Securing MIMO Wiretap Channel With DDPG-Based Friendly Jamming Under Non-Differentiable Channel

Bui Minh Tuan*, Diep N. Nguyen*, Nguyen Linh Trung§, Van-Dinh Nguyen†, Nguyen Van Huynh‡, Dinh Thai Hoang*, Marwan Krunz¶, Eryk Dutkiewicz*

*School of Electrical and Data Engineering, University of Technology Sydney, NSW 2007, Australia
†College of Engineering and Computer Science, VinUniversity, Vinhomes Ocean Park, Hanoi 100000, Vietnam
‡Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, UK
§AVITECH, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
¶Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ, USA

***Abstract*—6G communication systems, particularly massive Internet of Things (IoT), face critical security challenges in safeguarding transmissions against eavesdropping attacks. These challenges are exacerbated by the presence of intelligent eavesdroppers capable of exploiting impairments in wiretap channels. Traditional physical layer security (PLS) techniques, such as friendly jamming (FJ), typically rely on the differentiability and accurate availability of channel state information (CSI) to optimize performance. However, in real-world scenarios, non-differentiable channels (NDCs) resulting from hardware imperfections, mobility, and complex multi-path fading, pose significant obstacles to conventional gradient-based optimization methods. In this paper, we propose a novel deep learning-based FJ approach tailored specifically for NDC environments, where gradient-based techniques prove ineffective. Leveraging the Deep Deterministic Policy Gradient (DDPG) algorithm, our framework dynamically generates jamming signals to optimize secrecy rates while simultaneously minimizing the block error rate (BLER) at the legitimate receiver. Through extensive evaluation of realistic channel models, including both line-of-sight (LoS) and non-line-of-sight (NLoS) conditions, our proposed approach demonstrates superior security enhancements and robust performance against eavesdropping threats. The results highlight its effectiveness in securing communications under the challenging and dynamic conditions inherent to NDC environments.**

I. Introduction

The rapid expansion of 5G and 6G technologies, particularly in the massive Internet of Things (IoT) and multiple-input and multiple-output (MIMO) systems, has transformed modern communication systems, enhancing data rates and reliability. However, with these advancements come new challenges, particularly in maintaining reliable transmission and secure communication in the presence of smart eavesdroppers who can exploit wiretap channels' impairments. While traditional cryptographic solutions are effective, they are often impractical for dynamic, resource-constrained environments like 6G's aerial IoT networks [1]. These systems require lightweight and flexible methods to safeguard communications, making friendly jamming (FJ) in physical layer security (PLS) an effective solution [2]. FJ works by injecting artificial noise into the radio medium to degrades the eavesdropper's ability to intercept confidential signals while maintaining the integrity of the legitimate receiver's communication.

Despite its potential, FJ's effectiveness relies heavily on channel state information (CSI), often assumed to be perfectly known and differentiable. However, real-world wireless channels frequently exhibit non-differentiable behaviour due to factors like multipath fading, mobility, interference, and hardware imperfections. For example, the non-differentiable behavior of wireless channels can be a direct result of the frequent transitions between line-of-sight (LoS) and non-line-of-sight (NLoS) states. These transitions are common in real-world environments where obstacles intermittently block the direct signal path between the transmitter and receiver. For instance, in urban scenarios, the movement of vehicles or pedestrians can create temporary obstructions, while in indoor settings, furniture, or people moving through the space can disrupt the LoS. Environmental factors, such as signal reflections and scattering from buildings or other surfaces, further contribute to these transitions. This realistic modeling distinction between Rician and Rayleigh fading is rooted in their probability distributions [3]. These nondifferentiable channels prohibit traditional gradient-based optimization techniques, which are essential in many modern learning-based frameworks like end-to-end (E2E) autoencoder (AE) systems. Only a few works have addressed this issue, for example, [4], which proposed a gradient-free method to train AE using a Kalman cubature filter. Provisioning reliable and secure communications in NDC is hence particularly challenging, yet underexplored.

This paper introduces a novel DL approach to tackle the above challenges of NDC in wireless systems. The proposed solutions are designed to be robust and effective in practical conditions, e.g., with imperfect CSI. Specifically, we focus on line-of-sight (LoS) and non-line-of-sight (NLoS) channels, which serve as representative examples of NDC. We then propose a novel deep learning-based FJ approach explicitly designed for non-differentiable channels. Using Deep Deterministic Policy Gradient (DDPG), the framework dynamically generates jamming signals to optimize secrecy rates while minimizing the block error rate (BLER) at the legitimate receiver. Evaluated over realistic channel models that include line-of-sight (LoS) and non-line-of-sight (NLoS) conditions, our approach demonstrates enhanced security and robust performance against eavesdropping in NDC environments, offering a robust solution to secure communications under these challenging wireless conditions. Our contributions are summarized as follows:

- We propose a novel DDPG-based communication framework to ensure reliable communication in non-differentiable channels. By leveraging reinforcement learn-

ing (RL), our method overcomes the limitations of gradient-based optimization, effectively addressing NDC challenges to deliver robust communication performance.
- We extend the DDPG framework to maximize the secrecy rate by generating jamming signals to degrade the eavesdropper's ability to intercept communication while minimizing the BLER at the legitimate receiver. The proposed exhibits resilience and robustness in the most challenging wireless conditions.
- We comprehensively evaluate the proposed approach using realistic channel models. The results show that our approach significantly improves communication reliability and security against eavesdroppers compared to traditional methods under differentiable channels.

