Deep Reinforcement Learning-Based Secure Standalone Intelligent Reflecting Surface Operation

Yasaman Omid[†], Yansha Deng[‡], Arumugam Nallanathan[†].
[†]School of Electronic Engineering and Computer Science, Queen Mary University of London, U.K.
[‡] Department of Engineering, Kings College London, U.K.
Emails: y.omid@qmul.ac.uk, yansha.deng@kcl.ac.uk, a.nallanathan@qmul.ac.uk

***Abstract*—In this paper, we investigate secure wireless communication in an intelligent reflecting surface (IRS)-assisted system where the IRS is used to secure the communication of one legitimate receiver in presence of an eavesdropper. We assume that the IRS is standalone, i.e. the passive beamforming of the IRS is carried out completely on its own. Thus, we design an IRS with several passive elements and only two RF chains that can obtain a partial channel state information (CSI) among each node and the IRS. The partial CSI is then mapped into full CSI by using the correlation information between the channels of different IRS elements. We develop a deep reinforcement learning (DRL)-based framework using the deep deterministic policy gradient (DDPG) algorithm to obtain the IRS beamforming vector resulting in maximizing the secrecy rate. Numerical results demonstrate the ability of this technique to secure the wireless communication system.**

***Index Terms*—Intelligent Reflecting Surface, Reconfigurable Intelligent Surface, Deep Reinforcement Learning, DDPG, Physical layer security.**

I. Introduction

The intelligent reflecting surface (IRS) is a novel promising technology to improve the spectral efficiency, energy efficiency and security in 5G and beyond communications [1], [2]. An IRS is a meta-surface that employs a large number of passive elements to reconfigure the propagation environment and bring full coverage for the blind spots. IRS is specially used in physical layer security due to its ability to strengthen or weaken the reflected signals. As a result, it can easily suppress the signals at the eavesdroppers or strengthen the signals at the legitimate receivers, thus increasing the secrecy rate [3]. However, the benefits of the IRS heavily rely upon acquisition of accurate channel state information (CSI), which still remains a challenge. Many beamforming methods in the literature have assumed availability of the perfect or imperfect cascaded channels between the transmitter and the receivers through the IRS [4]–[6], which motivated several researchers to focus on estimation of the cascaded channels [7], [8]. However, due to the large dimension of the cascaded channels, the channel estimation would impose a large overhead on the system [9]. Against this background, in the recent literature, the authors have focused on estimation of the IRS-related channels, by assuming that the IRS is equipped with a few active elements (RF chains) and it can partially estimate the IRS-receiver and IRS-transmitter channels [9]–[13]. By doing so, the IRS can calculate its passive reflection matrix since it has access to partial IRS-related channels. This initially reduces the overhead since there is no need for the transmitter to adjust the IRS's phases. This design which we refer to as an standalone IRS, was first introduced in [12].

In [2], the physical layer security was investigated in an IRS-aided system. The authors designed a DRL-based secure beamforming approach where the active beamforming at the base station (BS) and the passive beamforming the IRS were jointly optimized at the BS to maximize the secrecy rate. In [9], an IRS-aided mm-wave massive multiple input multiple output (MIMO) system was considered, where the IRS was employed with a few active elements to estimate IRS-user channels. To address this issue, the authors proposed a complex valued denoising convolution neural network (CNN) assisted compressive sensing channel estimation technique. The authors in [10] presented two approaches for the beamforming at the IRS. In the first approach, they attempted to leverage the compressive sensing technique to construct all the IRS-user channels from the partial channels gathered at the active elements. In the second approach, they used deep learning tools so that the IRS could learn how to interact with the incident signal, given the partial channels at the active elements. In the second approach, the channels between the passive elements of the IRS and the users remained unknown. The authors also presented their findings on the second approach [11], [12], where supervised learning and deep reinforcement learning (DRL) were used to map the partial channels to the optimal IRS beamforming. In [13], an IRS-aided transmission of a single-antenna transmitter to a single-antenna receiver was studied. The IRS was employed with a few active elements which were chosen by an offline deep learning (DL)-based antenna selection network. The full channels where then extrapolated from the partial CSI obtained by the active elements using a CNN. Finally, using a fully-connected neural network (NN), the partial channels were mapped to the optimal set of IRS phase shifts.

