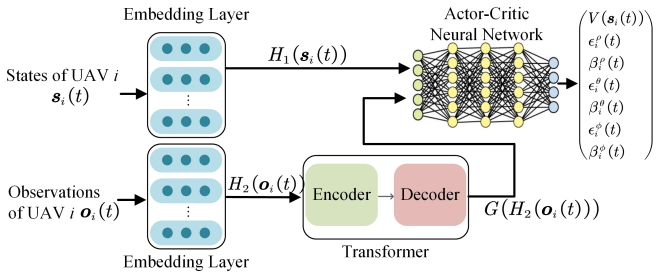


Fig. 2: The TransRL architecture.

In this paper, we design TransRL, which incorporates the transformer model and curriculum learning into the existing RL model, i.e., Proximal Policy Optimization with Generalized Advantage Estimation (PPO-GAE) [6], [7].

**Incorporating transformer into PPO-GAE**: Transformers have been widely used to process a sequence with varying length (e.g., a sentence with a varying number of words) in Natural Language Processing [8]. In general, a transformer comprises a permutation-invariant encoder and decoder, thus capable of handling an indefinite number of time-varying inputs. Hence, it is reasonable to apply the transform model to process the observations of UAV $i$. We design the structure of TransRL as shown in Fig. 2, where the states and observations of UAV $i$ are first fed into their embedding layers to normalize the input values and standardize the input dimensions. Denote $H_1(\boldsymbol{s}_i(t))$ and $H_2(\boldsymbol{o}_i(t))$ as the output of two embedding layers, where $H_1()$ and $H_2()$ are the functions achieved by the two embedding layers. $H_2(\boldsymbol{o}_i(t))$ is then fed into the transformer to extract features of the observations. Denote $G(H_2(\boldsymbol{o}_i(t)))$ as the output of the transformer, where $G()$ is the function achieved by the transformer. Note that the dimension of $G(H_2(\boldsymbol{o}_i(t)))$ is fixed and does not change by the dimension of $\boldsymbol{o}_i(t)$. Finally, $G(H_2(\boldsymbol{o}_i(t)))$ and $H_1(\boldsymbol{s}_i(t))$ are concatenated together to be the inputs of the actor and critic networks. Here, instead of creating two independent neural networks, we combine the actor and critic networks into a single neural network, which can potentially lead to more efficient training. The output of the actor and critic neural network comprises 1) the estimated state value $V(\boldsymbol{s}_i(t))$, and 2) three distributions, i.e., $B(\epsilon_i^\rho(t), \beta_i^\rho(t))$, $B(\epsilon_i^\theta(t), \beta_i^\theta(t))$, and $B(\epsilon_i^\phi(t), \beta_i^\phi(t))$, which are used to sample the actions $\rho_i(t)$, $\theta_i(t)$, and $\phi_i(t)$, respectively. Here, instead of using a Gaussian distribution, we apply a Beta distribution to allow the agent to explore different actions, where $\epsilon$ and $\beta$ in $B(\epsilon, \beta)$ are the two parameters to control the shape of the distribution. The major reason to use a Beta distribution instead of a Gaussian distribution for exploration is because a Beta distribution naturally generates values between 0 and 1. This feature is particularly useful as it directly aligns with the range of possible action values, eliminating the need for additional steps, such as clipping or normalization. Therefore, the TransRL model can be represented as

$$\begin{aligned} &V(\boldsymbol{s}_i(t)), \epsilon_i^\rho(t), \beta_i^\rho(t), \epsilon_i^\theta(t), \beta_i^\theta(t), \epsilon_i^\phi(t), \beta_i^\phi(t)] \\ &= F\left(H_1(\boldsymbol{s}_i(t)) \bigoplus G(H_2(\boldsymbol{o}_i(t)))\right), \end{aligned} \tag{14}$$

where $F()$ is the function achieved by the actor and critic neural network and $\bigoplus$ denotes vector concatenation.