II. Background on MIMO-FJ and Channel Models

The typical MIMO-FJ configuration [5] comprises a transmitter (Alice), a receiver (Bob), and an eavesdropper (Eve), each with N_t, N_r, and N_e antennas, respectively. At time slot k, the Tx-Rx and Tx-Eve channel matrices are $\mathbf{H}_k \in \mathbb{C}^{N_t \mathrm{x} N_r}$ and $\mathbf{G}_k \in \mathbb{C}^{N_t \mathrm{x} N_e}$. Assuming perfect CSI, the matrices $\mathbf{H}_k$ and $\mathbf{G}_k$ remain constant over a transmission block. Let $\mathbf{s}_k$ represent the intended message, and $\mathbf{w}_k$ denote the FJ signal designed to satisfy $\mathbf{H}_k^\dagger \mathbf{w}_k = 0$. With transmitted signal $\mathbf{x}_k = \mathbf{s}_k + \mathbf{w}_k$, the received signals are:

$$\mathbf{y}_k = \mathbf{H}_k^\dagger \mathbf{s}_k + \mathbf{n}_b, \tag{1}$$

$$\mathbf{z}_k = \mathbf{G}_k^\dagger \mathbf{s}_k + \mathbf{G}_k^\dagger \mathbf{w}_k + \mathbf{n}_e, \tag{2}$$

where $\mathbf{n}_b \sim \mathcal{CN}(0, \sigma_b^2)$ and $\mathbf{n}_e \sim \mathcal{CN}(0, \sigma_e^2)$ are AWGN at Rx and Eve, respectively. $\mathbf{w}_k$ is set as $\mathbf{Z}_k \mathbf{v}_k$, with $\mathbf{Z}_k$ as the orthogonal basis of the null space of $\mathbf{H}_k^\dagger$, and elements of $\mathbf{v}_k$ are i.i.d. Gaussian with variance σ_v^2. Eve's noise covariance is then formulated as $\mathbf{K}_k = (\mathbf{G}_k^\dagger \mathbf{Z}_k^\dagger \mathbf{Z}_k \mathbf{G}_k)\sigma_v^2 + \mathbf{I}_{N_e}\sigma_e^2$. The secrecy rate R_k^s is:

$$\begin{aligned} R_k^s &= \left[\log(1+\mathrm{SINR_B}) - \log(1+\mathrm{SINR_E})\right]^+ \\ &= \left[\log\det\left(\mathbf{I} + \mathbf{H}_k^\dagger \mathbf{Q}_s \mathbf{H}_k\right) - \log\frac{\det\left(\mathbf{K}_k + \mathbf{G}_k^\dagger \mathbf{Q}_s \mathbf{G}_k\right)}{\det(\mathbf{K}_k)}\right]^+, \end{aligned}$$

where $\mathbf{Q}_s = \mathbf{E}[\mathbf{s}_k \mathbf{s}_k^\dagger]$ and $[x]^+ = \max(0, x)$. Since Eve's CSI is unavailable at Tx, we focus on maximizing the first term using SVD, with $\mathbf{H}_k^\dagger = \mathbf{U}_k \mathbf{\Gamma}_k \mathbf{V}_k^\dagger$. After precoding $\mathbf{r}_k = \mathbf{V}_k^\dagger \mathbf{s}_k$, the secrecy rate R_k^s is rewritten as:

$$R_k^s = \left[\log\det(\mathbf{I} + \mathbf{\Gamma}_k \mathbf{Q}_r \mathbf{\Gamma}_k^\dagger) - \log\frac{\det(\mathbf{K}_k + \mathbf{F}_k)}{\det(\mathbf{K}_k)}\right]^+, \tag{3}$$

