A Transfer Learning Approach for UAV Path Design With Connectivity Outage Constraint

Gianluca Fontanesi, Graduate Student Member, IEEE, Anding Zhu, Fellow, IEEE, Mahnaz Arvaneh, Member, IEEE, and Hamed Ahmadi, Senior Member, IEEE

Abstract—The connectivity-aware path design is crucial in the effective deployment of autonomous unmanned aerial vehicles (UAVs). Recently, reinforcement learning (RL) algorithms have become the popular approach to solving this type of complex problem, but RL algorithms suffer slow convergence. In this article, we propose a transfer learning (TL) approach, where we use a teacher policy previously trained in an old domain to boost the path learning of the agent in the new domain. As the exploration processes and the training continue, the agent refines the path design in the new domain based on the subsequent interactions with the environment. We evaluate our approach considering an old domain at sub-6 GHz and a new domain at millimeter-wave (mmWave). The teacher path policy, previously trained at the sub-6 GHz path, is the solution to a connectivity-aware path problem that we formulate as a constrained Markov decision process (CMDP). We employ a Lyapunov-based model-free deep Q-network (DQN) to solve the path design at sub-6 GHz that guarantees connectivity constraint satisfaction. We empirically demonstrate the effectiveness of our approach for different urban environment scenarios. The results demonstrate that our proposed approach is capable of reducing the training time considerably at mmWave.

Index Terms—Cellular networks, deep reinforcement learning (DRL), millimeter-wave (mmWave), path design, transfer learning (TL), unmanned aerial vehicle (UAV).

I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) are expected to be a promising solution in diverse applications, such as fast delivery, surveillance, and disaster management thanks to their easy deployment and high mobility [1]. From the standpoint of wireless communications, on the one hand, ground base stations (BSs) can be leveraged to support Unmanned aerial vehicles (UAVs) as flying user equipments (UEs) for remote operations and high data rate transmissions [2]. On the other hand, cellular networks can be used to provide a Backhaul (BH), or Fronthaul (FH) link to UAVs when deployed as wireless BSs or remote radio heads (RRHs). UAV–BSs/RRHs offer rapid deployment of on-demand communication in hotspots and provide emergency service operations [3].

A high quality and reliable ground-to-air link along the entire path [4] represents a crucial challenge for the effective deployment of UAVs in the above scenarios. An efficient UAV path design shall thus optimize the UAV path to minimize the traveling time and comply with the quality-of-connectivity constraint on the ground-to-air link. However, designing a connectivity-aware path is particularly challenging for two main reasons. First, conventional cellular networks are equipped with downtilted antennas to serve UEs on the ground. Consequently, the ground-to-air link is likely capacity limited or prone to low connectivity at specific areas or heights [5]. Second, when UAVs are deployed in unseen environments, the unavailability of knowledge about the environment increases the complexity of the path design.

A. State of the Art

Prior solutions to the UAV path optimization problem usually use conventional optimization techniques. Works in [6], [7], [8], and [9] discuss graph-search methods, whereas a dynamic programming approach is used in [10]. These approaches reformulate the corresponding nonconvex path planning optimization problems in a more tractable form that suffers from poor scalability and is based on simplified assumptions on the antenna and propagation models.

The above issues can be circumvented by exploiting detailed information on the propagation channels in a given geographical area, such as radio maps. The work in [11] utilizes a radio map of the environment to find the shortest path using graph algorithms. The radio map is assumed known as a priori. Esrafilian et al. [12] first reconstructed the radio map of the area to estimate the channel parameters. Then, the path is optimized to maximize the data collected from the ground. Similarly, in [9], [13], and [14], the UAV uses a coverage map that provides accurate locations of coverage holes in the network to maintain effective communication with the ground during the flight while moving from a starting to a final position in the shortest amount of time. $A^*$ is thus applied for finding the shortest (or approximately shortest) path in a much shorter computation time than canonical Dijkstra’s algorithm, by considering a smaller search subspace.
The above works show that the availability of radio or coverage maps makes algorithms like A* attractive for UAV path design problems. However, the assumption of full map availability is generally impractical since radio and coverage maps need to be estimated by collecting beforehand many radio measurements in a specific environment [15]. Notably, using the A* algorithm for UAV path design would require the availability of coverage maps for each UAV potential and allowed height in the area where the UAV is planned to fly. This is hard and expensive to be reached in practice.

In reinforcement learning (RL) algorithms, such as the one proposed in this article, prior knowledge of the environment like radio and coverage maps is not required. RL algorithms can learn the environment and autonomously determine the optimal path through UAV-environment interactions and only with UAV’s measured data, such as the received signal power. For this reason, applying RL algorithms in the UAV path design has received increasing interest. In [16], a Q-learning path algorithm is proposed to design the UAV path. A UAV interacts with the environment collecting positive or negative feedback. The algorithm considers the continuous and total outage during the UAV path to prevent the UAV from losing communication with the ground. However, when the size of the considered flying environment increases, Q-Table becomes too large and tabular methods such as Q-learning do not represent an efficient solution. It is thus beneficial for UAV path planning to use deep reinforcement learning (DRL) methods that combine RL with deep neural network (DNN) to address more challenging tasks. Zeng et al. [17] studied the use of the deep Q-network (DQN) to minimize the weighted sum of the UAV mission completion time and the communication outage duration. In [18], the connectivity-aware path is proposed in a similar DQN fashion, but it also includes the optimal selection of the ground BS transmitter.

One major issue of a model-free approach in UAV communication, such as DQN, is the need for a relatively high number of learning trials to converge. During the initial training, the algorithm performance is poor and improves only when enough information about the scenario environment is collected. However, this lengthy training is equivalent to thousands of flights where the reliability of the ground-to-air link is not guaranteed and it is costly, due to the UAV’s onboard battery and energy waste. Preliminary works have investigated methods to improve the learning efficiency. Using a model-based RL, the work in [17] uses the measurements collected during the flights in the training to build a radio map of the environment. The radio map is exploited to generate simulated UAV trajectories and predict their corresponding outage duration. In [19], the radio map of the environment is built in a distributed fashion using federated learning (FL) through the collaboration of multiple UAVs. The joint flight and connectivity optimization problem is then solved collectively. The work in [18] reuses past successful trajectories to imitate the same behavior and achieve faster convergence.

B. Contribution

The above-mentioned solutions contribute to reducing the algorithms’ execution time but still fail to generalize when applied to different unseen scenarios. In fact, these approaches are tailored for a single environment only or used to build a radio map of the environment [17], [18]. This affects the ability of the UAV to make good decisions when facing an unseen environment. As a consequence, the agent would require to rerun the lengthy training process for any new environment faced by the UAV.

For these reasons, we believe that, to make DRL-based solutions attractive for UAV connectivity-aware path design in real scenarios, there is a need for a framework that can significantly improve the performance in unseen environments. Motivated by this challenge, we address the UAV connectivity-aware path using a transfer learning (TL) approach. TL is the process of utilizing knowledge gained from other tasks, or prior knowledge, to benefit the target task’s learning process. The core idea of this article is to transfer the experience gained in learning to perform the proposed robust-double deep Q-network (DDQN) path design in a new domain. Our method for TL translates advice, or preferences, from a teacher path policy learned in an old domain $D_1$ at $f_1$ into a new domain $D_2$ at $f_2$. Since future wireless networks will support the sub-6 GHz and millimeter-wave (mmWave) frequency ranges [20], we believe that a different frequency band represents an interesting and practical use case of unseen environment in UAV path design. Our approach hinges on a Lyapunov method in the search for a robust teacher policy that can effectively guarantee the connectivity constraint satisfaction during training. To test our TL approach in a challenging scenario, we consider sub-6 GHz and mmWave frequency bands, which have different propagation characteristics (blockage sensitivity and scattering loss) and bandwidth availability. While other papers focused on exploiting the correlation between these two frequencies [21], we exploit that TL approaches are suitable for equivalent or different domains [22]. To demonstrate the generality of our approach, we have considered different blockage scenarios corresponding to the Urban, Dense Urban, and High Rise environments.