**Integrating curriculum learning into PPO-GAE for efficiency training**: Curriculum learning is a training strategy that starts with simpler tasks and gradually increases task complexity, allowing the model to learn more effectively and converge to a better solution [3], [9]. The complexity of air coordination task is denoted as $\zeta$, where $\zeta = 0.1$ indicates the lowest complexity of the task. Since TransRL is trained based on a single truncated cylinder or partial torus in each episode, it is reasonable to define the task complexity based on the length of a truncated cylinder or partial torus. Specifically,

- Task complexity for training TransRL in a truncated cylinder. The length of a truncated cylinder $l$ is randomly selected in each episode, following a uniform distribution $l = U(l^{min}, l^{min} + \Delta l \times \zeta)$, where $l^{min}$ and $\Delta l$ are predefined. Hence, a larger $\zeta$ implies a longer truncated cylinder would be generated to have a higher task complexity.
- Task complexity for training TransRL in a partial torus. The angle of a partial torus $\mu^{tor}$ is randomly selected in each

episode, following a uniform distribution $\mu^{tor} = \frac{\pi}{2} \times U(0.9 - \zeta, 1 - \zeta)$, where a smaller $\mu^{tor}$ indicates a longer length of a partial torus. Hence, a larger $\zeta$ implies a wider spanning torus would be generated to have a higher task complexity.

During the curriculum learning, $\zeta$ is initialized as 0.1 and incrementally increased by $0.1$ if the probability of the UAV successfully arriving at its destination is higher than $80\%$ among 50 episodes. The curriculum learning continues until $\zeta = 1.0$. It should be noted that TransRL also applies the coordinate conversion by mapping UAVs and air corridors in the global 3D Cartesian coordinate system into the corresponding standard truncated cylinder or partial torus coordinate to reduce the dimension of the states, as mentioned in Section II, thus significantly improving the training efficiency.

## IV. Simulation Results

### A. Training Performance

In the training process, there are five UAVs in the system and their trajectories are the same, meaning that they will start at the same source locations (i.e., the same beginning plane of the first air corridor), traverse the same air corridors and end at the same destination locations (i.e., the same end plane of the last air corridor). Also, they take off simultaneously, thus having a high collision probability if they are not well-coordinated. The distance between any two UAVs' source locations is larger than the safe distance $d^{safe}$, thus ensuring that the initial states of UAVs are safe. In addition, we assume that each UAV can observe the states of the other 4 UAVs. Moreover, to test the scalability of TransRL, we train TransRL in four different scenarios, denoted as TransRL-1, TransRL-2, TransRL-3, and TransRL-4, where TransRL-$n$ indicates the TransRL model is trained based on the environment, where 5 UAVs traverse $n$ interconnected air corridors, which are randomly generated in each episode. These TransRL models will be tested in different environments to see their scalability. Finally, they will be tested traversing 4 interconnected air corridors, cylinder-torus-torus-cylinder, shown in Fig.1.

TABLE I: Training Parameters

| Parameter | Value |
|---|---|
| Minimum length of a truncated cylinder ($l^{min}$) | 5 |
| Maximum length of a truncated cylinder ($l^{min}+\Delta l$) | 20 |
| Radius of a truncated cylinder ($r^{cyl}$) | 2 |
| Maximum velocity of a UAV ($v^{max}$) | 1.5 |
| Maximum acceleration of a UAV ($a^{max}$) | 0.3 |
| Safe distance to avoid collisions ($d^{safe}$) | 0.4 |
| Tube radius of a partial torus ($r^{tor}$) | 2 |
| Distance from the anchor to the axis of a torus ($R^{tor}$) | $U(5, 10)$ |
| Maximum duration of an episode ($T^{max}$) | 1,000 time steps |
| Duration of a time step ($\Delta t$) | 1 |

During the training process, we gather 16,192 transitions and then group them into several mini-batches, each containing 1,024 transitions. The training extends across 10 epochs, alternating between the actor and critic, starting from the actor. Adam is used with learning rates to be $10^{-4}$ for the actor and $10^{-5}$ for the critic. Other training parameters are listed in Table I. Note that some numbers in Table I are relative values, and so we do not provide the corresponding units.