Our objective is to maximize the average secrecy rate subject to the power constraint at the Tx, which is mathematically formulated as in (4), where $\mathbf{F}_k = \mathbf{G}_k^\dagger \mathbf{V}_k^\dagger \mathbf{Q}_r \mathbf{V}_k \mathbf{G}_k$, and $\mathbf{Q}_r = \mathbb{E}[\mathbf{r}_k \mathbf{r}_k^\dagger] = \mathrm{diag}(\sigma_{r,1}^2, \sigma_{r,2}^2, ..., \sigma_{r,N_t}^2)$, $\sigma_{r,i}^2$ is derived by applying water filling algorithm with power constraint $P_{\mathrm{info}} \leq P$. The objective is to maximize the average secrecy rate $\bar{R}$ over multiple channel realizations with power constraint $\mathrm{Tr}(\mathbb{E}[\mathbf{x}_k \mathbf{x}_k^\dagger]) \leq P$, reformulated as $\mathrm{Tr}(\mathbf{V}_k^\dagger \mathbf{Q}_r \mathbf{V}_k + N_{\mathrm{FJ}}\sigma_v^2 \mathbf{I}_{N_t}) \leq P$, where N_{FJ} is the number of dimensions for FJ.

$$\begin{aligned} \bar{R} \doteq \max_{\mathrm{Tr}(\mathbb{E}[\mathbf{x}_k \mathbf{x}_k^\dagger]) \leq P} \mathbb{E}\Big[&\log\det\left(\mathbf{I} + \mathbf{\Gamma}_k \mathbf{Q}_r \mathbf{\Gamma}_k^\dagger\right) \\ &- \log\frac{\det(\mathbf{K}_k + \mathbf{F}_k)}{\det(\mathbf{K}_k)}\Big]. \end{aligned} \tag{4}$$

A. Existing Solutions

To address the problem in (4), two main strategies can be considered: (i) exhaustive search and (ii) gradient-based techniques. While exhaustive search methods can achieve optimal outcomes, they are highly computationally demanding [5], [6]. Conversely, gradient-based techniques, such as steepest descent (or gradient descent), are commonly applied for optimization within the deep neural network (DNN) frameworks [7], [8]. Other conventional gradient-based strategies generally presume a differentiable channel model, including alternating optimization and semidefinite relaxation [9].

In the DL-based FJ framework, the transmitter encodes a message m into channel symbols $\mathbf{x}_k$ using a mapping function $f_{\omega_k}^{(T)}: \mathcal{M} \to \mathbb{R}^{2N}$, where N represents the count of complex channel uses, and ω_k denotes the associated parameters. The receiver, defined by $f_{\omega_R}^{(R)}: \mathbb{R}^{2N} \to \{\mathbf{p} \in \mathbb{R}_+^M \mid \sum_{i=1}^M p_i = 1\}$, converts the received signal $\mathbf{y}_k$ into a probability distribution $\mathbf{p}$ across messages. The loss function $\mathcal{L}$, minimized through gradient descent, is specified as:

$$\mathcal{L}(\omega_k, \omega_R) = \mathbb{E}_m\left[\int l(f_{\omega_R}^{(R)}(\mathbf{y}_k), m) p(\mathbf{y}_k \mid f_{\omega_k}^{(T)}(m)) d\mathbf{y}_k\right],$$

$$\nabla_{(\omega_R, \omega_k)}\mathcal{L} = \left[\left(\nabla_{\omega_R}\mathcal{L}\right)^T, \left(\nabla_{\omega_k}\mathcal{L}\right)^T\right]^T, \tag{5}$$

where $\nabla_{\omega_R}\mathcal{L}$ and $\nabla_{\omega_k}\mathcal{L}$ are the gradients for the receiver and transmitter, respectively. The gradient for the receiver parameters ω_R, with S is the batch size and $\{m^{(i)}, \mathbf{y}_k^{(i)}\}$ are the training samples, is given by:

$$\nabla_{\omega_R}\mathcal{L} = \frac{1}{S}\sum_{i=1}^{S} \nabla_{\omega_R} l(f_{\omega_R}(\mathbf{y}_k^{(i)}), m^{(i)}). \tag{6}$$

The transmitter gradient is computed as follows:

$$\begin{aligned} \nabla_{\omega_k}\mathcal{L} = &\mathbb{E}_m\Bigg[\int l(f_{\omega_R}^{(R)}(\mathbf{y}_k), m) \nabla_{\omega_k} f_{\omega_k}^{(T)}(m) \\ &\nabla_{\mathbf{x}_k} p(\mathbf{y}_k \mid \mathbf{x}_k)\Big|_{\mathbf{x}_k = f_{\omega_k}^{(T)}(m)} d\mathbf{y}_k\Bigg], \end{aligned} \tag{7}$$

where $\nabla_{\omega_k} f_{\omega_k}^{(T)}(m)$ is the Jacobian of the transmitter output, and $\nabla_{\mathbf{x}_k} p(\mathbf{y}_k \mid \mathbf{x}_k)$ is the channel gradient with respect to its input. As the actual channel model $p(\mathbf{y}_k \mid \mathbf{x}_k)$ may be unknown or non-differentiable, its gradient might be undefined [10].