To better highlight the contribution of this article, Table I presents a comparison of this work with different works in the literature. A systematic search was implemented to identify the most important related works in the connectivity-aware design. The research is restricted to journal and conference papers only and keywords such as UAV, connectivity and disconnectivity constraint, path and prior knowledge. It can be noted that the connectivity outage constraint is formulated in different forms to cater for different UAV application scenarios flexibly. We propose a framework that can solve the communication-aware trajectory problem efficiently without the assumptions of coverage maps while, at the same time, representing a robust teacher policy to improve the training in new environments through TL. To the best of our knowledge, while TL is becoming a crucial topic in DRL and various domains [23], [24], this is one of the first times Teacher Advice TL is combined with a Lyapunov approach and applied to the UAV connectivity-aware path design. The TL method allows us to create and incorporate prior knowledge in our DRL solution without performing expensive measurement campaigns, speed
up the learning process, and optimally solve the optimization problem.

The contribution of this article can be listed as follows.

1) First, we formulate a 3-D UAV path problem under ground-to-air link connectivity outage constraint as constrained Markov decision process (CMDP).

2) Thus, we propose a Lyapunov approach to solve the CMDP and obtain a strategy that ensures the UAV reaches the destination while respecting the connectivity outage constraint at all times. We then develop a robust-DDQN-based algorithm to learn an optimal policy.

3) Utilizing the concepts of teacher advice and TL, we present a novel algorithm that uses the derived trained policy as a teacher policy at sub-6 GHz to guide the exploration process at mmWave and reduce the training time.

4) We first demonstrate the efficiency of the robust-DDQN comparing its performance to a benchmark conventional Dueling DDQN. At the sub-6 GHz frequency band, we show that our approach can better explore the environment and achieve higher mission success.

5) Finally, we also evaluate the proposed teacher advice and TL strategy in terms of the percentage of successful missions. Results show that using a teacher policy trained at sub-6 GHz frequency band significantly speeds up the learning process at mmWave than starting the training from scratch. Moreover, the robust-DDQN results in a better teacher policy than the state of the art Dueling DDQN.

The system model and the problem formulation are presented in Section II. In Section III, we transform the problem into a CMDP and propose a robust-DDQN-based trajectory design algorithm to play as a teacher for TL. The TL approach is presented in Section IV while the numerical results are in Section V. Finally, Section VI concludes this article.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a set, $G$, of $B$ cellular BSs providing downlink wireless service in a geographical area of interest $X \in \mathbb{R}^3$. UAVs can be deployed to reach an area of interest, $F \subseteq X$, as UAVs–UE for delivery of supplies, or UAVs as BS or RRH to provide service to a demand hotspot [7]. We assume that the path of the UAV starts from a random starting position $q_I = [x_0, y_0, h_0] \in X, \in F$, and ends in a final predetermined position $q_F = [x_F, y_F, h_F] \in F$ for a duration $T$. For the convenience of illustration, we divide the finite UAV mission completion time $T$ into a sequence of discrete time instances $t_1, t_2, \ldots, t_{\omega}$ such that $T = \omega \Delta T$ and $|t_n - t_{n-1}| \leq \Delta T$. The UAV path can be thus approximated by the sequence $\{q_n\}_{n=1}^{\omega}$ where each step point at instant $n$ is thus described by its discrete coordinates $q_n = [x_n, y_n, h_n]$. The location and the transmit power of the ground BSs can be assumed as known. We also assume that all ground BSs have equal altitude $h_BS$. Each BS and the UAV can operate at $f_1$ and $f_2$ but we assume that data transmission occurs in a single frequency band at a time.

Let $b_m = [x_m, y_m, h_m]$ be the coordinate of the $m$th ground BS in a 3-D coordinate system, the distance between the UAV and the $m$th ground BS at step $n$ is given by

$$d_{m,n} = ||q_n - b_m||, \quad m \in G.$$  

(1)

Next, we describe the channel model and formulate the problem.
TABLE II
LIST OF NOTATIONS AND SYMBOLS SUMMARY

| Notation | Description | Notation | Description |
|----------|-------------|----------|-------------|
| $\Omega$ | Area of Interest | $\Pi$ | Set of markov stationary policies |
| $G$ | The set of GBSs | $C_{\pi}$ | Expected cumulative cost function |
| $V_{\text{max}}$ | Maximum UAV speed | $D_{\pi} (\cdot)$ | Expected cumulative constraint function |
| $\Omega$ | Path duration | $\pi_B$ | Baseline policy |
| $N$ | Maximum UAV mission duration | $L$ | Lyapunov function |
| $q_L$ | Path starting location | $T_{\pi, \lambda}$ | Bellman operator |
| $q_F$ | Path final location | $\epsilon, \delta$ | Auxiliary constraint |
| $q_n$ | UAV Discrete path step $n$ | $P_L$ | Set of robust policies |
| $d_{m,n}$ | $m$-th Ground BS coordinate | $L_n$ | Approximated Lyapunov function |
| $h_{BS}$ | BS height | $Q_D (s, a, \theta)$ | Constraint value network |
| $d_{m,n}$ | Distance between the $m$th GBS and the UAV | $Q_L (q_n, \alpha)$ | Lyapunov value function |
| $L (\cdot)$ | Path loss model | $Q_C (s, a, \theta)$ | Cost value network |
| $\alpha_L$, $\alpha_{NL}$ | Path loss exponents for LoS and NLoS | $Q_T (q_n, \alpha)$ | Stopping time value network |
| $X_L$, $X_{NL}$ | LoS, NLoS intercepts | $P_{\pi}$ | Samples priority |
| $\phi_1$, $\phi_2$ | Antenna tilt at $f_1$, $f_2$ | $\delta$ | TD-error |
| $P_{TX}$ | Transmit power of ground BS | $\pi_T$ | Teacher policy |
| $\sigma_n^2$ | Thermal noise power | $H$ | Prioritized replay memory |
| $h$, $m_u$ | Fading, Nakagami fading parameter | $B$ | Minibatch |
| $\gamma_{m,n}$, $\tilde{\gamma}$ | SINR, SINR threshold | $\Sigma$, $\Upsilon$ | Known, Unknown space |
| $S_D$ | Subset of outage regions | $D_1$, $D_2$ | Old domain, new domain |
| $F (q_n)$ | Radio failure indicator | $C$, $Z$ | Size of known space memory, size |
| $d_O$ | Connectivity outage constraint | $\Theta$, $\Lambda$ | Density threshold, Risk function |
| $d_{\text{ch}}$ | Max tolerated radio failures | $\pi_2$ | Policy in new domain |

Fig. 1. UAV is flying in an urban environment where the ground BS–UAV link might be occasionally blocked by buildings based on the building distribution and UAV height, leading to radio failures.

A. Ground-to-Air Channel Model

We consider wireless ground-to-air channel ground BS–UAV characterized by deterministic large-scale path loss and random small-scale fading. We consider a generic urban environment where the ground BS–UAV link might be occasionally blocked by obstacles and buildings based on the building distribution in $X$ and UAV height. In order to present the results more generically, the path loss at $f_1$ and $f_2$ between the $m$th ground BS and the UAV can be modeled to take into account the Line-of-Sight (LoS) and Nonline-of-Sight (NLoS) cases as for [4]

$$L (d) = \begin{cases} \frac{X_L d^{-\alpha_L}}{X_{NL} d^{-\alpha_{NL}}} & (1) \\
\end{cases}$$

where $d_{m,n}$ is the ground BS–UAV distance as for (1), and parameters $\alpha_L$, $\alpha_{NL}$ and $X_L$, $X_{NL}$ represent, respectively, the path-loss exponent for LoS/NLoS and the path loss at 1-m distance. To capture the LoS and NLoS effect at sub-6 GHz, we model the small-scale fading power as Rician for the LoS and as Rayleigh for the NLoS link [17]. At mmWave, we model the small-scale fading power $h_{0,i}$ with $i \{\text{LoS}, \text{NLoS}\}$ as a Nakagami-$m$ fading model [4]. Accordingly, the fading power at mmWave follows a Gamma distribution with $\mathbb{E}[h_{0,i}^2] = 1$.

