Esraa M. Ghourab¹, Wael Jaafar², Shimaa Naser³, Sami Muhaitat³, Mahmoud Al-Qutayri⁴, and Ernesto Damiani¹

¹Center for Cyber-Physical Systems, Department of Computer and Communication Engineering, Khalifa University
²Department of Software and Information Technology Engineering, École de Technologie Supérieure
³6G Research Center, Khalifa University
⁴Systems-on-Chip (SoC) Center, Department of Computer and Communication Engineering, Khalifa University

May 17, 2024
Physical Layer Security of Wireless Communications: When Moving Target Defense Meets Unmanned Aerial Vehicles

Esraa M. Ghourab\textsuperscript{a}, Wael Jaafar\textsuperscript{b}, Shimaa Naser\textsuperscript{c}, Sanii Muhaidat\textsuperscript{c}, Mahmoud Al-Qutayri\textsuperscript{d}, and Ernesto Damiani\textsuperscript{a}

\textsuperscript{a}Center for Cyber-Physical Systems, Department of Computer and Communication Engineering, Khalifa University, UAE (e-mail: esraa.m.ghourab@mena.vt.edu, and ernesto.damiani@ku.ac.ae),
\textsuperscript{b}Department of Software and Information Technology Engineering, École de Technologie Supérieure, Montreal H3C 1K3, QC, Canada, (e-mail: wael.jaafar@etsmtl.ca),
\textsuperscript{c}6G Research Center, Khalifa University, UAE (e-mail: shimaa.naser@ku.ac.ae and muhaidat@ieee.org),
\textsuperscript{d}Systems-on-Chip (SoC) Center, Department of Computer and Communication Engineering, Khalifa University, UAE (e-mail: mahmoud.alqutayri@ku.ac.ae).

Abstract

In the realm of unmanned aerial vehicle (UAV) communication, the utilization of UAVs as aerial relays for ground nodes (GNs) introduces strategic flexibility, especially in scenarios where ground base stations may experience unforeseen impairments. However, this form of communication is vulnerable to eavesdropping by malicious entities due to the broadcast nature of wireless channels. In this paper, we tackle this problem by introducing a spatiotemporal diversification-based artificial noise (AN) injection strategy, known as moving target defense (MTD), aiming to confuse potential attackers without compromising legitimate communications. The proposed approach targets maximizing the average secrecy rate (ASR) by jointly optimizing the UAV’s trajectory and transmit power, aligned with optimizing the MTD transmit power splitting factor between legitimate and AN signals at the GN source. The formulated problem is a non-convex mixed integer nonlinear programming (MINLP) problem due to the non-convexity of the secrecy rate. To solve it, we formulate our system as a Markov decision process, and then we propose a novel deep reinforcement learning (DRL)-based approach to enhance the ASR under various system constraints. Numerical results demonstrate the superiority of the proposed algorithm over benchmarks in terms of ASR and intercept probability, showcasing its effectiveness in enhancing the security of UAV-assisted communications.

Keywords:
Artificial noise, intercept probability, moving target defense, physical layer security, power control, secrecy rate, trajectory design, UAV.

1. Introduction

Unmanned aerial vehicles (UAVs) have become integral components of future wireless networks owing to their unique attributes of mobility, flexibility, and adaptable flight patterns [1, 2]. Properly controlling the trajectories of UAVs and capitalizing on potential line-of-sight (LoS) communication between UAVs and ground nodes (GNs) over air-to-ground (A2G) channels can significantly enhance the performance of UAV-assisted communications in terms of coverage, connectivity, throughput, and energy efficiency [3]. In various applications, such as military operations, surveillance, and data collection for the Internet of Things (IoT), UAVs can enhance the operations of existing terrestrial networks [4, 2]. Despite the anticipated pivotal role of UAVs in 5G and beyond [5], the inherent openness of A2G links exposes these systems to eavesdropping attacks, highlighting the critical need to secure UAV-assisted networks [6]. Although security was conventionally relegated to the application layer tackled for instance, with cryptographic methods, these techniques face a drawback due to their high computational complexity. This results in elevated energy consumption, which may be impractical for UAV systems. A recent alternative, physical layer security (PLS), has emerged as a promising and computationally efficient approach to guaranteeing wireless communications secrecy by exploiting the channels’ inherent randomness [7].

In the context of UAV-based PLS, substantial research has been conducted over the past decade to protect nonterrestrial wireless networks [8]. The main metric widely adopted in PLS design is the secrecy rate defined as the rate at which a confidential message can be reliably transmitted without the eavesdropper inferring any information about the message [9]. Achieving a non-zero secrecy rate requires that the strength of the legitimate link surpass that of the eavesdropping link. In PLS state-of-the-art, the average channel quality of the legitimate/eavesdropping link primarily depends on path loss and shadowing from the transmitter to the receiver, determined by the locations of the legitimate transmitter/receiver and the eavesdropper. To achieve positive secrecy rates in scenarios where the average channel gain of the legitimate receiver is smaller than that of the eavesdropper, techniques that exploit wireless channel small-scale fading in time, frequency, and/or space are essential. Various methods, including power control in time and/or frequency, as well as multi-antenna beamforming, have been investigated.

Preprint submitted to Astronomy & Computing May 4, 2024
implement e-channel state information (CSI) of the eavesdropper to the legitimate transmitter typically needs to acquire the potential eavesdropper, the secrecy rate is degraded. Second, greater than the distance between the transmitter and a transmitter and the designated receiver is fixed and notably unsolved. First, when the distance between the legitimate of communication networks.

This can be achieved through appropriate UA V trajectory

or degrade the channels towards eavesdroppers. It is worth noting that the location of any GN as a potential eavesdropper can be practically detected and tracked by the UAV using a mounted optical camera or synthetic aperture radar [21]. Instead of employing static system configurations, this work advocates for the adaptive injection of AN using a spatiotemporal diversification approach, a.k.a., MTD. Specifically, the UAV trajectory design changes over space, while the power allocation of AN varies over time. Through this novel MTD-based method, potential attackers can be confused while legitimate receivers decode their signals without any performance degradation. By invalidating an attacker’s prior knowledge, MTD significantly reduces the likelihood of successful eavesdropping by effectively changing the attack surface [22].