B. Channel Data Set Generation

To simulate the switching between LoS and NLoS conditions, we utilize a Markov process to model a non-differentiable MIMO channel that reflects the likelihood of remaining in or switching between these states over time. The LoS condition is

characterized by Rician fading, which combines a direct LoS component with scattered multipath components influenced by factor K. In contrast, the NLoS condition is modeled using Rayleigh fading, accounting for scattered signals only. For the LoS state, the channel matrix is modeled as

$$\mathbf{H}_{\text{LoS}} = \sqrt{\frac{K}{K+1}}\mathbf{H}_{\text{LoS}}^{\text{dir}} + \sqrt{\frac{1}{K+1}}\mathbf{H}_{\text{LoS}}^{\text{scat}}, \tag{8}$$

where $\mathbf{H}_{\text{LoS}}^{\text{dir}}$ is the direct component, and $\mathbf{H}_{\text{LoS}}^{\text{scat}}$ is the scattered component. For the NLoS state, the channel gain follows a Rayleigh distribution, $\mathbf{H}_{\text{NLoS}} \sim \mathcal{CN}(0,1)$, representing scattered multipath components. Transitions between LoS and NLoS are governed by the Markov process with a transition probability matrix:

$$P_{tr} = \begin{bmatrix} p(\text{LoS} \to \text{LoS}) & p(\text{LoS} \to \text{NLoS}) \\ p(\text{NLoS} \to \text{LoS}) & p(\text{NLoS} \to \text{NLoS}) \end{bmatrix}. \tag{9}$$

The Markov process induces non-differentiability, as the channel remains constant within each state but switches abruptly between LoS and NLoS. At each time step t, a MIMO channel realization $\mathbf{H}(t) \in \mathbb{C}^{N_r \times N_t}$ is generated, capturing the sharp transitions and stability intervals characteristic of real-world wireless channels.

III. The Proposed DDPG-based FJ Framework

A. DDPG Preliminary

In this model, an agent engages with its environment by adjusting its actions based on accumulated experiences to maximize cumulative rewards over time. At each time step t, the agent perceives the current state s_k and selects an action a_k according to its policy π. Upon taking the action, the agent gains an immediate reward r_k and transitions to a new state S_{t+1}. The action-value function, often referred to as the Q-function, represents the expected return for specific state-action pairs under a given policy π and is defined as:

$$Q^{\pi}(s,a) = \mathbb{E}\left[G_t \mid s_k = s, a_k = a, \pi\right], \tag{10}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ denotes the total discounted reward, with discount factor $\gamma \in (0,1]$. The Q-learning algorithm iteratively updates the Q-value as follows:

$$\begin{aligned} Q(s,a) \leftarrow & Q(s,a) + \alpha\big[R(s,a) \\ & + \gamma \max Q(s',a') - Q(s,a)\big], \end{aligned} \tag{11}$$

where $R(s,a)$ is the immediate reward, s' is the next state, and a' is the next action. DNN-based Deep Q-Network (DQN) improves upon traditional Q-learning by approximating Q-values for complex problems. However, DQN struggles with continuous action spaces, prompting the development of DDPG. This actor-critic method handles continuous actions and trains both transmitter and receiver without requiring prior knowledge of the channel. In DDPG, the actor-network $\mu(s|\omega^{\mu})$ maps the states to actions, while the critic network $Q(s,a|\omega^{Q})$ assesses the quality of actions. Target networks μ' and Q' are periodically updated to follow the main networks, which helps stabilise the training process. An experience replay buffer stores transitions (s,a,r,s') to enable decorrelated sampling. The critic network seeks to reduce the loss L between the estimated Q-values and the target Q-values.

$$L = \frac{1}{N}\sum_{i}\left(y_i - Q(s_i,a_i|\omega^{Q})\right)^2, \tag{12}$$

where $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\omega^{\mu'})|\omega^{Q'})$, and N is the mini-batch size. The actor-network updates through the policy gradient:

$$\nabla_{\omega^{\mu}} J \approx \frac{1}{N}\sum_{i} \nabla_a Q(s,a|\omega^{Q})|_{s=s_i,a=\mu(s_i)} \nabla_{\omega^{\mu}} \mu(s|\omega^{\mu})|_{s_i}. \tag{13}$$

DDPG refines the policy and action-value functions by iteratively updating the networks until convergence or termination.