While the role of IRS in securing the wireless communication systems has been studied in the literature, to the best of our knowledge, the physical layer security in communication systems aided by an standalone IRS has never been addressed. Thus, in this paper, we first design an standalone IRS that only contains two active elements. Hence, the IRS can acquire

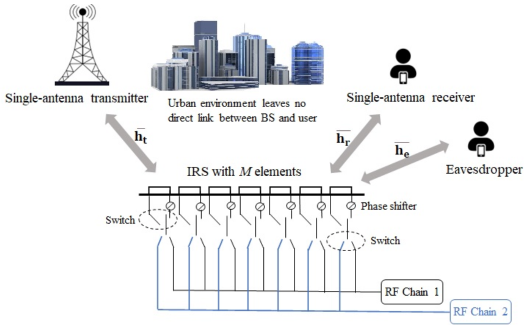

Single-antenna transmitter
Urban environment leaves no direct link between BS and user
Single-antenna receiver
$\bar{\mathbf{h}}_\mathbf{t}$
$\bar{\mathbf{h}}_\mathbf{r}$
$\bar{\mathbf{h}}_\mathbf{e}$
Eavesdropper
IRS with M elements
Phase shifter
Switch
Switch
RF Chain 1
RF Chain 2

Figure 1: System model.

partial channels and use an online mapping technique to map them to full channels by leveraging the correlation property of channels in the IRS. Then, we use this information to design a DRL-based approach determining the IRS reflection matrix that maximizes the secrecy rate. We use the deep deterministic policy gradient (DDPG) technique to select the phase shifts from a continuous bound. DDPG works well in continuous and high-dimensional action-spaces. Our numerical results show the accuracy of mapping the partial channels to estimate the full channel gains. Also, it demonstrates the efficiency of the algorithm to achieve physical layer security.

II. System Model

We consider the IRS-aided transmission of a single-antenna transmitter to a single-antenna receiver in presence of a single-antenna eavesdropper. The locations of the IRS and the transmitter are fixed while the receiver and the eavesdropper are mobile. The IRS is equipped with $M = M_y \times M_z$ reflecting elements, two RF chains and $2M$ switches that can connect any two elements to the two RF chains. The switches are programmed such that only two elements would be connected to the RF chain and the rest would be passive. The first M switches decide whether an element is active or passive, while the second M switches determine which active element is connected to the RF chains. The design of the switching structure is motivated by [9]. It is assumed that there is no direct link between the transmitter and the receiver and it is blocked by the elements of an urban environment. Thus, the focus of this study is on the IRS-related channels. Fig. 1 depicts the system model.

A. Channel Acquisition

We adopt the orthogonal frequency division multiplexing (OFDM) scheme with K sub-carriers in order to combat the frequency-selective fading. For the sake of simplicity, we investigate the transmission over a single sub-carrier resulting a flat fading scenario, thus the size of the channels are $M \times 1$. Due to path loss, we only consider the first pilot signal reflected by the IRS and ignore the signals that are reflected two or more times. Assuming that the set of active elements is denoted by $\mathbf{a} = [m, m^{'}]$ with m and $m^{'}$ being the index of the two active elements, the received signal at the active elements of the IRS is

$$\mathbf{Y}^{[\mathbf{a}]} = \bar{\mathbf{h}}_t^{[\mathbf{a}]}\mathbf{x}_t + \bar{\mathbf{h}}_r^{[\mathbf{a}]}\mathbf{x}_r + \bar{\mathbf{h}}_e^{[\mathbf{a}]}\mathbf{x}_e + \mathbf{W}^{[\mathbf{a}]}, \tag{1}$$