Our proposed MTD-based method relies on the joint optimization of dynamic AN injection in source signals, power allocation, and trajectory design of the assisting UAV, aiming to maximize the system’s ASR while reducing the intercept probability at the eavesdropper. Given the non-convexity of the formulated problem, we formulate our system as a Markov decision process (MDP), subsequently, we propose an innovative iterative algorithm that aims to find a locally optimal solution through the application of deep reinforcement learning (DRL). DRL has been widely employed for the trajectory optimization of UAVs in uncertain environments, leveraging its powerful non-linear approximation capability [23, 24]. To the best of our knowledge, this work is among the first to propose joint and intelligent optimization of GN source and UAV parameters to enhance the ASR performance in the presence of an eavesdropper. The main contributions of this work can be summarized as follows:

- Develop an MTD-based method incorporating joint optimization problems in terms of trajectory, power control, and AN injection to maximize the average secrecy rate (ASR) while reducing the intercept probability of UAV-based networks.
- Propose an intelligent spatiotemporal diversification mechanism based on the double-deep Q-network (DDQN) algorithm to solve the complexity of the proposed joined optimization problem.
- Through extensive simulations, the efficiency of the proposed scheme is validated. Moreover, the obtained results, in terms of ASR and intercept probability, outperform those of the benchmarks, thus showcasing the robustness of our method and its suitability for PLS in wireless communication systems.

The remainder of the paper is organized as follows: Section 2 presents the system model details, respectively. Sections 3 and 4 illustrate the problem formulation and the proposed DRL-based approach. Section 5 provides the experiments’ results. Finally, Section 6 concludes the paper.

2. System Model

2.1. Network Model

We consider a scenario involving a base station (BS) as source node $S$, a ground destination node $D$, and a passive eavesdropper $E$, where the communication between $S$ and $D$ is assisted by a UAV relay, denoted by $R$. The communication occurs in two hops: 1) $S$ transmits to the UAV in the first hop, denoted as $t_1$, and 2) the UAV relays the signal to $D$ in the second hop, denoted as $t_2$, using amplify-and-forward (AF)
technique. We assume that all nodes are equipped with a single transmit/receive antenna and that they operate in a half-duplex mode. Moreover, we assume that the UAV is on a mission and travels between start and destination locations while assisting the $S-D$ communication.

![Figure 1: System Model.](image)

Based on Fig. 1, the UAV’s mission has a duration $T$, discretized into $N$ time slots, where $\mathcal{N} = \{1, \cdots, N\}$, and each time slot with a time duration of $\Delta t = T/N$, such that $\Delta t$ is small enough so that the UAV location within $\Delta t$ is assumed to be unchanged. We assume that each time slot is divided into two hops that correspond to two-hop AF communications.

For clarity, the 2-dimensional (2D) Cartesian coordinate system is used, where $\mathbf{q}_D[n] = [q_{RX}[n], q_{RX}[n]]$ denotes the horizontal position of the UAV at the $n$th time slot, assuming its mission operated at a constant altitude $H$. Since the UAV’s mission is executed between dock stations, the UAV’s initial and final locations are preset as $\mathbf{q}_D[0] = \mathbf{q}_0$ and $\mathbf{q}_D[N] = \mathbf{q}_F$. Typically, the on-ground mobile user and eavesdropper move slowly with respect to the UAV’s speed; consequently, their locations can be assumed to be static and are denoted by $\mathbf{q}_D = [q_{D0}, q_{D0}]$ and $\mathbf{q}_E = [q_{E0}, q_{E0}]$, while the transmitter’s location is given by $\mathbf{q}_S = [q_{Sx}, q_{Sy}]$. Finally, the maximum speed of the UAV is bounded by $V_{\text{max}}$, thus, the maximum displacement of the UAV between two successive time slots is $d_{\text{max}} = V_{\text{max}} \Delta t$. Accordingly, over the flight period $T$, all segments connecting $N$ discrete locations $\mathbf{q}[n], n \in \mathcal{N}$, approximate the trajectory of the UAV, which needs to satisfy the following mobility constraint:

$$
\| \mathbf{q}[n+1] - \mathbf{q}[n] \| \leq d_{\text{max}}, \forall n \in \mathcal{N}\backslash\{N\}
$$

where $\| . \|$ is the Euclidean norm.

2.2. Channel Model

Communications of $S-R$, $R-D$, and $R-E$ are over air-to-ground (A2G) channels, which include both line-of-sight (LoS) and non-LoS (NLoS) path loss components [25].

However, according to [26, 27], A2G is dominated by LoS and follows a quasi-static block fading such that small-scale fading effects can be neglected, i.e., the channel gain $g_{XY}$ depends on only the distance $d_{XY}[n]$, where $XY \in \{SR, RE, RD\}$ and $d_{XY}[n]$ represents the distance between nodes $X$ and $Y$ at time slot $n$, i.e., $d_{XY}[n] = \sqrt{d^2 + \| \mathbf{q}[n] - \mathbf{q}_{Z} \|^2}$, where $Z \in \{S, E, D\}$. Therefore, the A2G channel gain at the time slot $n$ can be written as

$$
g_{XY}[n] = \beta_0 (d_{XY}[n])^{-\alpha} = \frac{\beta_0}{d_{XY}[n]^{\alpha}}, \quad (2)
$$

where $\beta_0$ is the channel gain at reference distance $d_0 = 1$ m.