B. DDPG-Based MIMO Communication

Regarding DDPG-based MIMO configuration, the state space can incorporate additional parameters such as antenna configurations and spatial beamforming vectors, while the action space can represent multi-dimensional signals corresponding to multiple antennas. The antenna selection and power allocation is conducted automatically via the training process. The reward function would also need to account for optimising multiple spatial streams, balancing the secrecy rate for all streams, and maintaining reliable communication for legitimate users. Particularly, the transmitted message $m_k \in \mathcal{M}$ is embedded in the symbol $\mathbf{s}$ and transmitted through a noisy channel. The transmitter encodes the symbol $\mathbf{s}$ into the signal $\mathbf{x}_k \in \mathbb{C}^n$, representing n discrete channel uses. At the same time, the receiver aims to decode the received signal $\mathbf{y}_k \in \mathbb{C}^n$ back into the estimated symbol $\hat{\mathbf{s}}$ and then maps it to the decoded message $\hat{m}_k \in \mathcal{M}$. The E2E system is described as a concatenation of functions:

$$\hat{m}_k = f_D(f_h(f_E(m_k;\omega_E));\omega_D), \tag{14}$$

where f_E is the encoder function that maps the message to the encoded signal, i.e. $\mathbf{x}_k = f_E(m_k;\omega_E)$, with ω_E being the trainable weights of the transmitter. The channel effect is denoted as f_h, and the decoder f_D maps the received signal $\mathbf{y}_k$ to the estimated symbol $\hat{m} = f_D(\mathbf{y}_k;\omega_D)$, where ω_D are the receiver's trainable weights. The system is trained using supervised learning to minimize $\mathcal{L}(\mathbf{s},\hat{\mathbf{s}})$, which evaluates the difference between the original and estimated symbols.

Figure 1 illustrates the DDPG-based end-to-end (E2E) communication system. In this framework, the message m_k serves as the observation state $s_k = m_k$, the encoded signal $\mathbf{x}_k$ is the action $a_k = \mathbf{x}_k$, the received signal is $\mathbf{y}_k$, and the decoded message is denoted by $\hat{m}$. The DDPG framework comprises actor and critic networks: the actor (transmitter) encodes m into $\mathbf{x}_k$, while the critic (receiver) decodes $\mathbf{y}_k$ into $\hat{m}$. At each time slot k, the communication loss r_k^{com} is defined as the negative log-likelihood of the categorical cross-entropy loss between the original and decoded messages, expressed as:

$$r_k^{\text{com}} = -\frac{1}{N}\sum_{i=1}^{N}(s_k)_i \log\left[f_{\omega_R}(\mu((s_k)_i|\omega_k))\right], \tag{15}$$

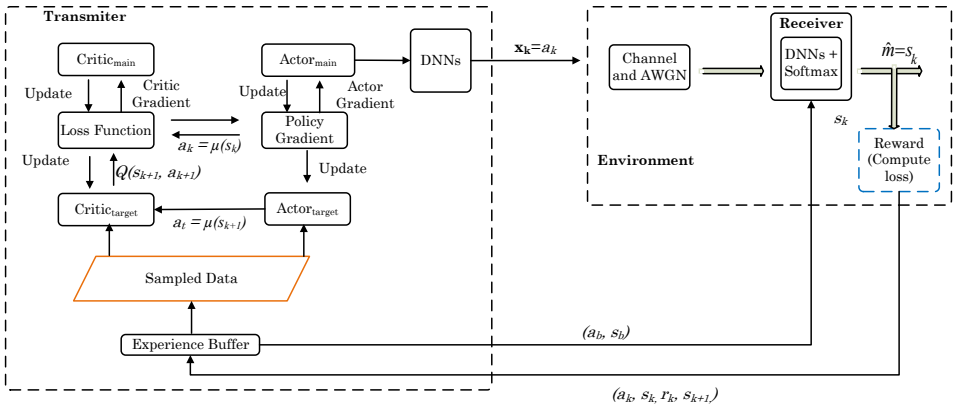

Fig. 1: The proposed DDPG-based communication framework.

where N is the number of samples, ω_R and ω_k are the parameters of the receiver and transmitter, respectively. $\mu(s_k|\omega_k)$ represents the action generated by the transmitter for the given state s_k, and f_{ω_R} denotes the output of the receiver model. In each episode, the actor-network generates an encoded signal $\mathbf{x}_k$, which is transmitted over the channel, resulting in the altered received signal $\mathbf{y}_k$. The receiver network then decodes $\mathbf{y}_k$ to produce $\hat{m}_k$. The state, action, communication loss, and next state are stored in an experience replay buffer, allowing mini-batch sampling for network updates. During training, batches of state-action pairs, denoted s_b and a_b, are randomly sampled from this experience buffer to update the receiver model. Here, s_b and a_b refer to the batches of past states and actions used for training. The target Q-value is given by [11]:

$$y_k = r_k^{\text{com}} + \gamma Q(s_{t+1}, a_{t+1}), \tag{16}$$

where γ is the discount factor. The actor and critic networks are optimized to maximize the expected Q-value:

$$J(\omega) = \mathbb{E}[Q(s,a)|s=s_k, a=\mu(s_k)], \tag{17}$$

with the gradient of the Q-value with respect to the actor's parameters ω_μ as:

$$\nabla_{\omega_\mu} J \approx \nabla_a Q(s,a) \nabla_{\omega_\mu} \mu(s|\omega_\mu). \tag{18}$$

The target network parameters ω' are updated using a soft update mechanism:

$$\omega' = \tau\omega + (1-\tau)\omega', \tag{19}$$

where ω represents the main network parameters, and τ (with $\tau \ll 1$) controls the update rate. This process enables the DDPG framework to optimize transmitter and receiver performance without prior knowledge of the channel model.