where $\mathbf{x}_t$ is the $1 \times \gamma_t$ pilot signal transmitted by the transmitter, $\mathbf{x}_r$ is the $1 \times \gamma_r$ pilot signal transmitted by the receiver and $\mathbf{x}_e$ is the $1 \times \gamma_e$ pilot signal transmitted by the eavesdropper. Also, $\bar{\mathbf{h}}_t^{[\mathbf{a}]}$, $\bar{\mathbf{h}}_r^{[\mathbf{a}]}$ and $\bar{\mathbf{h}}_e^{[\mathbf{a}]}$ are the IRS-transmitter, the IRS-receiver and the IRS-eavesdropper partial channels, respectively, i.e. $\bar{\mathbf{h}}_t \in \mathbb{C}^{M\times 1}$, $\bar{\mathbf{h}}_r \in \mathbb{C}^{M\times 1}$ and $\bar{\mathbf{h}}_e \in \mathbb{C}^{M\times 1}$ while $\bar{\mathbf{h}}_t^{[\mathbf{a}]} \in \mathbb{C}^{2\times 1}$, $\bar{\mathbf{h}}_r^{[\mathbf{a}]} \in \mathbb{C}^{2\times 1}$ and $\bar{\mathbf{h}}_e^{[\mathbf{a}]} \in \mathbb{C}^{2\times 1}$. We assume $\mathbf{W}^{[\mathbf{a}]} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ is the additive white Gaussian noise at the active elements of IRS. The partial IRS-related channels are estimated via the least square (LS) technique as [14]

$$\mathbf{h}_t^{[\mathbf{a}]} = \frac{1}{\gamma_t\sqrt{p_t}}\mathbf{Y}^{[\mathbf{a}]}\mathbf{x}_t^H, \tag{2}$$

$$\mathbf{h}_r^{[\mathbf{a}]} = \frac{1}{\gamma_r\sqrt{p_r}}\mathbf{Y}^{[\mathbf{a}]}\mathbf{x}_r^H, \tag{3}$$

$$\mathbf{h}_e^{[\mathbf{a}]} = \frac{1}{\gamma_e\sqrt{p_e}}\mathbf{Y}^{[\mathbf{a}]}\mathbf{x}_e^H, \tag{4}$$

where p_t, p_r and p_e are the pilot power of the respective nodes. Since the IRS elements are located very close to one another, there is high correlation among the channels between the neighboring IRS elements and each of the nodes. We intend to find the correlation between each couple of IRS elements. By leveraging this information, we can enhance the performance of our design. To do so, we define $\mathbf{C}_t \in \mathbb{C}^{M\times M}$, $\mathbf{C}_r \in \mathbb{C}^{M\times M}$ and $\mathbf{C}_e \in \mathbb{C}^{M\times M}$ as the correlation matrices, with the entries $\mathbf{C}_t^{[m,m^{'}]} = \frac{\mathbf{h}_t^{[m]}}{\mathbf{h}_t^{[m^{'}]}}$, $\mathbf{C}_r^{[m,m^{'}]} = \frac{\mathbf{h}_r^{[m]}}{\mathbf{h}_r^{[m^{'}]}}$ and $\mathbf{C}_e^{[m,m^{'}]} = \frac{\mathbf{h}_e^{[m]}}{\mathbf{h}_e^{[m^{'}]}}$. We design the switching elements such that the two active elements ($\mathbf{a} = [m, m^{'}]$) are selected based on **Algorithm 1**. In this algorithm, a matrix named $\mathbf{A} \in \mathbb{C}^{\frac{M^2-M}{2}\times 2}$ is generated whose rows contain the index of the two activated IRS elements. In each time step of the DRL algorithm (which will be discussed later on), one row of matrix $\mathbf{A}$ is selected sequentially to determine the active elements. By using this algorithm, in each time step, the correlation of two IRS elements to one another can be determined, and the correlation matrix is fully updated every $\frac{M^2-M}{2}$ steps. After