1) Antenna Model: We adopt the three-sectors antenna model as characterized by 3rd generation partnership project (3GPP) specification [35]. Similar to [36], we consider that each sector is separated by 120° and equipped with a vertical $N$-element uniform linear array (ULA) tilted with angle $\phi_1$ at $f_1$ and a uniform planar square array (UPA) $N \times N$ tilted with angle $\phi_2$ at $f_2$. The dB gain experienced by a ray with elevation and azimuth angle pair $\theta, \phi$ due to the effect of the element radiation pattern can be expressed as

$$A_{3GPP} (\theta, \phi) = G_{\max} - \min \left[ -A_{E,V} (\theta) + A_{E,H} (\phi) , A_m \right]$$

where $G_{\max} = 8$ dBi is the maximum directional gain of the antenna element. The 3GPP element radiation pattern of each single antenna element is composed of horizontal and vertical radiation patterns $A_{E,H} (\phi)$ and $A_{E,V} (\theta)$. Specifically, this last pattern $A_{E,V} (\theta)$ is obtained as

$$A_{E,V} (\theta) = -\min \left\{ \frac{12 (\theta - 90^\circ)^2}{\theta_{3dB}}, SLAV \right\}$$

where $\theta_{3dB} = 65^\circ$ is the vertical beamwidth, and $SLAV = 30$ dB is the side-lobe level limit. Similarly, the horizontal pattern is computed as

$$A_{E,H} (\phi) = -\min \left\{ \frac{12 (\phi_{3dB})^2}{\phi_{3dB}}, A_m \right\}$$

where $\phi_{3dB} = 65^\circ$ and $A_m = 30$ dB is the front–back ratio. The relationship between the array radiation pattern and a single pattern is defined as $A_A (\theta, \phi, n) = A_E (\theta, \phi) + A (\theta, \phi, n)$,
where \( n \) is the number of antenna elements and \( AF(\theta, \phi, n) \) is the array factor. \( AF(\theta, \phi, n) \) is given in [36] as
\[
AF(\theta, \phi, n) = 10 \log_{10}[1 + \rho(\mathbf{a} \cdot \mathbf{w}^T)]
\]
(6)
where \( \rho \) represents the correlation coefficient set to unity, \( \mathbf{a} \) is the amplitude vector and \( \mathbf{w} \) is the beamforming vector. The definition of \( \mathbf{w} \) can be found in [18] and [36] and is omitted here. As in [18] and [36], we consider an equal and fixed amplitude for the antenna elements. As a consequence, \( \mathbf{a} \) is set to \((1/\sqrt{n})\) and we leave the integration of optimized weights in beamforming techniques for future investigation.

Finally, we assume the UAV is equipped with a conventional isotropic antenna of unitary gain in any direction to maintain low complexity and cost.

For simplicity, but without loss of generality, we focus on one typical UAV. At any time step during the UAV’s mission, the UAV associates with one ground BS’s sector only, following a maximum Signal-to-Noise-plus-Interference Ratio (SINR) scheme [4]. The maximum instantaneous SINR received at the omnidirectional UV from the attached \( m \)th ground BS can be defined as
\[
\gamma_{m,n} = \frac{P_{BS}L(d_{m,n})h_{BS}^2(d_{m,n})g_{m,n}}{\sigma^2 + I_t}
\]
(7)
where \( L() \) is the path loss, the random variable \( h \) accounts for the fading, and \( I_t \) the interference associated with the non-attached BSs. The term
\[
g_{m,n} = \arg \max_{m \in \mathcal{G}, j \in \mathcal{J}} g_{m,j}
\]
is the antenna gain from the \( j \)th sector of the \( m \)th ground BS and \( j \in \mathcal{J} = \{1, 2, 3\} \) denotes the set of sectors.

For a given SINR threshold \( \gamma \), an outage occurs at step \( n \) if at the UAV the condition \( \gamma_{m,n} < \gamma \) is not satisfied. The resulting outage probability can be denoted as
\[
P_{\text{outage}}(\mathbf{q}_n, m) = \Pr(\gamma_{m,n} < \gamma)
\]
(8)
where \( \Pr \) is the probability of the event taken with respect of the randomness of the fading. Note that the value of \( \Delta T \) can be considered small enough to satisfy \( \Delta TV \ll h_n \), and in the generic flying step \( n \), the UAV can be considered stationary [17], [37]. Let us define a radio failure indicator on the ground-to-air link as
\[
F(\mathbf{q}_n) = \begin{cases} 
1, & \text{if } P_{\text{outage}}(\mathbf{q}_n, m) \geq P_{th} \\
0, & \text{otherwise}
\end{cases}
\]
(9)
Thus, we can introduce \( \mathcal{S}_0 \) as the subset of outage regions where \( (9) \) holds true. Then, for an arbitrary outage probability threshold \( P_{th} \) and a given path \( \mathbf{q}_n \) with \( \mathbf{q}_n \notin \mathcal{S}_0 \), the connectivity outage constraint \( d_O \) can be expressed as
\[
d_O = \sum_{n=1}^{N} F(\mathbf{q}_n).
\]
(10)

B. Problem Formulation

We would like the UAV to reach the destination in the shortest possible number of moves, while keeping the outage events lower than \( d_O \). The UAV’s velocity is limited to its maximum speed. We consider that during its mission, the UAV moves at constant \( V = V_{\text{max}} \). This assumption of constant maximum speed makes the mathematical modeling more tractable as the variable speed will have control and/or aerodynamic related reasons which are out of our control and scope of work. In addition, using UAV’s maximum speed allows the UAV to reach the destination in the minimum path steps. The UAV maximum speed used in this article to derive the numerical results is a realistic and aerodynamic supported maximum speed, used in several related connectivity-aware path design works [17], [19], [27]. Thus, the mission variable \( T \) can be expressed as
\[
T = \left( \left. \sum_{n=1}^{w} \| \mathbf{q}_n - \mathbf{q}_{n-1} \right\| \right)/V_{\text{Max}}
\]
and optimization problem can be formulated as in (11)
\[
\min_{\omega} \sum_{n=1}^{w} \| \mathbf{q}_n - \mathbf{q}_{n-1} \| \quad \text{s.t.} \\
\quad \mathbf{q}_0 = \mathbf{q}_f, \quad \mathbf{q}_N = \mathbf{q}_f \quad \text{(11b)} \\
\quad h_n > h_{\text{BS}} \quad \text{(11c)} \\
\quad \| \mathbf{q}_n - \mathbf{q}_{n-1} \| \leq \Delta TV_{\text{max}} \quad \text{(11d)} \\
\quad d_O \leq d_{\text{th}} \quad \text{(11e)} \\
\quad \omega \leq \hat{N}. \quad \text{(11f)}
\]
Variable \( \omega \) represents the mission completion steps. The constraint (11b) guarantees the initial and final points, (11c) prevents the collision between the UAV and the ground BSs. Equation (11d) constraints on the UAV’s maximum speed. The connectivity constraint \( d_{\text{th}} \) must not exceed a predefined threshold \( d_{\text{th}} \). For this reason, (11e) defines the connectivity outage duration tolerance. Constant \( \hat{N} \) in (11f) is the upper bound on the UAV steps to take into account the limited UAV endurance. In our design, \( d_{\text{th}} \) is not fixed but can be tuned to suit different application scenarios. Longer paths may help the UAV avoid \( \mathcal{S}_0 \) areas and satisfy stringent values of \( d_{\text{th}} \). In scenarios where the UAV is deployed for timing intervention a higher \( d_{\text{th}} \) might be tolerated to achieve shorter paths. Although UAV–UE and UAV–BS cases generally have different design problems, the above connectivity constraint path optimization applies to both the scenarios from the ground-to-air link point of view.

The connectivity-aware problem (11) is a nonconvex optimization problem that is generally intractable to solve via conventional optimization techniques. The proof of NP-hardness of a similar path design problem can be found in [27] and omitted here. In addition, a closed-form expression of the outage probability (8) used to compute (11e) is highly dependent on the network topology, channel fading, and antenna gain. In our previous work [4], taking into account the channel characteristics at \( f_1 \) and \( f_2 \), we have investigated a stochastic geometry approach to deduce a tractable form of \( P_{\text{outage}} \). However, statistical approaches provide useful insights on the average performance of the network but they do not capture the actual complexity of the local environment where the UAVs are deployed.
RL approaches, that interact iteratively with the environment, circumvent these issues solving the path optimization problem using the power measurements at the UAV in a certain time step. While RL algorithms for the design of UAV connectivity-aware path have been proposed already in literature (Table I), this article aims to propose a novel TL approach to improve the efficiency of DQN for UAV path design. More specifically, adopting the dual band system model described in Section II, we focus on two fundamental problems: 1) how to improve the efficiency of DQN for UAV path design. More interchangeably.

Note that we assume the UAV to be controlled by a dedicated agent and the terms agent and UAV will be used hereafter.

agents optimize one objective while satisfying cost constraints. In this work, the UA V moves from

radio failure indicator

The position of UAV at step \( q_{n+1} \) depends on the position and moving direction chosen by the agent at step \( q_n \). Hence, the UAV’s flight process can be regarded as a discrete-time CMDP, an extension of the Markov Decision Process (MDP) framework that suits optimization problems as (11), where agents optimize one objective while satisfying cost constraints. Note that we assume the UAV to be controlled by a dedicated agent and the terms agent and UAV will be used hereafter interchangeably.