C. DDPG-Based FJ System Model

The primary objective of the DDPG-based FJ system is to maximize the secrecy rate by generating jamming signals that selectively interfere with the eavesdropper while preserving the signal integrity for the legitimate receiver. We leverage the DDPG-based communication model to design a DDPG-based FJ system. The goal of the training is twofold: maximizing

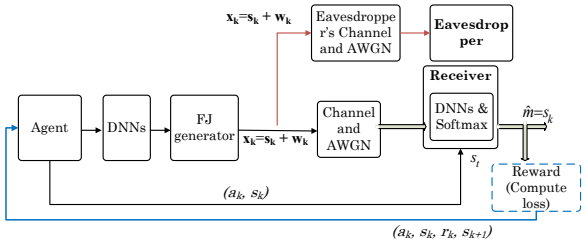

Fig. 2: DDPG-based Friendly Jamming (FJ) framework.

the mutual information (MI) difference between the transmitter and the receiver and designing the FJ signal to degrade Eve's channel quality, see (4). The system model is illustrated in Fig. 2, where the FJ signal $\mathbf{w}_k$ is generated via the FJ generator and injected into the transmitted signals $\mathbf{x}_k$.

1) Handling NDC for FJ: NDC, with its abrupt state changes caused by factors like multipath fading and LoS/NLoS transitions, presents major challenges for conventional gradient-based FJ optimization, which relies on continuous gradients for parameter tuning. In contrast, DDPG, a model-free reinforcement learning approach, overcomes this limitation by learning from state-action-reward sequences without requiring gradients. Utilizing an experienced replay buffer and continuous action space, DDPG effectively adapts to sudden channel variations, dynamically generating FJ signals to optimize secrecy rates while maintaining legitimate communication quality. This adaptability ensures robust security performance in real-world wireless environments with unpredictable channel dynamics.

2) Friendly Jamming (FJ) Generation: Our approach employs an FJ-based beamforming strategy to optimize the secrecy rate reward r_k^{sec}. In the FJ generator block, the beamforming vector $\mathbf{t}_k$ is derived through a process involving the channel matrix $\mathbf{H}_k$ and the signal-to-noise ratio (SNR), see Fig. 3. Specifically, $\mathbf{H}_k$ and SNR are input into dense embedding layers. Next, the phase function and normalization produce a temporary variable θ_k representing the beamforming vector's phase. This phase variable θ_k is then used to construct $\mathbf{t}_k$ as $\mathbf{t}_k = \cos(\theta_k) + j\sin(\theta_k)$, where θ_k is the output of the dense layer processing, specifically designed to ensure that $\mathbf{t}_k$

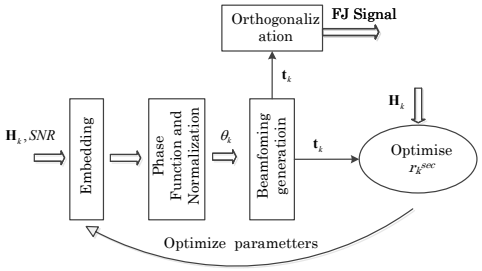

Fig. 3: Friendly generator structure.

maximizes r_k^{sec}.

The reward secrecy function in the DDPG-based FJ framework, represented by the reward r_k^{sec}, is defined as:

$$r_k^{\text{sec}} = \mathbb{E}\Big[r_k^{\text{com}} - \log \frac{\det(\mathbf{K}_k + \mathbf{F}_k)}{\det(\mathbf{K}_k)}\Big]. \tag{20}$$

The FJ signals are designed to be orthogonal to the main beamforming vector $\mathbf{t}_k$. This ensures the jamming signal does not interfere with the legitimate receiver's signal, as proposed in [7]. By degrading the eavesdropper's channel quality while preserving the legitimate receiver's, $\mathbf{t}_k$ and $\mathbf{w}_k$ adapt to current channel conditions, maximizing the loss function and enhancing the secrecy rate by focusing interference on the eavesdropper.

3) Implementation: MIMO channels are modeled with block-fading behavior, alternating between LoS and NLoS states. The system state s_k captures real and imaginary components of signals, enabling real-valued actor and critic processing. The actor-network generates the action a_k, including information-bearing and FJ signals, to boost main channel capacity while disrupting Eve's. Using the DDPG algorithm, transmit signals are optimized to maximize secure communication, with the expected secrecy rate evaluated in the non-differentiable environment.