Algorithm 1 The algorithm for choosing the index of the active elements

```
Input : $M$, $counter = 1$
for $m = 1, \ldots, M-1$ do
    for $m^{'} = m+1, \ldots, M$ do
        $\mathbf{a} = [m, m^{'}]$
        $\mathbf{A}^{[counter]} = \mathbf{a}$
        $counter += 1$
    end for
end for
Output : $\mathbf{A} \in \mathbb{C}^{\frac{M^2-M}{2}\times 2}$, which includes the indexes
    of the two active elements in every $\frac{M^2-M}{2}$ time steps.
```

the correlation matrices have been updated once, i.e. in the time steps $n > \frac{M^2-M}{2}$, the correlation matrices would be updated by using an expectation over all past time steps. When the IRS partially collects the channels in each time step, it can estimate the full channels using the correlations matrices, i.e. the channels at the ith element of the IRS, $i \neq m, m^{'}$, are

$$\mathbf{h}_t^{[i]} = \frac{1}{2}(\mathbf{h}_t^{[m^{'}]}\mathbf{C}_t^{[i,m^{'}]} + \mathbf{h}_t^{[m]}\mathbf{C}_t^{[i,m]}), \tag{5}$$

$$\mathbf{h}_r^{[i]} = \frac{1}{2}(\mathbf{h}_r^{[m^{'}]}\mathbf{C}_r^{[i,m^{'}]} + \mathbf{h}_r^{[m]}\mathbf{C}_r^{[i,m]}), \tag{6}$$

$$\mathbf{h}_e^{[i]} = \frac{1}{2}(\mathbf{h}_e^{[m^{'}]}\mathbf{C}_e^{[i,m^{'}]} + \mathbf{h}_e^{[m]}\mathbf{C}_e^{[i,m]}). \tag{7}$$

B. Problem Formulation

After the channel acquisition process, in the downlink the transmitter sends its data to the receiver. Since the transmission is aided by an IRS, the received signal at the receiver is

$$y_r = \mathbf{h}_r^T \mathbf{\Psi} \mathbf{h}_t x + n_r, \tag{8}$$

where $\mathbf{h}_r \in \mathbb{C}^{M\times 1}$ and $\mathbf{h}_t \in \mathbb{C}^{M\times 1}$ are the IRS-receiver and the IRS-transmitter estimated channels, $\Psi = \mathrm{diag}[e^{j\psi^{[1]}}, ..., e^{j\psi^{[M]}}]$ is the diagonal reflection matrix of the IRS, x is the precoded signal where $\mathbb{E}[|x|^2] = P$ and $n_r \sim \mathcal{CN}(0, \sigma_r^2)$ is the additive Gaussian noise at the receiver. Additionally, the received signal at the eavesdropper is

$$y_e = \mathbf{h}_e^T \mathbf{\Psi} \mathbf{h}_t x + n_e, \tag{9}$$

where $\mathbf{h}_e \in \mathbb{C}^{M\times 1}$ is the estimated IRS-eavesdropper channel and $n_e \sim \mathcal{CN}(0, \sigma_e^2)$ is the additive Gaussian noise at the eavesdropper. Based on (8), the data rate of the receiver is

$$R^r = \log_2\left(1 + \frac{P}{\sigma_r^2}|\mathbf{h}_r^T \mathbf{\Psi} \mathbf{h}_t|^2\right), \tag{10}$$

while the wiretapped data rate of the eavesdropper is given by

$$R^e = \log_2\left(1 + \frac{P}{\sigma_e^2}|\mathbf{h}_e^T \mathbf{\Psi} \mathbf{h}_t|^2\right). \tag{11}$$

The achievable secrecy rate from the transmitter to the receiver can now be expressed by [2], [15]

$$R^{sec} = \max\left(0, R^r - R^e\right). \tag{12}$$

The goal is to maximize the secrecy rate by optimizing the IRS reflection matrix. Thus, the optimization problem is written by

$$\begin{aligned} \max_{\mathbf{\Psi}} \quad & \max\left(0, R^r - R^e\right) \\ \mathrm{s.t.} \quad & |\psi^{[i]}| \leq \pi, i = 1, ..., M. \end{aligned} \tag{13}$$

This optimization problem is non-convex and very hard to tackle. Additionally, the system parameters including the channel gains are changing dynamically. The model-free RL is a dynamic tool which can solve decision-making problems in dynamic environments. Thus, we employ this technique, by modeling our problem as a Markov decision process (MDP).