As shown in Fig. 1, communications between $S$ and $D$ via $R$ occur in two phases $t_1$ and $t_2$. However, due to the broadcasting nature of the radio signal, an eavesdropper $E$ may intercept the transmitted signal. Let $x_S$ be the signal transmitted by $S$ with power $\mathbb{E}[|x_S|^2] = 1$, where $\mathbb{E}[.]$ is the expectation operator. To protect the legitimate signal of $S$, we use the MTD approach in designing the transmit signal. Specifically, signal $x_S$ is composed of the legitimate signal $x_L$ and artificial noise (AN) signal $x_F$ such that

$$
x_S[n] = \sqrt{\alpha_L[n]} x_L[n] + \sqrt{\alpha_F[n]} x_F[n],
$$

where $\alpha_L[n] \in [0,1]$ and $\alpha_F[n] = 1 - \alpha_L[n]$ are the ratios of transmit power allocated to the legitimate and AN signals, respectively, and $\mathbb{E}[|x_L|^2] = \mathbb{E}[|x_F|^2] = 1$. Additionally, we assume that both signals ($x_L$ and $x_F$) are identical and independent meaning that they have the same statistical properties and are not correlated. Subsequently, the signal received by the UAV at a given time slot $n$ can be expressed as follows:

$$
y[k][n] = \sqrt{P_S} g_{SR}[n] x_S[n] + m_R[n],
$$

where $P_S$ is the transmit power of $S$ and $m_R$ is the complex additive white Gaussian noise (AWGN) received at the UAV, with power $\sigma_0^2$.

The UAV applies AF to the received signal and sends it to $D$. The amplification factor, $\beta_A$, at each time slot $n$ is determined by the UAV’s power constraint as follows:

$$
\beta_A[n] = \sqrt{\frac{P_R[n]}{P_S g_{SR}[n] + \sigma_0^2}}
$$

Hence, the received signals at $D$ and $E$ can be given by:

$$
y_D[n] = \beta_A[n] y_R[n] \sqrt{P_R}[n] g_{RD}[n] + m_D[n]$$
$$+ \beta_A[n] \sqrt{\alpha_L[n]} P_R[n] P_S g_{SR}[n] g_{RD}[n] x_L[n]$$

Legitimate signal
$$+ \beta_A[n] \sqrt{\alpha_F[n]} P_R[n] P_S g_{SR}[n] g_{RD}[n] x_F[n]$$

AN signal
$$+ \beta_A[n] \sqrt{P_R[n] g_{RD}[n]} m_R[n] + m_D[n]$$

Noise
and

\[ y_E[n] = \beta_E[n] y_E[n] \sqrt{P_R[n] g_{RE}[n]} + m_E \]
\[ = \beta_E[n] \sqrt{\alpha_L[n] P_R[n] P_S g_{SR}[n] g_{RE}[n]} x_L[n] \]

\text{Legitimate signal}
\[ + \beta_E[n] \sqrt{\sigma_f[n] P_R[n] P_S g_{SR}[n] g_{RE}[n]} x_f[n] \]

\text{AN signal}
\[ + \beta_E[n] \sqrt{P_R[n] g_{RE}[n]} m_E[n] + m_E[n] \]

(7)

where \( P_R[n] \) is the transmit power of the UAV at time slot \( n \), and \( m_D \) and \( m_E \) are the AWGNs received at \( D \) and \( E \) with a noise power \( \sigma_0^2 \). The transmit power level \( [P_R[n]] \) is constrained by the limitations of both average power, \( P_{\text{avg}} \), and peak power, \( P_{\text{peak}} \), \( \forall n \in N \), which can be expressed as

\[ \frac{1}{N} \sum_{n=1}^{N} P_R[n] \leq P_{\text{avg}}, \quad (8) \]

and

\[ 0 \leq P_R[n] \leq P_{\text{peak}}, \quad \forall n \in N. \quad (9) \]

For (8)-(9) to be non-trivial, we assume that \( P_{\text{avg}} < P_{\text{peak}} \). Equivalently, (8) can be rewritten as

\[ \sum_{n=1}^{N} P_R[n] \leq \bar{P} \]

(10)

where \( \bar{P} = NP_{\text{avg}} \) represents the total power available during the whole mission.

Assuming that \( D \) knows \( P_S, \alpha_L[n], \) and \( \sigma_0^2 \) [15], it can explicitly remove the AN from the received signal, thus the signal-to-interference-plus-noise ratio (SINR) expression for the legitimate signal:

\[ \gamma_D[n] = \frac{\beta_E^2 \alpha_L[n] P_R[n] P_S g_{SR}[n] g_{RE}[n]}{\beta_E^2 P_R[n] g_{RO}[n] \sigma_0^2 + \sigma_0^2}. \quad (11) \]

It is worth noting that the cancellation of AN at \( D \) can never be perfect in reality, yet the residual interference due to the imperfect cancellation of AN could be included in \( \sigma_0^2 \).

On the other hand, the eavesdropper has no information about the AN and cannot decode it successfully through a simple receiver architecture. Consequently, the AN term, as introduced in (7), cannot be removed. This leads to the received SINR at \( E \) being written as

\[ \gamma_E[n] = \frac{\beta_E^2 \alpha_L[n] P_R[n] P_S g_{SR}[n] g_{RE}[n]}{\beta_E^2 (P_S \sigma_f[n] P_R[n] g_{SR}[n] g_{RE}[n] + P_R[n] g_{RE}[n] \sigma_0^2) + \sigma_0^2}. \quad (12) \]

According to Shannon’s capacity [28] and using (11) and (12), the achievable secrecy rate in bits/seconds/hertz (bps/Hz) for the legitimate user \( D \) at a given time slot \( n \) can be expressed as

\[ R_D[n] = \left[ \log(1 + \gamma_D[n]) - \log(1 + \gamma_E[n]) \right]^+, \]

\[ = \left[ R_0[n] - R_E[n] \right]^+ \quad (13) \]

where the operator \( [.]^+ = \max(.,0) \).

Without loss of generality, and based on Wyner’s theory, if the legitimate receiver’s channel falls below the wiretap (i.e., eavesdropping) channel, then the eavesdropper \( E \) succeeds in intercepting the legitimate data. In other words, when the secrecy rate \( R_D[n] - R_E[n] \) falls below zero, the probability that an eavesdropper can intercept meaningful data becomes non-null and is written as

\[ P_{\text{ins}}[n] = Pr[R_D[n] - R_E[n] < 0]. \quad (14) \]

3. Problem Formulation

In this section, we formulate the spatiotemporal diversification optimization problem, aiming to maximize the secrecy rate of the UAV-assisted network by jointly optimizing the UAV trajectory and the MTD-based power allocation.