A. Problem Formulation as CMDP

Each CMDP consists of a 7-tuple \( (S, A, \mathcal{P}, c, d, \alpha, d\hbar) \). In this work \( S \) is the space state consisting of the UAV positions within the feasible flying region with single state \( q_n \). The agent’s action corresponds to one of the UAV’s flying directions, that together represent the action space \( A \). The UAV moves in a custom environment consisting of area \( X \) covered by the dual band network. The immediate cost function is modeled as \( c(q_n, a_n) = -1 - \|q_n - q_f\| \). The first term penalizes the UAV for each move and accounts for UAV battery usage. At the same time, the second cost term measures the relative distance to the destination and encourages the UAV to complete its mission in the shortest possible number of moves. We define the immediate constraint cost \( d(q_n) \) as the radio failure indicator \( F(q_n) \) in (9), \( d\hbar \) is an upper bound on the expected cumulative constraint cost. Finally, variable \( \alpha \in [0, 1] \) in \( (S, A, \mathcal{P}, c, d, \alpha, d\hbar) \) represents the discount factor.

Under this framework, at any step \( n \), the UAV moves from \( q_n \) to \( q_{n+1} \) during step length \( \Delta_T \) at speed \( v_{\text{max}} \) based on the action \( a_n \), selected according the current policy \( \pi \). We define a policy \( \pi(a \mid q_n) \) as the condition probability to take action \( a \) given the state \( q_n \). After taking action \( a_n \), the UAV interacts with the environment receiving the immediate cost \( c(q_n, a_n) \) and constraint cost \( d(q_n) \). A sequence of interactions leads to a terminal or goal state, where an episode ends. For the computation \( F(q_n) \), note that \( \Delta_T \) usually contains many channel coherence blocks due to small-scale fading.

As a result, it can be assumed that as long as the UAV performs signal measurements sufficiently frequently, the outage probability (8) can be evaluated by its empirical value \( \hat{P}_{\text{outage}}(q_n, m) = 1 / \sum_{j=1}^{J} O(q_n, m) \) where \( J \gg 1 \) and \( O() \) is 1 if \( \gamma_{m,n} \leq \gamma \) and 0 otherwise (17). To complete the CMDP formulation of (11), we need to formalize the constraint (11e) that bounds the total frequency of visiting \( S_O \) with a predefined threshold \( d\hbar \) into a CMDP. We rewrite (11e) using the immediate constraint notation \( d(q_n) = I(q_n \in S_O) \), where \( I(x) \) denotes the indicator function so that its value is 1 if \( x \in S_O \) and 0 otherwise [38]. Thus, constraint (11e) becomes

\[
\mathbb{E} \left[ \sum_{n=1}^{N} I(q_n \in S_O) \mid q_f, \pi \right] \leq d\hbar. \tag{12}
\]

Let us now denote \( \Pi \) the set of Markov stationary policies with policy element \( \pi \), such that \( \Pi(q_n) = \{ \pi(\cdot \mid q_n) : S \rightarrow R_{\geq 0} : \sum_{a} \pi(a \mid q_n) = 1 \}, \forall q_n \in S \) follows from the stationary property. Given a policy \( \pi \in \Pi \) that maps states to actions and an initial state \( q_f \), we can define the expected cumulative cost function as

\[
C_\pi(q_f) = \mathbb{E} \left[ \sum_{n=1}^{N} c(q_n, a_n) \mid q_f, \pi \right]. \tag{13}
\]

and the robustness constraint function as

\[
D_\pi(q_f) = \mathbb{E} \left[ \sum_{n=1}^{N} d(q_n) \mid q_f, \pi \right]. \tag{14}
\]

The optimization problem (11) becomes then

\[
\pi^* \in \min_{\pi} \mathbb{E} \left[ \sum_{n=1}^{N} c(q_n, a_n) \mid q_f, \pi \right] \tag{15a}
\]

\[
\text{s.t. } \mathbb{E} \left[ \sum_{n=1}^{N} d(q_n) \mid q_f, \pi \right] \leq d\hbar. \tag{15b}
\]

The goal of the agent is to find the optimal policy \( \pi^* \) that minimizes the long term cost while satisfying the connectivity constraint.

We propose using the Lyapunov function-based method to derive a robust optimal policy \( \pi^* \), solution of (15) in domain \( D_1 \) at \( f_1 \). The rationale behind the Lyapunov approach is to find a set of robust actions that meet the condition (15b) and guarantee global robustness during training. As a consequence, it can be considered a suitable candidate as teacher in the TL process. From here on, for simplicity, we will refer to the robust policy \( \pi_1 \) in domain \( D_1 \) as teacher policy \( \pi_T \). To the best of our knowledge, this is the first time a Lyapunov approach is used to derive a teacher policy for transfer advice for the UAV connectivity-aware path problem.

B. Background of the Lyapunov-Based Robust Policy

We introduce the notion of Lyapunov function following the definition in [38]: given a baseline policy \( \pi_B \), i.e., \( D_{\pi_B}(q_f) \leq d\hbar \), a function \( L : S \rightarrow R \) is said to be a Lyapunov function.
function w.r.t initial state $q_1$ and constraint threshold $d_{th}$ if it satisfies the following conditions:

\begin{equation}
T_{\pi, \theta}(q_1) \leq 0 \quad \forall q_1 \in S
\end{equation}

\begin{equation}
T_{\pi, \theta}(q_1) \leq d_{th}
\end{equation}

\begin{equation}
L(q_1) \leq 0.
\end{equation}

We denote the constraints in (16a) as Lyapunov constraints and (16b) as robustness condition. Term $T_{\pi, \theta}$ is the generic Bellman operator w.r.t a policy $\pi$ and constraint cost function $d$

\begin{equation}
T_{\pi, \theta}(V(q_1)) = \sum_{a} \pi(a \mid q_1) \left[ d(q_1, a) + \sum_{q'_1 \in S} P(q'_1 \mid q_1, a) V(q'_1) \right]
\end{equation}

where $q'_1$ is next state $q_{n+1} \in S$ under action $a$.

Given any arbitrary Lyapunov function $L$, consider the set $F_L(q_1) = \{ (\pi \mid q_1) \in \Pi : T_{\pi, \theta}(L(q_1)) \leq L(q_1) \} \forall q_1 \in S$. Given the contraction property of $T_{\pi, \theta}$ [37], together with $L(q_1) \leq d_{th}$, any policy $\pi$ in this set satisfies the robustness conditions and is a feasible solution of (15). This set of robust policies is defined as the $L$-induced policy set. Since it is not guaranteed that the set $F_L(q_1)$ contains any optimal solution of (15), the goal of the Lyapunov approach in this article is to formulate an appropriate function $L$, such that $F_L$ contains an optimal policy $\pi^*$ to work as $\pi_T$. Finding an appropriate Lyapunov function may not be an easy task. [38, Lemma 1] ensures that without loss of optimality, the Lyapunov function that satisfies the above criterion can be expressed as

\begin{equation}
L_{\pi^*}(q_1) = \mathbb{E} \left[ \sum_{n=0}^{N} d(q_n) + \epsilon(q_n) \mid \pi^*, q_1 \right]
\end{equation}

in which $\epsilon(q_n)$ is an auxiliary constraint. Thus, finding $L$ that satisfies the above condition is equivalent to performing appropriate cost-shaping with auxiliary $\epsilon$ which can be built using the method proposed in [38]. This method approximates $\epsilon$ to a constant function, which is independent of state and can be computed more efficiently as

\begin{equation}
\hat{\epsilon} = \frac{d_{th} - D_{\pi^*}(q_1)}{\mathbb{E}[T^* \mid q_1, \pi^*]} q_1 \in S
\end{equation}

where $\mathbb{E}[T^* \mid q_1, \pi^]$ is the expected stopping time of the CMDP. To speed up the computation of the expected stopping time, we replace the denominator of (19) with the upper bound $N$, maximum number of allowed steps, leading to $\hat{\epsilon} = (1/N)(d_{th} - L(q_1))$. Substituting this last equation into (18), the Lyapunov function becomes

\begin{equation}
L_{\pi^*, \epsilon}(q_1) = \mathbb{E} \left[ \sum_{n=0}^{N} d(q_n) + \hat{\epsilon}(\pi^*, q_n) \right]
\end{equation}

and the set of robust policies $F_L(q_1)$ can be written as

\begin{equation}
F_L(q_1) = \{ (\pi \mid q_1) \in \Pi : T_{\pi, \theta}(L(q_1)) \leq L_{\pi^*, \epsilon}(q_1) \}.
\end{equation}