III. The DRL-Based Solution

In this section, a DRL-based method is introduced to solve (13). To do so,we first present the basics of DRL using the DDPG technique and then we formulate our problem.

A. Basics of DRL

In a reinforcement learning (RL) system, an agent interacts with the environment through a series of discrete time steps, a set of states and a set of actions. This interaction can be described as a MDP. In the nth time step, the agent takes an action a_n based on the observed state s_n and a policy π, and receives a reward r_n. Then, by taking this action, the agent transitions from the state s_n to the next state s_{n+1}. RL comes with two algorithms to find the optimal policy: the value-based and the policy based algorithms. Deep Q network (DQN) [16] is a value-based algorithm that uses a NN to estimate the state-action Q-function. To do so, for a policy π, an action a and a state s the Q-function is presented by $Q^{\pi}(s, a, \boldsymbol{\theta}) = \mathbb{E}_{\pi}\{R_n | s_n = s, a_n = a, \pi\}$, where $\boldsymbol{\theta}$ defines the parameters of the deep NN (DNN), $\mathbb{E}_{\pi}\{.\}$ is expectation over π, $R_n = \sum_{n=0}^{\infty} \lambda^n r_n$ represents the expected cumulative reward and $\lambda \in (0, 1]$ stands for the discount factor. Through training the DNN, the DQN algorithm attempts to maximize the Q-value for a set of state-action pair. For this, it randomly samples the training batch from a replay buffer. Although DQN can solve problems with high-dimensional observation spaces, it can only work in low-dimensional and discrete action spaces. On the other hand, policy gradient (PG) [17] is a policy-based algorithm which can handle continuous action spaces and seeks to maximize the discounted cumulative episodic reward. In the time step n in the episode, the agent selects an action based on the policy $\pi_{\boldsymbol{\theta}}$; hence, training the policy can be done through gradient descend as $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \beta \mathbb{E}_{\pi_{\boldsymbol{\theta}_n}}\left\{\nabla_{\boldsymbol{\theta}_n} \log \pi_{\boldsymbol{\theta}}(s, a) Q_{\pi_{\boldsymbol{\theta}_n}}(s, a)\right\}$, in which β represents the learning rate.

B. Basics of the DDPG algorithm

To apply DRL for continuous IRS beamforming, we need an off-policy technique that can work in a continuous high-dimensional action space. DDPG, which is a model-free off-policy actor-critic algorithm, combines the advantages of both DQN and PG, thus it can handle continuous and high-dimensional action spaces [18].

In DDPG, the actor network is a deterministic policy network (DPN) which chooses actions from a continuous action space, i.e. $a = \mu(s; \boldsymbol{\theta}_{\mu})$, where $\boldsymbol{\theta}_{\mu}$ represents the parameters of the actor network. The critic network, on the other hand, is a Q network $Q(s, a; \boldsymbol{\theta}_q)$ in which $\boldsymbol{\theta}_q$ stands for the parameters of the critic network. The critic evaluates the action taken by the actor network by assigning it a Q-value, and the main goal of DDPG is to maximize this Q-value. Similar to DQN, here we use an experience replay to reduce the correlation of different training samples. Furthermore, in order to calculate the corresponding target values, a copy of the actor and the critic network is created which are called the target networks and are denoted by $\mu^*(s; \boldsymbol{\theta}_{\mu^*})$ and $Q^*(s, a; \boldsymbol{\theta}_{q^*})$, respectively. We treat $\mu(s; \boldsymbol{\theta}_{\mu})$ and $Q(s, a; \boldsymbol{\theta}_q)$ as the evaluation networks.