Fig. 3(a) shows the normalized cumulative reward for training different TransRL models over time. Since curriculum learning is used to improve training efficiency, the complexity of controlling UAVs during each episode may vary. Hence, it is reasonable to measure the normalized cumulative reward, which is calculated based on $\zeta \sum_t \gamma^t r_i(t)$. From Fig. 3(a), we can conclude that 1) normalized cumulative rewards for all models are converged, thus proving the training stability; 2) the training curve of each TransRL model exhibits ladder-type growth. For example, the normalized cumulative reward of TransRL-4's training curve (i.e., the orange curve) is dramatically increased around 2.4 $\times 10^6$ and 3.0 $\times 10^6$ time steps, indicating the complexity $\zeta$ increases from 0.1 to 0.2 and from 0.2 to 0.3, respectively; 3) TransRL-4 has higher normalized cumulative reward after the convergence than TransRL-3, TransRL-2, and TransRL-1 because UAVs in TransRL-4 need to traverse more air corridors, and each agent receives a +40 reward if it arrives at the end plane of any of these air corridors. Fig. 3(b) shows the normalized successful arrival rate for training different TransRL models over time. Here, a successful arrival means that a UAV arrives at its destination without having any collisions or breaching air corridor boundaries. The normalized successful arrival rate is also used by multiplying the successful arrival rate by complexity $\zeta$. All the curves in Fig. 3(b) reach almost 100%.

### B. Testing Performance

The setups of the testing process are very similar to those in the training process. One major difference is when generating several interconnected air corridors in the testing process, the generated partial torus is always a quarter torus (i.e., $\mu^{tor} = 0$ when the torus is mapped into the standard partial torus coordinate), and the length of the generated truncated cylinder is always the longest, i.e., $l^{tor} = l^{min} + \Delta l$, thus maximizing the task difficulties. Note that, for all the testing results, we average the values over 2,000 episodes.

TransRL is first tested with different observation dimensions. Since TransRL is trained on the basis of 5 UAVs, each UAV can observe the states of the other 4. The number of UAVs varies, while keep the number of interconnected air corridors identical to each individual training environment (i.e., there are $n$-interconnected air corridors to test the performance of TransRL-$n$). As shown in Fig. 4, all models can achieve a successful arrival rate higher than $90\%$ when the number of UAVs is no more than 7. Note, all UAVs simultaneously take off in the same plane, having the highest collision probability.

Assuming that there are 5 UAVs in the system, we then test the scalability of TransRL by changing the number of interconnected air corridors, which is different from that in the training process. As shown in Fig. 5, the successful arrival rates of TransRL-1 and TransRL-2 decrease notably when the number of interconnected air corridors is 3 and 4. The

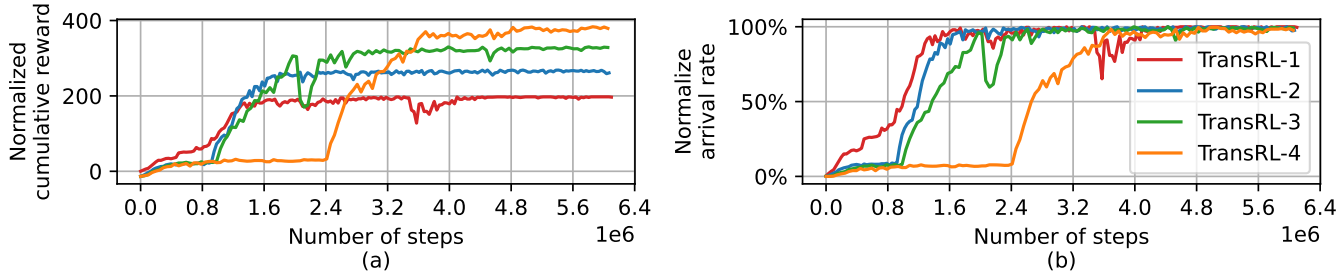


Fig. 3: Training performance over time steps, where a) normalized cumulative reward, and b) normalized arrival rate.