The above formulation can be used to propose a robust policy and value iteration algorithm, in which the goal is to solve the Linear Programming (LP) problem [38]

\begin{equation}
\pi^* \mid q_1 = \arg \min_{\pi \in \Pi} \{ \pi(\cdot \mid q_1)^T Q_c(q_1, \cdot) \}
\end{equation}

\begin{equation}
(\pi(\cdot \mid q_1) - \pi^*(\cdot \mid q_1))^T L_{\pi^*}(q_1, \cdot) \leq \hat{\epsilon}
\end{equation}

where $Q_c(q_1, a) = d(q_1) + \hat{\epsilon} + \alpha \sum_{q_n \in S} P(q_n \mid q_1, a) L_{\pi^*, \epsilon}(q_n)$ is the Lyapunov function and $Q_c(q_1, a) = c(q_1, a) + \alpha \sum_{q_n \in S} P(q_n \mid q_1, a) V_C(q_n)$ and $V_C(q_1) = T_{\pi^*, \theta}(V_C)(q_1)$ are the state action value function and the value function (w.r.t. the cost function $c$).

Since we assume that the environment in which the UAV is flying is composed by a large and continuous state space, solving (22) becomes numerically intractable. To address this issue, in the next section, we propose a DDQN approach.

C. Lyapunov Approach DDQN for Connectivity-Aware Path Design

In this section, we use the above-derived Lyapunov function to derive an optimal robust policy via DDQN. Using the notation of action-value function [37], we can write the Lyapunov state–action–value function $Q_\theta(q_1, a)$ as

\begin{equation}
Q_\theta(q_1, a) = Q_D(q_1, a) + \epsilon Q_T(q_1)
\end{equation}

where $Q_D(q_1, a)$ represents the constraint state–action–value function. The stopping time value network $Q_T(q_1)$ is a function related to the number of remaining steps and discount factor, and can be computed as $Q_T(q_1) = \sum_{n=0}^{N} \alpha^{n} q_1 \in S$.

If $\pi_B, Q_D(q_1, a)$ and $Q_T(q_1)$ are known, the auxiliary cost in (19) can be computed as

\begin{equation}
\epsilon'(q_n) = \epsilon' = \frac{d_{th} - \pi_B(\cdot \mid s_0)^T Q_D(q_1, a) \pi_B(\cdot \mid s_0)^T Q_T(q_1)}{\pi_B(\cdot \mid s_0)^T Q_T(q_1)}.
\end{equation}

Finding the optimal policy $\pi^*$ through (22) and (24) requires accurate calculation of $Q_D(q_1, a)$, $Q_C(q_1, a)$, and $\pi_B$. One traditional way to derive optimal action-value functions is table-based method, which requires storing and maintaining a state–action–value table, one value for each state–action pair. However, for the path design under consideration, the state–action–value table would exponentially grow with the size of the flying area. To overcome this issue, parametric functions can be trained to approximate the state–action value. Specifically, we utilize Neural Networks (NNs) to perform function approximation. Let $\hat{Q}_D(q_1, a; \theta_D)$, $\hat{Q}_C(q_1, a; \theta_C)$ be the parameterized evaluation networks with weights $\theta_D$ and $\theta_C$, then (23) becomes

\begin{equation}
\hat{Q}_L(q_1, a; \theta_D) = \hat{Q}_D(q_1, a; \theta_D) + \epsilon' Q_T(q_1)
\end{equation}

where $\epsilon'$ is computed as

\begin{equation}
\epsilon'(q_n) = \epsilon' = \frac{d_{th} - \pi_B(\cdot \mid s_0)^T \hat{Q}_D(q_1, a; \theta_D)}{\pi_B(\cdot \mid s_0)^T \hat{Q}_T(q_1)}.
\end{equation}

To train the networks $\hat{Q}_D$ and $\hat{Q}_C$ we minimize squared error of prioritized Bellman residuals as for a loss function that can be defined as

\begin{equation}
L_n(\theta) = p_{c,n} \left( \gamma_p - \hat{Q}_C(q_1, a; \theta_C) \right)^2
\end{equation}
where $p_{c,n}$ and $p_{d,n}$ are the sample priority. In the above equations, term $y_n^d$ is the target cost value, expressed as

$$y_n^c = c_{n,n+1} + \alpha^N \pi (\cdot | q_n')^T \hat{Q}_C(q_{n+1}, a^*, \theta_C^-)$$

(29)

where $a^*$ is

$$a^* = \arg \max \hat{Q}_C(q_{n+1}, a', \theta_C)$$

(30)

to separate the action selection and the action evaluation as for double Q-learning technique [40]. Similarly, the target $y_n^d$ for the constraint cost can be denoted as

$$y_n^d = d_{n,n+1} + \alpha^N \pi (\cdot | q_n')^T \hat{Q}_D(q_{n+1}, a^*, \theta_D^-).$$

(31)

In each iteration, the agent takes action $a_n$ generated by current baseline policy $\pi_B$ and perform a DDQN update, computing the loss to update the weights $\theta_C$ and $\theta_D$ of networks $\hat{Q}_D$ and $\hat{Q}_C$. To derive a reasonable baseline policy $\pi_B$ for the UAV path design under study we create another DNN. As a result, the baseline strategy action probability is approximated by the output of the DNN, namely $\pi_B \approx \hat{\pi}(\cdot | q_n, \theta_C)$. We train the policy network by optimizing a loss function that considers that target functions are.

$$L(\theta_C) = E_{q_n}[D_{KL}(\hat{\pi}(\cdot | q_n; \theta_C) || \pi^*(\cdot | q_n))].$$

(32)

Note that in (27)–(31), to improve the stability and convergence of our algorithm, we exploit different techniques. Unlike the conventional Q-learning where target functions are produced by using one-step look-ahead, we use $n$-step look-ahead, or multistep learning technique. Specifically, in the target (29), (31) the truncated $N_1$-step cost, and constraint cost from a given state $q_n$ are defined as

$$c_{n,n+1} = \sum_{i=0}^{N_1-1} \alpha^i c_{n+1+i}$$

(33)

$$d_{n,n+1} = \sum_{i=0}^{N_1-1} \alpha^i d_{n+1+i}.$$  

(34)

In conventional DRL, after executing the action, the agent stores the state–action–reward transition into a replay memory. In a second step, the agent performs the weight updates selecting a random sample of $|B|$ instances to break the correlation between instances [41]. However, sampling randomly the minibatch $B$ may affect the convergence of the training procedure. For this reason samples can be selected according to a priority determined by their temporal difference (TD) error, which can be computed as $\delta_i = |y_j^i - Q_D(q_n, a, \theta_D)|$,

$$\delta = \frac{\sum_{j=1}^{|B|} |y_j^i - Q_D(q_n, a, \theta_D)|}{|B|}.$$  

In this work, we apply a replay prioritization scheme that considers that target functions are produced by using a multistep learning technique. Samples and TD errors are stored in a sliding window $W$ for $N_1$ transitions to enable multistep learning. The sampling priority $p_{c,n}$ and $p_{d,n}$ in (27) and (28) are given by a weighted sum of two different components as

$$\eta \max_i \delta_i + (1 - \eta) \bar{\delta}$$  

(35)

where in the general $\delta$, we omitted the subscript $c$ or $d$ to simplify the notation. $\bar{\delta}$ is used for the computation of $p_{c,n}$ and $\delta_d$ for $p_{d,n}$. The term $\max_i \delta_i$ is the max absolute $N_1$-step TD error $\delta$ contained within the $|B|$-length sequence, $\eta$ is a tunable parameter $\in [0, 1]$. The second term is the sequence mean absolute $N_1$-step TD error. Finally, it can be noted in (29) and (31) that $\hat{Q}_C(q_{n+1}, a^*, \theta_C^-)$, $\hat{Q}_D(q_{n+1}, a^*; \theta_D^-)$ are target networks of the evaluation networks. A target network has the identical NN structure of the related evaluation network, but its weights $\theta_C^-; \theta_D^-$ are updated only each $i$ iterations by copying the weights from the evaluation. In this way, the correlation between the target and estimated $Q$-values is reduced.