Both networks process real-valued inputs, separating real and imaginary components, with the actor's output recombined for complex-valued transmission. The reward function, based on the secrecy rate, guides the actor to improve security despite channel non-differentiability. Training starts with network initialization, experience replay setup, action generation, and applying actions to obtain the next state and reward. The critic updates the value function, while the actor optimizes the secrecy rate. Target networks are updated gradually to ensure stable learning in the challenging NDC environment.

IV. Simulation Results and Discussion

This section compares our method with NDC to the baseline approaches in [8] and our previous works, in [7], [12], with differentiable channels. The reliable transmission and security are evaluated by BLER and secrecy rate, respectively. The channels are influenced by several key parameters, such as LoS/Rician fading and NLoS/Rayleigh fading. The number of transmit and receive antennas are set to $N_t = 10$ and $N_r = 4$, respectively. The Rician K-factor is $K = 10$. The Markov process governs the dynamic switching between LoS and NLoS states through the matrix $P = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$. This implies an 80% chance of staying in LoS and a 30% chance of transitioning from NLoS to LoS.

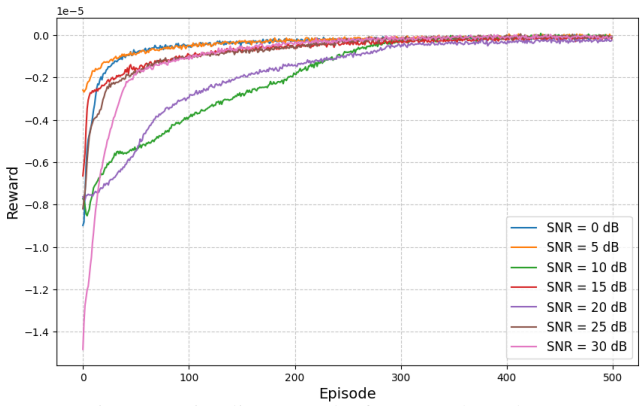

Fig. 4: Episodic reward of DDPG-based FJ.

Regarding the DDPG model, the soft update rate τ is set to 0.007, allowing slow updates of the target networks, while the actor and critic learning rates are 0.001 and 0.0005, respectively. The input message size M is 64. As shown in Fig. 4, the reward convergence over 500 episodes varies with SNR levels (0-30 dB). Lower SNRs (0-10 dB) converge faster, stabilizing after 100 episodes with higher rewards, while higher SNRs (15-30 dB) take longer, stabilizing after 200 episodes. Despite slower learning at higher SNRs, all levels eventually converge, with lower SNRs yielding slightly better performance.

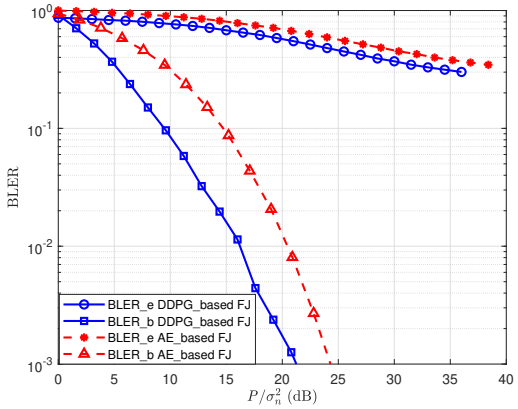

Fig. 5: BLERs at Bob and Eve in AE and DDPG-based FJ.

Fig. 5 compares BLER between DDPG-based FJ and AE-based FJ for Bob and Eve. For Bob, DDPG-based FJ (blue squares) achieves near-zero BLER at 25 dB, outperforming AE-based FJ (red triangles). For Eve, DDPG-based FJ (blue circles) maintains high BLER, effectively limiting decoding, while AE-based FJ (red stars) allows rapid BLER reduction at higher SNRs. Overall, DDPG-based FJ offers better reliability for Bob and stronger security for Eve across the SNR range.

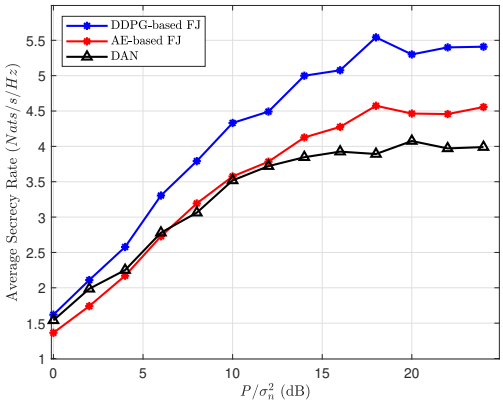

Fig. 6: The average secrecy rate versus SNR.