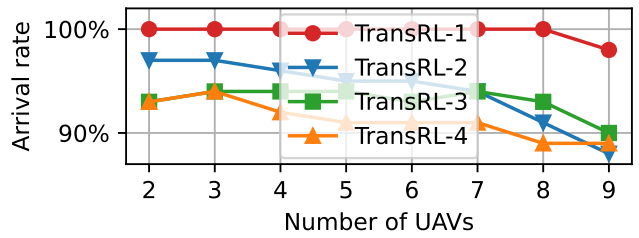


Fig. 4: Average arrival rate over the number of UAVs.

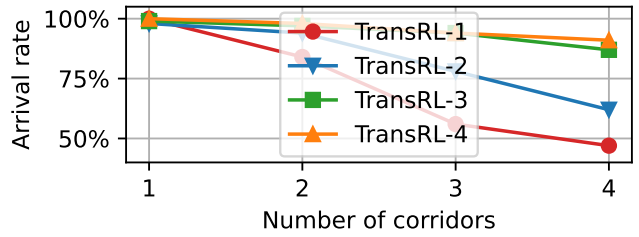


Fig. 5: Average arrival rate over the number of air corridors.

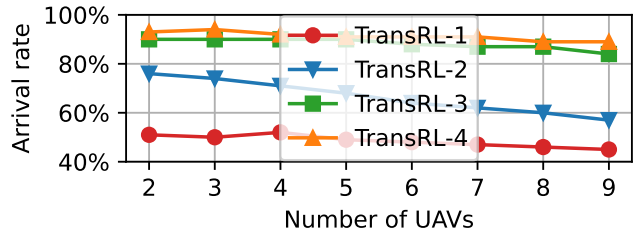


Fig. 6: TransRL performance in 4-interconnected air corridors.

reason for this performance drop has been identified, i.e., the models have poor performance in maneuvering multiple UAVs as they enter the subsequent air corridor, characterized as a partial torus. The generated acceleration cannot provide the necessary centripetal force to avoid drifting. This is evident from the calculation $\frac{(v^{max})^2}{\min(R^{tor})} = \frac{1.5^2}{5} = 0.45$, which exceeds the maximum acceleration $a^{max} = 0.3$. As a result, UAVs inevitably breach the boundaries of the partial torus. One possible solution to mitigate this issue is to input the states of the next air corridor in a UAV's trajectory to the actor-critic network such that TransRL may reduce the UAVs' acceleration and velocity before they enter the next air corridor.

The last testing is designed to change both the number of UAVs and air corridors. Specifically, we create the environment shown in Fig. 1 where there are four interconnected air corridors, starting from a truncated cylinder, quarter torus, quarter torus, and truncated cylinder. As shown in Fig. 6, TransRL-4 has the highest performance, which is reasonable as TransRL-4 is trained based on the environment with four interconnected air corridors. Also, the performance drop is not significant in TransRL-1, TransRL-3, and TransRL-4 as the number of UAVs increases, which demonstrates TransRL is capable of handling different observation dimensions.

## V. Conclusion

In this paper, we modeled and simulated the 3D air corridor environment, and formulated the multiple UAV control problem in 3D air corridors to minimize the overall travel time among all the UAVs, while avoiding collisions and boundary crossings. To solve this problem, we designed TransRL, which incorporates 1) a transformer to handle dynamic observation dimension, and 2) curriculum learning to improve the training efficiency. The test results show that TransRL-4 is capable of achieving a successful arrival rate of more than 90% in different numbers of UAVs when the number of interconnected air corridors is not greater than 4.

## References

[1] F. A. Authority, *Urban Air Mobility: Concept of operations*. Department of Transportation, 2020.

[2] A. W. Moore, "Efficient memory-based learning for robot control," University of Cambridge, Tech. Rep., 1990.

[3] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Proceedings of the 1st Annual Conference on Robot Learning*, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78, 13–15 Nov 2017, pp. 482–495.

[4] P. Yu, M. Yang, A. Xiong, Y. Ding, W. Li, X. Qiu, L. Meng, M. Kadoch, and M. Cheriet, "Intelligent-driven green resource allocation for industrial internet of things in 5g heterogeneous networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 520–530, 2022.

[5] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[7] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2022.