The proposed algorithm to derive a robust UAV path via Lyapunov method is summarized in Fig. 2, while Algorithm 1 presents the pseudocode.

### IV. TRANSFER LEARNING VIA TEACHER POLICY

In this section, we describe the Teacher Advice algorithm to provide external knowledge and allow the agent pretrained in $D_1$ to quickly adapt to the new environment $D_2$. Let us assume there exists a policy $\pi_T$, solution function of (15) mapping states to actions, in a defined domain $D_1 = (S, A, P, C_1, D_1, \alpha, d_{th})$ at $f_1$ to get from a particular
Set $I$ Update the target networks after $I$ iterations.

Obtain

Update the DNN of state action constraint $Q_C$ performing gradient descent on loss (27) with respect to $\theta_C$;

Update the DNN of state action cost function $Q_D$ performing gradient descent on loss (28) with respect to $\theta_D$;

Update the priority weights $p_{c,n}, p_{d,n}$ based on TD error;

Obtain $\pi^*$ by (22);

Update $\hat{\pi}(\cdot | q_n; \theta_\pi)$ via $\theta_\pi \leftarrow \theta_\pi - \alpha \nabla_\theta L(\theta_\pi)$;

end

Update the target networks after $I$ iterations.

Set $\pi_T = \pi^*$ and $\theta_T = \theta_\pi$;

end

Algorithm 1: Robust DDQN Algorithm for Connectivity-Aware Path

1. **Initialize:** maximum number of episodes, the prioritized replay memory $H$ with capacity $N$, mini batch size $|B|$;

2. **Initialize:** Upper limit of Radio Failures $TH_1$, UAV flight speed; for episode $= 1,...,Max\ episode$ do

   3. Initialize a sliding window queue $W$ with capacity $N_1$;

   4. Initialize $q_0 = (q_1) \in S \setminus S_0$, set step $k \leftarrow 0$;

   5. for each step of episode do

      6. Select action $a_n$ according to parameterized network $\hat{\pi}(\cdot | q_n; \theta_\pi)$;

      7. Agent execute action $a_n$, observe $[q_{n+1}]$ and $c_n,d_n$;

      8. Store experience $(q_n, a_n, c_n, d_n, q_{n+1}, \delta_c, \delta_d)$ in sliding window queue $W$;

      9. When reached a number $N_1$ of transitions, store them in replay memory $H$ and compute (33) and (34);

     10. From buffer $H$ sample minibatch $B$ of $N_1$ experience according to the priority as for (35);

     11. Update the DNN of state action cost function $Q_C$ performing gradient descent on loss (27) with respect to $\theta_C$;

     12. Update the DNN of state action constraint function $Q_D$ performing gradient descent on loss (28) with respect to $\theta_D$;

     13. Update the priority weights $p_{c,n}, p_{d,n}$ based on TD error;

     14. Obtain $\pi^*$ by (22);

     15. Update $\hat{\pi}(\cdot | q_n; \theta_\pi)$ via $\theta_\pi \leftarrow \theta_\pi - \alpha \nabla_\theta L(\theta_\pi)$;

     16. end

Thus, the definition of a known state is as follows.

**Definition 1:** Given a density threshold $\Theta$, a state $q_n$ is considered known when $\Lambda_{\pi_T}(q_n) = 0$ and unknown in all other cases. Formally, $\Sigma \subseteq S$ is the set of known states, while $\Upsilon \subseteq S$ is the set of unknown states with $\Sigma \cap \Upsilon = 0$.

Using the known space set, we could transfer the learner the advice to prefer some actions over others in specific regions of the feature space. However, a direct translation of the action frequency of risky states. Here, it is important to note that the role of the teacher policy is not to supply the best action but to advice an action more robust than the one obtained through random exploration. Fig. 3 summarizes the overall TL process adopted in this article. Note that the proposed TL method is applicable to any DRL algorithm and it is not specific to the robust-DDQN only.

A. Initial Known Space

The agent, equipped with an empty memory $C$ of size $Z$, builds the initially known space by storing new experiences. Using $\pi_T$, we run $Q < Z$ iterations with the environment and collect states, actions taken, reward received (cost and constraint cost in this case), and if the current state is terminal. The stored data follows the structure of the experience replay memory used in conventional DQN. Each memory element represents a transition the agent has experienced in domain $D_1$. The resulting data forms the known space. When the agent enters a new state $q_{n}$, it computes the Euclidean distance to determine if $q_{n}$ belongs to the known space. Hence, we define a density threshold $\Theta$ and a risk function as [42]

$$
\Lambda_{\pi_T}(q_{n}) = \begin{cases} 
0, & \text{if } \min_{q_{n}\in \Sigma} d_{n,q} \leq \Theta \\
1, & \text{otherwise}
\end{cases}
$$

where $d_{n,q} = \|q_{n} - q_{n}\|$ is the Euclidean distance between a new state and the states in memory. The parameter $\Theta$ defines the classification region for a new state $q_{n}$, and it is dependent on the size of the action. In this work, we consider $\Theta = 2\Delta TV_{\text{max}}$. When the distance of the nearest neighbor to $q_{n}$ is greater than $\Theta$, the experience is added to the memory.

The agent, equipped with an empty memory $C$ of size $Z$, builds the initially known space by storing new experiences. Using $\pi_T$, we run $Q < Z$ iterations with the environment and collect states, actions taken, reward received (cost and constraint cost in this case), and if the current state is terminal. The stored data follows the structure of the experience replay memory used in conventional DQN. Each memory element represents a transition the agent has experienced in domain $D_1$. The resulting data forms the known space. When the agent enters a new state $q_{n}$, it computes the Euclidean distance to determine if $q_{n}$ belongs to the known space. Hence, we define a density threshold $\Theta$ and a risk function as [42]

$$
\Lambda_{\pi_T}(q_{n}) = \begin{cases} 
0, & \text{if } \min_{q_{n}\in \Sigma} d_{n,q} \leq \Theta \\
1, & \text{otherwise}
\end{cases}
$$

where $d_{n,q} = \|q_{n} - q_{n}\|$ is the Euclidean distance between a new state and the states in memory. The parameter $\Theta$ defines the classification region for a new state $q_{n}$, and it is dependent on the size of the action. In this work, we consider $\Theta = 2\Delta TV_{\text{max}}$. When the distance of the nearest neighbor to $q_{n}$ is greater than $\Theta$, the experience is added to the memory.

Thus, the definition of a known state is as follows.

**Definition 1:** Given a density threshold $\Theta$, a state $q_{n}$ is considered known when $\Lambda_{\pi_T}(q_{n}) = 0$ and unknown in all other cases. Formally, $\Sigma \subseteq S$ is the set of known states, while $\Upsilon \subseteq S$ is the set of unknown states with $\Sigma \cap \Upsilon = 0$.

Using the known space set, we could transfer the learner the advice to prefer some actions over others in specific regions of the feature space. However, a direct translation of the action

Fig. 3. Illustration of our proposed TL algorithm: a pretrained policy in domain $D_1$ is used as teacher in domain $D_2$. starting point to a goal, given a set of outage states. Let us now consider a domain $D_2$ at frequency $f_2$ that differs from domain $D_1$ by the constraint cost distribution: $D_1, \neq D_2$. We propose to consider $D_1$ as the old domain and $D_2$ as the new domain.

To reduce the computational burden of the training process in the new domain, we propose leveraging transfer learning to learn an optimal policy by leveraging exterior information from $D_1$ as well as internal information from $D_2$. The robust teacher policy $\pi_T$ guides the exploration process in domain $D_2$ at frequency $f_2$ in two ways (Fig. 3). In the first step, we use robust trajectories generated using a pretrained $\pi_T$ to provide prior knowledge about the task. To reach this goal, we utilize the concepts of known and unknown spaces [42] to cover some regions of the feature space. In a second step, the agent transfers the pretrained weights from $D_1$ and starts its training in the new domain $D_2$. The teacher policy $\pi_T$ is used to support the exploration process when the agent meets an unknown state. In new situations, the learning agent evaluates a state from the perspective of the old domain to reduce the
in the new domain would heavily limit the agent ability in domain $D_2$. To make our approach robust to imperfections in the advice or teacher policy, we are interested in providing the learning agent with the possibility to refine the transferred knowledge based on its subsequent trajectories in domain $D_2$. In what follows, we present the algorithm for the training of the learner agent in domain $D_2$.