The structure of each target network and its corresponding evaluation network is similar, but their parameters are different, i.e. $\boldsymbol{\theta}_{\mu^*} \neq \boldsymbol{\theta}_\mu$ and $\boldsymbol{\theta}_{q^*} \neq \boldsymbol{\theta}_q$. In order to update the target networks, the soft update is written as

$$\boldsymbol{\theta}_{j^*} = \tau\boldsymbol{\theta}_j + (1-\tau)\boldsymbol{\theta}_{j^*},\ j \in \{\mu, q\},\ \tau << 1. \tag{14}$$

One of the advantages of off-policy algorithms such as DDPG is that the problem of exploration can be treated independently from the learning process [18]. Here, an exploration policy $\mu^{'}$ is constructed by adding noise sampled from a noise process $\mathcal{N}$ to the actor network, i.e. $\mu^{'}(s_n) = \mu(s_n; \boldsymbol{\theta}_\mu) + \mathcal{N}$, where $\mathcal{N}$ is properly selected to match the environment.

C. The DRL formulation

The IRS, which is the DRL agent, interacts with its environment and defines the reflection matrix. In order to formulate the problem in the DRL format, the set of observations, actions and reward is presented in the following sub-sections.

1) Action

The action at the nth time step is denoted by $a_n = [\psi_n^{[1]}, ..., \psi_n^{[M]}]$ which is a $1 \times M$ vector that determines the IRS phase shifts. Thus, in each step, the agent needs to choose M values of $\psi^{[i]} \in [-\pi, \pi]$, $i = 1, ..., M$.

2) Observation

The goal of the DRL agent is to learn the IRS reflection matrix $\boldsymbol{\Psi}$ that maximizes the secrecy rate in (12). In each time step of DRL, the IRS-receiver and the IRS-eavesdropper channels change due to the mobility of the receiver and the eavesdropper. Thus, the transition from time step n into the next time step changes the observation space which contains: i) the estimated IRS-receiver and the IRS-eavesdropper channels, ii) the M phase shifts of the IRS $a_n = [\psi_n^{[1]}, ..., \psi_n^{[M]}]$, iii) the estimated cascaded transmitter-IRS-receiver and the transmitter-IRS-eavesdropper channels. Note that the observation space includes $(2M+2)$ complex values for the channels which are split into real and imaginary values, and M real values for the reflection phases. This makes the total size of the observation space as $(5M+4)$.

3) Reward

In the nth time step, the agent preforms an action, determining the reflection matrix of the IRS, $\boldsymbol{\Psi}$. Thus, the value of the data rate at the receiver and at the eavesdropper are changed. The reward at the time step n, where action a_n takes place, is determined by the secrecy rate in (12). To be more specific, we first define the continuous value of reward at the nth time step as $r_n^{con} = R_n^{sec} + (R_n^{sec} - R_{n-1}^{sec})$. Then, this expression is quantized to ensure the learning convergence. Thus, the final reward expression at the nth time step is given by

$$r_n = \begin{cases} -2, & r_n^{con} \leq \eta_1 \\ 0, & \eta_1 < r_n^{con} \leq \eta_2 \\ +1, & \eta_2 < r_n^{con} \leq \eta_3 \\ +2, & r_n^{con} > \eta_3, \end{cases} \tag{15}$$

Note that, to improve the convergence speed of the algorithm, the difference between the rates of two consecutive time steps has been added to the reward in r_n^{con}. This term aims for drawing an effective trajectory step by step towards a maximum secrecy rate by taking the relative rates into account. To be specific, the agent receives lower reward unless it proceeds in a correct trajectory. Furthermore, to avoid very low secrecy rates, we give a relatively large negative value to the reward in case of certain conditions.

Algorithm 2 The DRL-based solution

```
Input : discount factor $\lambda$, soft update coefficient $\tau$, buffer capacity $B$, batch size $T$, actor learning rate $\beta$, critic learning rate $\alpha$
Initialize : randomly initialize the parameters of the four networks $\mu(s;\boldsymbol{\theta}_\mu)$, $Q(s,a;\boldsymbol{\theta}_q)$, $\mu^*(s;\boldsymbol{\theta}_{\mu^*})$, $Q^*(s,a;\boldsymbol{\theta}_{q^*})$, empty the experience replay buffer.
for episode = 1 : $Z$ do
    Initialize a random process $\mathcal{N}$ for action exploration,
    Receive the initial observations,
    for n = 1 : $J$ do
        1. Select activated elements based on Algorithm 1.
        2. Obtain full channels using (5)-(7).
        3. Select the action $a_n = \mu(s_n;\boldsymbol{\theta}_\mu) + \mathcal{N}$.
        4. Store $a_n$ into the matrix $\boldsymbol{\Psi}$.
        5. Calculate the reward via (15) and store the transition $\{s_n, a_n, r_n, s_{n+1}\}$.
        6. Sample a mini-batch with size $T$ from the replay buffer as $\{s_j, a_j, r_j, s_{j+1}\}$.
        7. Determine the target Q-value via (16).
        8. Update $Q(s,a;\boldsymbol{\theta}_q)$ by minimizing (17).
        9. Update $\mu(s_n;\boldsymbol{\theta}_\mu)$ using the sampled policy gradient in (18).
        10. Update the target networks using (14).
    end for
end for
```