Let \( P_R = [P_R[n]]_{n=1}^{N} \) be the sequence of transmit powers of the UAV during its mission, i.e., during \( N \) time slots, while \( \alpha = [\alpha_L[n]]_{n=1}^{N} \) is the associated MTD power splitting at the transmitter between legitimate and AN signals. Also, we define \( Q_R = [q_R[n]]_{n=1}^{N} \) the UAV trajectory during its mission.

The main objective is to maximize the average secrecy rate (ASR) for the proposed system, denoted as \( \bar{R}_e \), by jointly optimizing \( P_R, \alpha \) and \( Q_R \). This optimization is subject to the constraints imposed by the mobility of the UAV and the secrecy quality of service. Thus, the optimization problem can be formulated as follows:

maximize \( \bar{R}_e = \frac{1}{N} \sum_{n=1}^{N} R_D[n] \)

subject to

\[ R_D[n] \geq \bar{R}_d, \quad (P1.a) \]
\[ P_{\text{avg}}[n] \leq \Delta_0, \quad (P1.b) \]
\[ q_R[n] \in Z, \quad (P1.c) \]
\[ q_R[1] = q_R[N] = q_r, \quad (P1.d) \]
\[ |q_R[n+1] - q_R[n]| \leq d_{\text{max}}, \forall n < N, \quad (P1.e) \]
\[ \sum_{n=1}^{N} P_R[n] \leq \bar{P}, \quad (P1.f) \]
\[ 0 \leq P_R[n] \leq P_{\text{peak}}, \forall n = 1, \ldots, N, \quad (P1.g) \]
\[ 0 \leq \alpha_L[n] \leq 1, \forall n = 1, \ldots, N \leq 1, \quad (P1.h) \]
\[ 0 \leq \beta_E[n] \leq 1, \forall n = 1, \ldots, N, \quad (P1.i) \]
\[ (2\beta_E[n] - 1)(\gamma_E[n] - \gamma_D[n]) \geq 0, \forall n = 1, \ldots, N \quad (P1.j) \]

where \( \beta_E[n] \) is a binary indicator that characterizes the strong-weak roles of the eavesdropper in comparison with the legitimate receiver. Specifically, it is given by

\[ \beta_E[n] = \begin{cases} 1, & \text{if } \gamma_E \geq \gamma_D \\ 0, & \text{otherwise.} \end{cases} \quad (16) \]

Based on [3], (16) can be expressed as (P1.j).

In (P1), the first two constraints (P1.a) and (P1.b) ensure that the received data rate at the legitimate user is higher than
a predefined threshold \( R_{th} \), and the intercept probability at the eavesdropper is lower than a predefined value \( \Delta_{th} \), respectively. Constraint (P1.c) guarantees that the UAV’s trajectory is always within the mission’s geographical limits, defined by area \( \mathcal{Z} \). In addition, (P1.d)-(P1.e) represent the kinetic conditions for the UAV trajectory and mission, while constraints (P1.f)-(P1.h) are employed to limit the MTD and UAV relay power allocation. Finally, (P1.i)-(P1.j) reflect on the relationship between channel quality and MTD.

Note that the objective function of (P1) is non-smooth due to the definition of \( R_{L}[n] \). Moreover, this problem is non-convex, coupled with multiple variables with non-convex objective functions and constraints. Consequently, solving (P1) is generally difficult. To tackle this issue, we reformulate (P1) as a constrained Markov decision process (CMDP) and then solve it using a novel DRL-based method [29, 30].

4. Proposed DRL-based Approach for Joint MTD and Trajectory Design in UAV-Assisted Communications

In this section, we start by reformulating problem (P1) into a CMDP. Subsequently, we detail our proposed DRL-based approach to solve it. For notation brevity, the time slot indication \( n \) is omitted in the following unless otherwise stated.

4.1. Constrained Markov Decision Process Modeling

A tuple \( < \mathcal{S}, \mathcal{A}, \mathcal{R}, \pi > \) is defined to model the MDP, where \( \mathcal{S} \) represents a set of states, \( \mathcal{A} \) set of actions, \( \mathcal{R} := \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R} \) is the reward function for accepting action \( a \) in state \( s \). Also, \( \pi \) denotes the policy of mapping from state \( s \in \mathcal{S} \) to action \( a \in \mathcal{A} \).

As shown in Fig. 2, the UAV acts as an individual agent and uses a DDQN to learn how to interact with the environment. Specifically, the UAV can be trained to associate a certain state, consisting of the UAV’s transmit power \( P_R \), MTD-based power allocation factor \( \alpha_L \), and UAV trajectory \( q_L \), with an appropriate action, denoted as \( a \) to improve its reward. It is worth noting that \( \alpha_L \) indicates not only the MTD-based power allocation but also the overall system security and how much manipulations should be induced according to the current power allocation. Therefore, state, action, reward, and policy can be expressed in the CMDP as follows:

4.1.1. State \( s^{(k)} \)

The system’s state at each iteration \( k \) consists of the following:

- Data rate at the legitimate user, \( R_D \),
- UAV location, \( q_L \),
- Legitimate user, and eavesdropper’s locations, \( q_D \) and \( q_E \), respectively,
- UAV’s transmit power, \( P_R \),
- MTD-based power allocation, \( \alpha_L \),


- Strong-weak variable \( \beta_L \),
- Intercept probability, \( P_{int} \).

Initially, the system state, \( \mathcal{S}_0 \), is a tuple consisting of \( [R_{th}, \quad q_L, \quad q_D, \quad q_E, \quad P_R, \quad \alpha_L, \quad \beta_L, \quad P_{max}] \), where it is obtained when the UAV relay is in its initial location \( q_L \). The final state \( \mathcal{S}_t \) is when convergence is achieved.