### B. Training in New Domain

The algorithm to train the learning agent in the new domain is composed of an initialization step and an RL step. The different steps that can be summarized as follows.

1) **Initialization Step:** In this step the hyperparameters, the density threshold $\Theta$ and the initial state $q_i$ are initialized. The algorithm transfers the weights of the teacher DNN pretrained in $D_1$ to domain $D_2$, in DNN networks with identical structure. In addition, to obtain new and improved ways to complete the task, we add Gaussian noise to the initial weights such that $\theta_{\pi_2} = \theta_{\pi_1} + N(0, \sigma^2)$.

2) **RL Step:** In this step, the training in $D_2$ starts and the algorithm refines the policy to satisfy the connectivity constraint in domain $D_2$. When the UAV flies in a new position, if the state is known, $q_i \in \Sigma$, the agent performs an action $a_n$ using policy network $\hat{\pi}(\cdot | q_i; \theta_{\pi_2})$ and train the networks in domain $D_2$. In unknown states, instead, the action $a_n$ is performed using the teacher policy $\pi_T$ and the experience is added to the known set in memory $Z$. As the exploration process and the training in $D_2$ continue, the knowledge of the agent of $D_2$ and the accuracy of $\pi_2$ improve. Hence, the algorithm utilizes the teacher policy $\pi_T$ only as a backup policy with to guide the learning away from risky states or, at least, reduce their frequency.

The pseudocode for the TL and Teacher advice is reported in Algorithm 2.

### Algorithm 2: TL for Connectivity-Aware Path Design

1. Given baseline behavior $\pi_T$ and memory $C$ with maximum size $Z$;
2. **Initialize** maximum number of episodes, density threshold $\Theta$, prioritized replay memory $H$ with capacity $N$, mini batch size $|B|$;
3. Create $\Sigma$ collecting $Q$ interactions;
4. Transfer Initial weights from $\pi_T$;
5. Set $\text{maxTotalRwEpisode} = 0$;
6. **for episode = 1,...,Max episode do**
   7. Initialize $q_0 = \{q_1\} \notin \mathcal{S}_C$, set step $k \leftarrow 0$;
   8. **for each step of episode do**
      9. Compute the closest $q_0 \in C$ to $q_o$ using (36);
      10. if ($q_n$ is known) then
          11. Select action using $\pi = \hat{\pi}(\cdot | q_n; \theta_{\pi_2})$;
          12. Execute lines 8-15 of Algorithm 1;
          13. else
              14. Choose an action using $\pi_T$;
              15. Agent execute action $a_n$, observe $\{q_{n+1}\}$ and $c_n,d_n$;
              16. Add experience to memory $C$;
      17. end
   18. end
   19. Remove least frequently used experiences in $C$;
 20. end
 21. Update the target networks after I iterations.
 22. Set $\pi_2 = \pi^*$;

### V. Numerical Results

In this section, we present the main numerical results of our findings. We first describe the radio environment used for generating the UAV trajectories. Then, we evaluate the performance of the proposed robust-DDQN algorithm in domain $D_1$ at $f_1$. We compare our approach with state of the art deep RL. Specifically, we implement an unconstrained Dueling DDQN that has been shown to suit UAV connectivity-aware path problems [17], [34]. We model the reward function to minimize the flight time and the number of radio failures for a fair comparison. Details about the implementation of the Dueling DDQN benchmark strategy will be presented in Appendix A. Finally, we validate and show the benefit of the TL approach from $D_1$ at $f_1$ to $D_2$ at $f_2$.

### A. Radio Environment

The radio environment where the UAV is flying is composed of buildings generated based on the International Telecommunication Union (ITU) model [43], which involves three parameters: 1) the ratio of land area covered by buildings to total land area; 2) the mean number of buildings per unit area; and 3) the height of buildings modeled by a Rayleigh Probability Density Function (PDF). The above parameters can be modified as specified in [44] to create Suburban, Urban, Dense Urban, and High Rise Urban environments. We have considered the last three mentioned environments as they are the most challenging for connectivity-aware UAV path and to demonstrate the generality of our approach. Each environment has a different BS number, BS power, and height within a geographical area of $L \times L$, as for BS density specified in [45]. At frequency $f_1 = 2$ GHz, we consider eight antenna elements at the ground BS, while 64 antennas at $f_2 = 28$ GHz [36], as for the ULA and UPA antenna models described in Section II-A1. At sub-6 GHz, we adopt the 3GPP Macro Path-Loss Model for Urban scenario [46], that includes modeling for LoS and NLoS channels. The presence/absence of obstacles is determined in the simulated environment by checking whether the line BS-UAV is blocked or not by any building. A ray tracing software would allow us to include in the propagation calculation the relative permittivity and conductivity of the surface material, which is different for any building. However, this information would limit the algorithm’s training to a specific scenario or condition. The statistical ITU building model [43] used in our approach reflects the average characteristics over a large number of geographic areas of similar type and has been widely used to characterize urban environments in UAV trajectory path design [15], [17]. Using data extracted by a simulator
allows us to train the proposed robust-DDQN and TL method on a broader general scenario, improving the algorithm’s generalization. At mmWave we consider the path-loss model in (2) with \( \alpha_L = 2, \alpha_{NL} = 4, X_L, X_{NL} = 5e^{-4} \). We have adopted a bandwidth of 10 MHz at sub-6 GHz, 100 MHz at mmWave and a transmit power of 36 dBm at sub-6 GHz and 30 dBm at mmWave, which are in line with the specifications envisioned for downlink transmission in Fifth Generation (5G) mmWave mobile networks. We consider a UAV speed of 20 m/s [18] and ease of illustration but without loss of generality, a fixed fly altitude.

The remaining simulation parameters can be found in Table III.

### B. Performance of the Robust Teacher Policy

In this section, we show the performance of the robust Teacher Policy derived using a robust-DDQN approach. Table IV shows some hyperparameters used to generate the results. More details about the implementation of the robust-DDQN approach can be found in Appendix B. The mission is considered successful if the UAV reaches the destination before the constraint threshold is exhausted. The destination is placed in \( q_F = [700, 800, 100] \) m and the actions at each step are left-right-forward-back. To make the path task more challenging, we consider in Fig. 4 a conservative constraint threshold \( d_{th} = 10 \). Note that the training phase of the robust-DDQN model is executed for a number of 5000 episodes, each of which accounts for a maximum of \( N = 100 \) steps. The fairness of the experiment episodes is ensured by running the trials with a different preset random seed. The mission success rate is averaged over 500 evaluation episodes with a random initial starting point. The initial starting point is chosen from a continuous space in the area \( L \times L \).

### C. Performance of Transfer Learning

In this section, we investigate the potential of the TL algorithm in domain \( D_2 \). Details about the implementation of the Teacher advice and TL algorithm are in Appendix C.

---

**TABLE III**

| Parameter | Description | Value |
|-----------|-------------|-------|
| \( L \)   | Area Size   | 1 [km] |
| \( V_{\text{max}} \) | UAV Speed   | 20 [m/s] |
| \( h_h \) | UAV Height   | 100 [m] |
| \( \phi_1, \phi_2 \) | Antenna Tilt | \( f_1, f_2 \) |
| \( G_{\text{ant},x} \) | max directional gain antenna element | 8 dBi |
| \( \sigma^2 \) | Noise Power sub-6/mmWave | \(-20\text{dB-}120 \text{[dBHz]}\) |
| \( m_v \) | Nakagami Fading param. | 3 |
| \( \Delta_T \) | Time Step Length | 0.5 [s] |
| \( \tau \) | SINR Threshold | 0 dB |
| \( P_{th} \) | Outage Threshold | 0.9 |
| \( d_{th} \) | Connectivity Outage Threshold | 10% |

**TABLE IV**

| Hyperparameters robust-DDQN | Value |
|-------------------------------|-------|
| \( N \) | \( \text{n-STEP} \) | 10 |
| \( N \) | Max Steps | 100 |
| \( \alpha \) | Discount Factor | 0.99 |
| \( |B| \) | Minibatch Size | 32 |
| \( H \) | Replay Memory Size | 200000 |
| \( c \) | Move Penalty | 0.5 |
| \( d \) | Radio Failure Penalty | 1 |
| \( \eta \) | Priority Sample Weight | 0.9 [41] |

Fig. 4. Results of the robust-DDQN compared with Dueling DDQN for three urban environment with different building distribution. Term HR stands for High Rise.