Fig. 6 illustrates the average secrecy rate as a function of the normalized transmit power P/σ_n^2 for three different methods: the proposed DDPG-based FJ, the AE-based FJ, and Deep Artificial Noise (DAN) in [8]. The DDPG-based FJ, blue circles, consistently achieve a higher secrecy rate across all SNR values than the AE-based FJ, red stars and DAN, black triangles. At low SNR, all methods show similar performance. However, as the SNR increases, the DDPG-based FJ rapidly improves and peaks at around 15 dB with a secrecy rate exceeding 5.5 nats/s/Hz. In contrast, the AE-based FJ gradually increases but reaches a lower peak of around 4.5 nats/s/Hz, while DAN exhibits the slowest improvement, peaking below 4 nats/s/Hz. This demonstrates that the DDPG-based approach is more effective in optimizing secrecy under various channel conditions, outperforming the baseline methods, especially at higher SNR values.

The DDPG-based FJ offers promising improvements in secure communication but faces challenges in computational complexity, energy efficiency, and hardware implementation. Computational complexity arises from the actor-critic framework and Markov process modeling abrupt LoS/NLoS transitions. As shown in Fig. 4, training requires significant resources, with convergence taking longer at higher SNRs (e.g., 200 episodes for 30 dB), reflecting increased optimization efforts in complex channel conditions. The system ensures the FJ signal is orthogonal to the beamforming vector, minimizing interference with legitimate receivers. Fig. 5 shows Bob achieves near-zero BLER, while Eve maintains high BLER, demonstrating effective energy use for jamming. Additionally, secrecy rate results in Fig. 6 show the DDPG-based FJ consistently outperforms AE-based FJ and DAN, achieving over 5.5 nats/s/Hz at 15 dB, though real-time adaptation remains energy-intensive.

Advanced GPUs or TPUs are required for high-dimensional MIMO setups and real-time optimization. Scaling to larger configurations introduces challenges, but pre-training on powerful hardware with lightweight edge deployment could address these. Performance also depends on antenna configuration and hyperparameters. Larger antenna setups ($N_t = 10, N_r = 4$) improve secrecy rates and reduce BLER but increase hardware demands. Hyperparameters like learning rates and the soft update rate ($\tau = 0.007$) significantly affect convergence stability.

V. Conclusion

This paper introduced a DDPG-based Friendly Jamming (FJ) approach designed for non-differentiable channels (NDC), which significantly improves both communication reliability and security compared to AE-based FJ methods. Our approach achieves lower BLER for the legitimate receiver (Bob) and consistently maintains a high BLER for the eavesdropper (Eve) across various SNR conditions, demonstrating enhanced protection against eavesdropping. Our results highlight the strength of DDPG in managing NDC environments, addressing a critical gap in secure wireless communication by offering a robust solution for mid-to-high SNR regimes. In future work, we aim to extend this approach to more realistic and complex channel models, addressing challenges like missing data, quantization, and multi-user scenarios, as well as exploring its scalability in massive MIMO settings.

References

[1] A. Mukherjee, "Physical-layer security in the internet of things: Sensing and communication confidentiality under resource constraints," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015.

[2] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey," *IEEE Commun. Surv. Tutor*, vol. 21, no. 2, pp. 1773–1828, 2018.

[3] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang, B. Ai, and Z. Zhong, "Machine learning-enabled los/nlos identification for mimo systems in dynamic environments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3643–3657, 2020.

[4] O. Jovanovic, M. P. Yankov, F. Da Ros, and D. Zibar, "Gradient-free training of autoencoders for non-differentiable communication channels," *Journal of Lightwave Technology*, vol. 39, no. 20, pp. 6381–6391, 2021.

[5] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE Trans. Wireless Commun*, vol. 7, no. 6, pp. 2180–2189, 2008.

[6] J. Choi, "A robust beamforming approach to guarantee instantaneous secrecy rate," *IEEE Trans. Wireless Commun*, vol. 15, no. 2, pp. 1076–1085, 2015.

[7] B. M. Tuan, D. N. Nguyen, N. L. Trung, V.-D. Nguyen, N. Van Huynh, D. T. Hoang, M. Krunz, and E. Dutkiewicz, "Securing mimo wiretap channel with learning-based friendly jamming under imperfect csi," *arXiv preprint arXiv:2312.07011*, 2023.

[8] S. Yun, J.-M. Kang, I.-M. Kim, and J. Ha, "Deep artificial noise: Deep learning-based precoding optimization for artificial noise scheme," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3465–3469, 2020.

[9] A. Mukherjee, S. A. A. Fakoorian, J. Huang, and A. L. Swindlehurst, "Principles of physical layer security in multiuser wireless networks: A survey," *IEEE Commun. Surv. Tutor*, vol. 16, no. 3, pp. 1550–1573, 2014.

[10] R. Ye, Y. Peng, F. Al-Hazemi, and R. Boutaba, "A robust cooperative jamming scheme for secure uav communication via intelligent reflecting surface," *IEEE Transactions on Communications*, vol. 72, no. 2, pp. 1005–1019, 2024.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[12] B. M. Tuan, T. D. Tuyen, N. L. Trung, and N. V. Ha, "Autoencoder based friendly jamming," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.