4.1.2. Action \( a^{(k)} \)

Based on the current state of the system, an action must be selected for execution. The action set is defined as follows:

\[
\mathcal{A}^{(k)} = \{ a_{Pa}^{(k)}, a_{q_L}^{(k)}, a_{q_E}^{(k)} \}
\]

where

- \( a_{Pa}^{(k)} = [a_{Pa}^{(k)}] \) with size \( 1 \times N \), where \( a_{Pa}^{(k)} \) is defined as an increase or decrease of the UAV’s transmit power \( P_R[n] \) by a prefixed step size \( \delta \), such that \( 0 \leq P_R[n] = P_R[n-1]+\delta \leq P_{peak} \),

- \( a_{q_L}^{(k)} = [a_{q_L}^{(k)}] \) with size \( 1 \times N \), where \( a_{q_L}^{(k)} \) represents the adjustment in the penalty associated with the allocated power for the real data \( \alpha_L[n] \). This adjustment is determined by the change, either an increment or decrement, denoted by the fixed step size \( \beta_P \), such that \( 0 \leq \alpha_L[n] = \alpha_L[n-1] + \beta_P \leq 1 \). The inclusion of \( \beta_P \) is motivated by situations where the UAV may be located far from \( D \), but close to \( E \). In such a situation, \( R_L[n] \) may exceed \( R_D[n] \), thus rendering the worst-case secrecy rate almost null, even if \( P_R[n] = P_{peak} \) and \( \alpha_L[n] = 1 \). Conversely, if \( R_E[n] \) is lower than \( R_D[n] \), the DRL agent can appropriately select the transmit power to maximize the secrecy rate. To account for these conditions and ensure optimal transmission power selection throughout the UAV’s trajectory, different penalties have been designed as follows [23]:

\[
\beta_P[n] = \begin{cases} 
-0.1, & \text{if } \beta_L[n] = 1 \\
0, & \text{otherwise.}
\end{cases}
\] (17)

- \( a_{q_E}^{(k)} = [a_{q_E}^{(k)}] \) with size \( 1 \times N \), where \( a_{q_E}^{(k)} \) is to adjust the UAV’s flight direction, allowing it to move forward, backward, leftward, or rightward. This action aims to confuse potential attackers by dynamically altering the system’s configuration, thereby enhancing overall system security and optimizing its performance. Specifically, the UAV coordinates are adjusted by a step size, such that \( q_{Rx}[n] = q_{Rx}[n-1]+\delta_{q_x} \) and \( q_{Ry}[n] = q_{Ry}[n-1]+\delta_{q_y} \), with respect to \( d_{max} \) and \( \mathcal{Z} \).

4.1.3. Reward \( r^{(k)} \)

The reward \( r^{(k)} \), which is a function of the ASR and system penalties, is returned when the agent takes action \( \mathcal{A}^{(k)} \) for the environment state \( \mathcal{S}^{(k)} \). To favor the UAV traveling close to the legitimate user and thus enhance its secrecy, we introduce
a reward/penalty variable, denoted $r_{RD}$, which relates to the distance between the UAV and the legitimate user. It is expressed by

$$r_{RD} = \begin{cases} -5, & \text{if } d_{RD} > d_{RE}, \\ +5, & \text{otherwise}, \end{cases}$$

where $d_{RD}$ is the normalized distance between $R$ and $D$. A positive reward of 5 is granted when $d_{RD}$ is below $d_{RE}$, and a negative reward of −5 is assigned if the distance exceeds the distance between the UAV and the potential attacker, $d_{RE}$. The selection of these thresholds is done manually by trial and error, depending on the desired proximity of the UAV to the legitimate user.

Using (17) and (18), the reward function can be given as

$$r^{(k)}(s, a) = \frac{1}{N} \sum_{n=1}^{N} (R[n] + r_{DR}[n] + \beta_P[n]),$$

where the superscript $k$ indicates the number of interaction steps.

### 4.2. Proposed DRL-based Method

The DQN algorithm tends to overestimate the action values when approximating highly complex functions. As a solution, the introduction of a DDQN-based approach can significantly improve performance compared to DQN [31]. Particularly, the DDQN algorithm utilizes double networks: an online/estimation network $Q(S, A; \theta)$ and a target network $\hat{Q}(S, A; \hat{\theta})$, where $\theta$ and $\hat{\theta}$ represent the respective network parameters. These dual networks decouple action generation and Q-value evaluation, which increases learning stability and overcomes over-optimism in large-scale problems.

Instead of training all the data in each iteration, DDQN employs a more efficient method by evaluating the gradients with a random subset of the experience pool referred to as “mini-batch” and applying the stochastic gradient descent method to update the network parameters. It leverages a replay buffer (RB), a fundamental technique in DDQN that stores transitions in an experience pool denoted by $e = (S, A, r, \hat{S})$, where $\hat{S}$ represents the next state. The accumulated experience, including actions, state transitions, and rewards received after the interaction, is stored in the replay memory, $\mathcal{D}$, of size $M$. This buffer, $\mathcal{D} = (e^{(1)}, e^{(2)}, \ldots, e^{(M)})$, is randomly accessed for further computation of the loss function. Alongside the mini-batch, the replay buffer effectively mitigates the high correlation with successive updates.

According to Fig. 2, the process of the DDQN-based algorithm can be summarized as follows:

1. The online network determines the next action by selecting the action that maximizes the Q-value for each state-action pair, employing a stochastic gradient descent (SGD) policy, i.e., finding $\arg \max_A Q(S, A; \theta)$.

2. The target network is responsible for evaluating the actions by calculating the temporal difference (TD)-error. This error is determined as the mean square error (MSE) across $M'$ randomly selected samples within a mini-batch. The calculation can be expressed as follows [7]:

$$\mathcal{L} = E_M \left( r + \Gamma \hat{Q}(\hat{S}, \arg \max_A Q(\hat{S}, A; \hat{\theta})) - Q(S, A; \theta) \right)^2,$$

where term $A_1$ represents the expected Q-value of the next state-action pair, discounted by a factor $\Gamma$ and approximated by the target network based on its current policy $\hat{\theta}$. The selection of the next action is performed by the online network, denoted as $\arg \max_A Q(S, A; \theta)$. On the other hand, term $A_2$ corresponds to the predicted Q-value of the current state-action pair, calculated by the deep neural network with parameters $\theta$. The mathematical expectation over the mini-batch is denoted as $E_M$. It is worth noting that the action selection and evaluation are carried out independently to mitigate the risk of overestimating the Q-value, as recommended by [31].