Fig. 5. Convergence of the proposed robust DDQN algorithm: (a) maximum radio failure satisfaction and (b) reward control.
The impact of the TL is measured considering the asymptotic performance of the agent at mmWave. The TL algorithm is executed for 5000 trials and is compared with the algorithm executed without TL. The mission rate is again averaged over 500 episodes with the same constraint threshold as the previous section. We investigate first the case of using a teacher policy pretrained in sub-6 GHz via robust-DDQN. The results are shown in Fig. 7. The curves show the average mission success of over 500 episodes. The TL is here very effective since the algorithm with TL needs few training trials to reach the asymptotic performance of the algorithm trained tabula rasa.

In addition, Fig. 8 shows the results of the teacher advice transfer approach using a Dueling DDQN as a teacher. TL is again very powerful, as the Dueling DDQN without transfer needs at least 300 episodes to perform comparably to the algorithm with transfer.

Fig. 9 shows an example of the radio map for the High Rise environment. Fig. 9 is colored according to the average SINR. Lighter color means a higher SINR and vice versa. Generally, it is visible a different behavior between the two bands. At Sub-6GHz, lower SINR is in interference regions between the BSs. At mmWave, the lower SINR areas are more irregular due to the combined effects of the higher BS antenna tilt and building blockage. In Fig. 9(a), we plot the radio map at sub-6 GHz for a UAV height of 100 m together with two paths that start from two different initial points. The UAV reaches the destination from two different starting points during the training in the old domain. (b) Returned UAV path design in the new domain at mmWave along with the radio map.
The UAV is reusing some of the previous knowledge to reach the destination.

In conclusion, results show that the transfer advice framework proposed in this article helps a learner agent reduce the training time in successful missions using both the proposed robust-DDQN and a conventional Dueling DDQN as a teacher. Different environments with different levels of complexity in terms of coverage aware UAV navigation have been tested. This shows that the proposed TL framework is versatile and not dependent on the algorithm used to train the teacher policy. However, it is important to observe that the robust-DDQN, creating a policy that respects the connectivity constraint throughout training, results in a better teacher policy.

VI. CONCLUSION

In this article, we have developed a DDQN Lyapunov-based approach to solve a nonconvex UAV connectivity-aware path design across different simulation environments. We then proposed a transfer learning technique to improve the agent learning in a new domain at mmWave using the knowledge gained in a domain $D_1$ at sub-6 GHz. We have evaluated the efficiency of our TL approach using a Lyapunov-based DDQN teacher policy derived at sub-6 GHz benchmarked with a Dueling DDQN. Our approach showed the potential of the proposed TL framework to save many training episodes for both the teacher policies, resulting in fewer UAV flights. The learning agent’s convergence using a teacher policy derived via the Lyapunov-based DDQN is faster for all the different urban scenarios under consideration. Future works include the evaluation of the sensitivity of our algorithm to the advice of an imperfectly trained teacher.

APPENDIX A

ROBUST DDQN IMPLEMENTATION

Table V displays the architecture of the neural networks used in our robust-DDQN algorithm. Especially, Table V shows the learning rate values adjusted in the algorithm to reach convergence. The hyperparameters in Table IV are selected to both achieve a good tradeoff between learning performance and model complexity. We implement the proposed robust-DDQN based on Tensorflow library in Python. The Cost and Constraint Cost layers are all fully connected and consist of four hidden layers with ReLU and with, respectively, 512, 256, 128, and 128. The policy network consists of eight hidden layers, activated with ReLU and with, respectively, 64, 32, and 4 nodes, respectively. Weights of the policy network are initialized using the inverse distance from the UAV location to the destination, that is considered known, so that $\hat{r}(\cdot | q_\pi^t; \theta_\pi)$ approximates $1/\|q' - q_\pi^t\|$, where $q'$ is the next state after taking action $a_\pi^t$. The policy networks weights are updated each five episodes, while the Cost and Constraint Cost’s ones each 25 episodes. Adam optimizer [47] is used to apply gradient descent for all the networks. The learning rate is reported in Table V.

APPENDIX B

DDQN IMPLEMENTATION

The set of known cases $\Sigma$ is created running a number $N = 250$ trajectories using the teacher policy $\pi_T$ and collecting $Q = 9000$ iterations with the environment. The teacher policy might be derived either via the Lyapunov approach or the conventional DDQN described in the previous sections. The memory $C$ has size 200000. Thus, in a second phase, the network models trained in $D_1$ are translated into domain $D_2$. Here, the weights of the networks are perturbed with Gaussian noise $\mathcal{N}(0, 0.1)$. The training is computed using a prioritized memory of same size as for Section V.

ACKNOWLEDGMENT

The authors wish to thank Brian Keogh from University College Dublin for the helpful discussions. The authors also thank the Editor and anonymous reviewers for their constructive comments.

REFERENCES

[1] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, May 2016.
[2] Y. Zeng, J. Lyu, and R. Zhang, “Cellular-connected UAV: Potential, challenges, and promising technologies,” IEEE Wireless Commun., vol. 26, no. 1, pp. 120–127, Feb. 2019.
[3] G. Fontanesi, H. Ahmadi, and A. Zhu, “Over the sea UAV based communication,” in Proc. EucNC, 2019, pp. 374–378.
[4] G. Fontanesi, A. Zhu, and H. Ahmadi, “Outage analysis for millimeter-wave fronthaul link of UAV-aided wireless networks,” IEEE Access, vol. 8, pp. 111693–111706, 2020.
[5] G. Geraci, A. Garcia-Rodriguez, L. G. Giordano, D. López-Pérez, and E. Björnson, “Understanding UAV cellular communications: From existing networks to massive MIMO,” IEEE Access, vol. 6, pp. 67853–67865, 2018.
[6] S. Zhang, Y. Zeng, and R. Zhang, “Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective,” IEEE Trans. Commun., vol. 67, no. 3, pp. 2580–2604, Mar. 2019.
Anding Zhu (Fellow, IEEE) received the Ph.D. degree in electronic engineering from University College Dublin (UCD), Dublin, Ireland, in 2004.

He is currently a Professor with the School of Electrical and Electronic Engineering, UCD. His research interests include high-frequency nonlinear system modeling and device characterization techniques, high-efficiency power amplifier design, wireless transmitter architectures, digital signal processing, and nonlinear system identification algorithms.

Prof. Zhu was a recipient of the 2021 IEEE MTT-S Microwave Prize. He served as the Secretary for the Administrative Committee (AdCom) for the IEEE Microwave Theory and Technology (MTT-S) in 2018. He is currently an Elected Member of the IEEE MTT-S AdCom, the Chair of the Electronic Information Committee, and the Vice-Chair of the Marketing and Communications Committee. He served as the Chair for the IEEE MTT-S Microwave High-Power Techniques Committee (TC-12) in 2020 and 2021. He is a Track Editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES and an Associate Editor of the IEEE Microwave Magazine. He is also a member of the IEEE Future Directions Committee.

Hamed Ahmadi (Senior Member, IEEE) received the Ph.D. degree from the National University of Singapore, Singapore, in 2012.

He was a Ph.D. scholar with Institute for Infocomm Research, A-STAR, National University of Singapore. Since then, he worked at different academic and industrial positions in the Republic of Ireland and U.K. He is a Senior Lecturer (Associate Professor) with the School of Physics, Engineering and Technology, University of York, York, U.K., He is also an Adjunct Academic with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, Ireland. He has published more than 70 peer reviewed book chapters, journal, and conference papers. His current research interests include design, analysis, and optimization of wireless communication networks, the application of machine learning in wireless networks, green networks, airborne networks, digital twins of networks, and Internet of Things.

Dr. Ahmadi is a member of the editorial board of IEEE SYSTEMS JOURNAL, IEEE Communication Standards Magazine, and Wireless Networks (Springer). He is a Fellow of the U.K. Higher Education Academy, and a Networks Working Group Chair of COST Action CA20120 (INTERACT).

Mahnaz Arvaneh (Member, IEEE) received the Ph.D. degree in advanced brain–computer interface from Nanyang Technological University, Singapore, in 2013.

She is a Senior Lecturer (Associate Professor) with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K., where she is also the Director of the Physiological Signals and Systems Laboratory. She also co-edited a book on brain–computer interface in IET, and contributed to i-Human, Royal Society Neural Interfaces Perspective, launched in September 2019. Her expertise is in brain–computer interface, neural signal processing, and their applications in monitoring and enhancing physical and mental health.

Dr. Arvaneh has been serving as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING since 2018 and the Nature Scientific Reports since 2021.