3. The online network parameters $\theta$ can be iteratively updated
by the SGD of the TD-error $\nabla_\theta L$, i.e.,

$$\theta^{k+1} = \theta^k + \epsilon_\theta \nabla_\theta L,$$

(21)

where $\epsilon_\theta$ is the step size for the parameter update and $\nabla_\theta L$ is the gradient of $L$ with respect to $\theta$, which can be easily obtained by the backpropagation algorithm.

4. The $\hat{\theta}$ parameters of the target network are copied every $\tau$ iteration from the online network, i.e., $\hat{\theta}^k = \theta^{k-\tau}$. Therefore, the power allocation factor and the transmitted power level, given each UAV trajectory’s time slot, can be determined once the online network is well-trained.

The detailed DDQN-based method is presented in Algorithm 1.

**Algorithm 1: Proposed DDQN-based approach**

**Input:** $D = D_1, \ldots, D_N$, UAV transmission powers $P_R$, UAV trajectory $q$, and MTD-power allocation factors $\alpha_L$

**Output:** Return best solution $(S^{(k)}, A^{(k)}, r^{(k)})$

/* Initialization */
1. Initialize $\Gamma$
2. Initialize the Q-network with random weights $\theta$
3. Initialize the target Q-network parameters by $\theta = \hat{\theta}$
4. Initialize the first state $S_0$
5. Initialize Gaussian noise $\mathbb{N}$

/* While no convergence or not aborted over each training sample */
for iteration $k = 1, \ldots, K$

6. Select action $a \in A$ by
   $$A^{(k)} = \arg\max_{A} Q(S^{(k)}, A^{(k)}) + \mathbb{N}$$
7. Interact with the environment given the current configuration
8. Calculate $P_m$ from (14) and $R_t$ from (13)
9. Calculate reward $r^{(k)}$ using (19), obtain next state $S^{(k+1)}$, and store $e = (S, A, r^{(k)}, S')$ in RB
10. Sample a random mini-batch of $M'$ from RB
11. Calculate Q-value $Q(S^{(k)}, A^{(k)}|a^{(k)})$ for sampled pairs in the mini-batch and calculate $A_2$ in (20)
12. Select action and calculate $A_1$ in (20)
13. Calculate TD-error with (20)
14. Update the online network parameters $\theta$ to minimize TD-error by $\epsilon$-gradient
15. Update the target network parameters $\hat{\theta}$ every $\tau$ iterations

end

4.3. Convergence and Complexity Analysis

**Lemma 1.** Algorithm 1 can converge to a local suboptimal solution, at least in a finite number of iterations.

The original problem (P1) is solved iteratively by applying the DDQN algorithm to obtain the suboptimal solution with the initial feasible solutions. The solution obtained in each iteration is used as input for feasible points for the next iteration.

Let $\eta(Q^k, P^k_R, A^k_R)$ be the solution of the original objective function in the $k^{th}$ episode. The proposed DDQN will output a better solution $(Q^{k+1}, P^k_R, A^k_R)$ satisfying:

$$\eta(Q^k, P^k_R, A^k_R) \leq \eta(Q^{k+1}, P^k_R, A^k_R)$$

(22)

It is worth mentioning that the objective function is non-decreasing after each iteration. Given the constraints, the maximum ASR is upper-bounded by a finite value. Consequently, Algorithm 1 is guaranteed to converge to at least a local optimum, as shown in Fig. 3.

The computational complexity of DDQN is measured in terms of floating-point operations (FLOPs) [32]. As previously mentioned, each agent comprises two isomorphic neural networks: the target network and the online network, both constructed as basic, fully connected multilayer perceptrons (MLPs). The configuration of the neural network is shown in Table 1.

Let $e_i$ represent the number of neurons in the $i^{th}$ layer of the online network, where $i \in \{0, \cdots, I\}$ and $I$ is the total number of layers. For a fully connected MLP layer with $e_i$ neurons as input and $e_{i+1}$ neurons as output, the computational cost in terms of FLOPs for the dot product from the $i^{th}$ to the $(i+1)^{th}$ layer is given by $(2e_i - 1) \times e_{i+1}$. This implies that the operation must be multiplied $e_i$ times and added $(e_{i+1} - 1)$ times for each neuron.

Let $\kappa$ be the corresponding FLOP parameter determined by the type of activation function. For example, the sigmoid function has $\kappa_{\text{Sigmoid}} = 4$ FLOPs since the function $\delta(z) = 1/(1 + e^{-z})$ involves four mathematical operations: Division, summation, exponentiation, and subtraction, each of which requires one FLOP. Similarly, $\kappa_{\text{Relu}} = 1$ FLOP. Therefore, according to [7], the computational complexity of DDQN can
be calculated as
\[
2 \sum_{i=0}^{I-1} ((2c_i - 1)e_{i+1} + \kappa_i e_i) = \mathcal{O}(\sum_{i=0}^{I-1} e_i e_{i+1}) = \mathcal{O}(32N^2 + 40N) 	imes I \times 2\mathcal{H} \times \mathcal{E}\mathcal{T}
\]
where \(\mathcal{E}\) is the number of steps per episode, \(\mathcal{H}\) is the number of hidden layers, and \(\mathcal{T}\) is the total number of episodes.

5. Simulation Results

5.1. Simulation Setup

In this section, we conduct experiments to assess the performances of our proposed approach for the joint optimization of the UAV relay trajectory, transmit power, and AN power allocation. We denote our proposed MTD-based framework as "T-OPT-PC-AN", i.e., trajectory optimization with transmit power and AN power allocation control. For the sake of comparison, we consider benchmark schemes that do not optimize the UAV trajectory and/or power allocation, as follows:

- T-OPT-PC: Trajectory optimization with transmit power control;
- T-OPT-AN: Trajectory optimization with power allocation control;
- BET-AN: Best-effort trajectory design with AN power allocation control;
- BET-PC: Best-effort trajectory design with transmit power control;
- EST-OPT-PC: Exhaustive search trajectory optimization with transmit power control;
- EST-OPT-AN: Exhaustive search trajectory optimization with AN power allocation control;
- RAT-PC: Random trajectory design with transmit power control;
- RAT-AN: Random trajectory design with AN power allocation control.

The T-OPT-AN algorithm is designed to determine the optimal UAV trajectory with an optimal AN injection factor to confuse potential attackers. In this case, the transmission power is uniformly distributed over time, specifically \(P_k[n] = P_{\text{avg}}, \forall n \in \mathcal{N}\). On the other hand, the T-OPT-PC algorithm focuses on crafting the best UAV trajectory with optimal transmit power levels, where no power allocation is employed, i.e., \(\alpha_T = 1\). Both of these algorithms exhibit a complexity of \(\mathcal{O}(18N^2 + 24N) \times N_{\text{iter}}\), where \(N_{\text{iter}}\) denotes the number of iterations needed to reach convergence.

In the heuristic trajectory designs of the BET-PC and BET-AN algorithms, the UAV initially accelerates at its maximum speed, heading directly to the point above the receiver GN. If time permits, the UAV hovers at that location for as long as possible before accelerating again at maximum speed to reach its final location \(q_f\) by the end of the last time slot. In cases where the UAV lacks sufficient time to reach the location above the receiver GN, it executes a turn at a specific midpoint and subsequently accelerates at maximum speed toward its final destination \(q_f\). In this trajectory, BET-PC optimizes the transmit power of the UAV, while BET-AN evenly distributes the transmit power over time, \(P_k[n] = P_{\text{avg}}, \forall n \in \mathcal{N}\). The schemes BET-PC and BET-AN achieve the same complexity \(\mathcal{O}(N)\) [11]. When using the exhaustive search trajectory algorithms EST-PA and EST-AN, the approach involves exploring all possible trajectories and identifying the shortest Hamiltonian cycle path while adhering to the system’s constraints [33]. Within the feasible trajectory set, EST-PA maximizes the ASR by jointly optimizing the UAV trajectory and the transmit power of the UAV. On the other hand, EST-AN maximizes the ASR by distributing the transmit power evenly over time, maintaining \(P_k[n] = P_{\text{avg}}, \forall n \in \mathcal{N}\). Both EST-PC and EST-AN algorithms demonstrate a complexity of \(\mathcal{O}(N!)\) [34]. It is worth mentioning that the exhaustive search and heuristic algorithms are based on the traveling salesman problem with time window (TSTPW) method. While the exhaustive search algorithm provides global optimality, its exponential computation complexity might limit its applicability in practical applications. In such a case, the heuristic algorithm with lower complexity can balance the performance-complexity trade-off. Although exhaustive search performs better than heuristic approaches, its long completion time provides enough time for the attacker to compromise the transmission, thus becoming suboptimal from a security perspective. Finally, for the random trajectory designs, RAT-PC and RAT-AN, the UAV follow any randomly designed route with any selected power allocation ratio given the normalized transmit power. The complexity of these last algorithms is determined as \(\mathcal{O}(1)\).

We assume a source GN that communicates with a receive GN via a UAV relay in the presence of an eavesdropper, all being deployed within an area of \([1000 \text{ m} \times 1000 \text{ m}]\). The coordinates of the source GN, destination GN, and the eavesdropper are set to \((0, 0, 0)\) m, \((200, 0, 0)\) m, and \((400, 0, 0)\) m respectively, and the UAV’s flying altitude is set to \(H = 100\) m. The maximum speed of the UAV is \(v_{\text{max}} = 10\) m/s. The flight period \(T\) is divided into several time slots of equal length of \(\Delta t = 0.5\) s. The communication bandwidth is set to 1 MHz with a carrier frequency of 1 GHz, while the noise power spectrum density is \(-110\) dBm/Hz. The threshold for the peak transmit power is defined as \(P_{\text{peak}} = 4 \times P_{\text{avg}}\). Finally, in the simulation, the predefined intercept probability threshold is set to \(\Delta_{\text{th}} = 0.05\) and the predefined ASR is defined as \(R_0 = 1.5\) Mbits/sec. For our simulations, we consider that the UAV’s initial and final locations are at \(q_i = (200, 100, 100)\) m and \(q_f = (200, -100, 100)\) m, respectively. The used parameters are summarized in Table 2.
Table 1: DDQN neural networks Set-up

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Dimension</th>
<th>Neurons Num.</th>
<th>Active Fun.</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Layer</td>
<td>Feature input</td>
<td>$</td>
<td>S</td>
<td>\times</td>
<td>S</td>
</tr>
<tr>
<td>Hidden layers, $</td>
<td>H</td>
<td>$</td>
<td>Fully connected</td>
<td>$</td>
<td>S</td>
</tr>
<tr>
<td>Output layer</td>
<td>Fully connected</td>
<td>$</td>
<td>S</td>
<td>\times</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 2: Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notation</th>
<th>Simulation value</th>
<th>Parameter</th>
<th>Notation</th>
<th>Simulation value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max UAV speed</td>
<td>$v_{\text{max}}$</td>
<td>10 m/s</td>
<td>Average UAV transmit power</td>
<td>$P_{\text{avg}}$</td>
<td>5 dBm</td>
</tr>
<tr>
<td>Peak UAV transmit power</td>
<td>$P_{\text{max}}$</td>
<td>20 dBm</td>
<td>Number of time slots</td>
<td>$T$</td>
<td>$20 : 2 : 180$</td>
</tr>
<tr>
<td>Time slot duration</td>
<td>$\Delta t$</td>
<td>0.5 s</td>
<td>Altitude of UAV</td>
<td>$H$</td>
<td>100 m</td>
</tr>
<tr>
<td>Channel power gain</td>
<td>$\beta_0$</td>
<td>$-20$ dB</td>
<td>Terrestrial pass-loss exponent</td>
<td>$\alpha$</td>
<td>2</td>
</tr>
<tr>
<td>Noise power</td>
<td>$\sigma_0^2$</td>
<td>50 dB</td>
<td>Bandwidth</td>
<td>$BW$</td>
<td>1 MHz</td>
</tr>
<tr>
<td>Size of replay buffer</td>
<td>$RB$-size</td>
<td>80</td>
<td>Discount factor</td>
<td>$\gamma$</td>
<td>0.95</td>
</tr>
</tbody>
</table>

5.2. Obtained Results

Figs. 4 and 5 illustrate the average secrecy rates (ASRs) of different algorithms as functions of the flight time $T$ under different average power values. Clearly, the ASR of any algorithm experiences an improvement with $T$. Indeed, as $T$ increases, the UAV hovers for longer at strategic locations, allowing it to enhance the ASR performance. Specifically, the proposed algorithm T-OPT-PC-AN consistently reaches the best performance, while the benchmark algorithms RAT-PC (solid blue line) and RAT-AN (dashed blue line) demonstrate the worst ASRs, as anticipated. Moreover, at very high $T$, most AN-based algorithms’ ASR increases slowly, whereas the PC-based algorithms continue to exhibit a significantly increasing trend. In the low-power regime, all algorithms except RAT-AN adhere to the predefined threshold, $R_b$, while RAT-AN starts to comply with $R_b$ starting from $T \geq 100$s.

Examining the ASR in Fig. 4b for $P_{\text{avg}} = -5$ dBm, we observe that the benchmark algorithm T-Opt-PC outperforms EST-OPT-PC and BET-PC. Initially, both EST-OPT-PC and BET-PC demonstrate similar ASR performances. However, as the flight period increases, the performance gap enlarges, resulting in BET-PC achieving an ASR approximately 28.45% higher than the EST-OPT-PC algorithm. Accordingly, we conclude that in a low-power regime, optimizing the trajectory jointly with the power allocation, in Fig. 4b, significantly improves the ASR more than with AN injection, illustrated in Fig. 4a.

In Fig. 5a, for $P_{\text{avg}} = 5$ dBm, the algorithms that consider AN injection showcase higher ASRs than those with power control and without AN, plotted in Fig. 5b. Specifically, T-Opt-AN exhibits ASR performances close to those of the proposed algorithm. Initially, for shorter flight periods, i.e., $T \leq 100$s, BET-AN outperforms EST-Opt-AN, but this trend reverses as $T$ increases. Moreover, trajectory adaptation with
increasing $T$ remains critical to improving the ASR, even using the heuristic BET and brute-force EST designs. Finally, in a high-power regime, optimizing the trajectory jointly with AN injection, as demonstrated in Fig. 5 significantly enhances the ASR more than with PC, shown in Fig. 5b.

In Fig. 6, we present the resulting intercept probability in both low-power and high-power regimes. The intercept probability results have been plotted for the different algorithms as functions of the flight time $T$. A noticeable trend emerges as the intercept probability of all algorithms consistently decreases with $T$. Particularly, our proposed algorithm, T-OPT-PC-AN, achieves the lowest $P_{\text{int}}$ values. In contrast, the benchmark algorithms RAT-PC (solid and dashed blue lines) and RAT-PC consistently exhibit suboptimal performances, which aligns with our expectations. In the low-power regime, Fig. 6a, not all algorithms respect the predefined threshold $\Delta_{th}$. For instance, RAT-PC fails to achieve the required intercept probability for any flight period value, while EST-Opt-PC (red dashed line) and BET-PC (yellow dashed line) satisfy the threshold for $T \geq 110$ s and $T \geq 90$ s, respectively. Only the proposed algorithm and T-Opt-PC consistently satisfy $\Delta_{th}$. In the high-power regime of Fig. 6b, all algorithms respect $\Delta_{th}$ except RAT-PC (solid blue line), which meets the requirement starting $T \geq 150$ s. Accordingly, T-Opt-PC-AN and T-Opt-PC can secure transmitted data for any power regime and mission time length.

Fig. 7 illustrates the ASRs and intercept probabilities of the different algorithms as functions of the average transmit power $P_{\text{avg}}$, and for fixed flight period $T = 80$ s. The proposed T-OPT-PC-AN (green line) outperforms all benchmarks in terms of ASR and $P_{\text{int}}$ for any $P_{\text{avg}}$ value. The benchmark RAT-PC (blue line) lags as the least-performing algorithm. In Fig. 7a, T-Opt-PC achieves the second best ASR, which is close to EST-Opt-PC (red line) and BET-PC (yellow line), particularly for $P_{\text{avg}} \leq 0$ dBm. In contrast, it realizes a better $P_{\text{int}}$ as shown in Fig. 7b, and both EST-Opt-PC and BET-PC do not correctly secure the communications when $P_{\text{avg}}$ is below $-8$ dBm and $-10$ dBm, respectively. Finally, T-OPT-PC-AN and T-OPT-PC consistently adhere to the $P_{\text{int}}$ constraint and achieve significantly lower intercept probabilities compared to the other benchmarks by approximately 96.7% and 93.2%, respectively.

6. Conclusion

In this work, we proposed a novel strategy to address the vulnerability of UAV-assisted communications, consisting of eavesdropping threats. Specifically, we proposed a spatiotemporal diversification-based AN injection strategy combined with power allocation and UAV trajectory optimization, aiming to protect the communication’s privacy. By jointly optimizing the UAV trajectory, transmission power splitting factor, and AN injection using a novel DRL-based method, our approach enhances the ASR while keeping the intercept probability below a preset threshold in any environmental condition. Through extensive simulations, we demonstrate the proposed algorithm’s superiority over several benchmarks, in terms of ASR and intercept probability. Key findings and observations include the following: i) Both ASR and intercept probability improve with extended flight periods $T$, with the proposed algorithm T-OPT-PC-AN demonstrating better adaptability to $T$ than the benchmarks. ii) The proposed T-OPT-PC-AN outperforms the benchmarks, in terms of ASR and intercept probability in both low- or high-transmit power regimes. iii) The proposed scheme is adaptive to any power splitting strategy between real and fake data, $\alpha_L$, thus realizing the best ASR and $P_{\text{int}}$ performances.

References