D. The overall algorithm

At first, four NNs are generated; the actor evaluation network $\boldsymbol{\theta}_\mu$, the critic evaluation network $\boldsymbol{\theta}_q$, the actor target network $\boldsymbol{\theta}_\mu^*$ and the critic target network $\boldsymbol{\theta}_q^*$. The parameters of these networks are uniformly distributed. An experience replay with capacity B is built where the training batches are selected from. The batch size is denoted by T. In each episode, at first based on **Algorithm 1**, the two activated IRS elements are chosen. Then, the pilot signals are received and two elements of the correlation matrices $\mathbf{C}_r$, $\mathbf{C}_t$ and $\mathbf{C}_e$ are updated. Based on the correlation matrices at hand, the full channels are estimated via (5)-(7). Afterwards, taking the state s_n, the actor evaluation network determines an action a_n which is reformed into the IRS reflection matrix $\boldsymbol{\Psi}$, based on which the reward is received and the agent transitions to the next state. The transition $\{s_n, a_n, r_n, s_{n+1}\}$ is then stored in the replay buffer. The critic evaluation network samples a mini-batch with size T from the replay buffer to calculate the target Q-value q_j as

$$q_j = \begin{cases} r_j, & j = T, \\ r_j + \lambda Q^*(s_{j+1}, \mu^*(s_{j+1}; \boldsymbol{\theta}_{\mu^*}); \boldsymbol{\theta}_{q^*}), & j < T. \end{cases} \tag{16}$$

Then, the loss function for the critic evaluation network is

$$L(\boldsymbol{\theta}_q) = \frac{1}{T} \sum_{j=1}^{T} (q_j - Q(s_j, a_j; \boldsymbol{\theta}_q))^2. \tag{17}$$

Afterwards, the critic evaluation network is updated by minimizing the loss function and by using PG the actor evaluation network is updated with the ascend factor

$$\Delta_{\boldsymbol{\theta}_\mu} = \frac{1}{T} \sum_{j=1}^{T} \Big(\nabla_a Q(s_j, \mu(s_j; \boldsymbol{\theta}_\mu); \boldsymbol{\theta}_q)|_{s=s_j, a=\mu(s_j)} \nabla_{\boldsymbol{\theta}_\mu} \mu(s_j; \boldsymbol{\theta}_\mu)|_{s_j} \Big) \tag{18}$$

In the end, the target networks are updated via soft update in (14). The overall algorithm is described in **Algorithm 2**, in which Z stands for the number of all episodes and J is the maximum number of time steps in each episode.

IV. Numerical Results

In this section, the performance of the proposed DRL-based scheme is evaluated. Here, the channels $\bar{\mathbf{h}}_t$, $\bar{\mathbf{h}}_r$ and $\bar{\mathbf{h}}_e$ are generated based on the DeepMIMO dataset in [19]. The 'O1' dataset is adopted with the parameters in Table I. The per-dataset normalization method is used, where all samples are normalized by the maximum absolute value over the entire dataset. This method keeps the distance information encoded in the multi-path signature. The location of the IRS and the transmitter are fixed, while the receiver and the eavesdropper are chosen randomly. Please refer to [12] for more details.

In the DRL algorithm, the maximum number of time steps per episode is $J = 200$. The size and number of layers in each NN depends on the size of action and observation spaces. The actor network contains an input layer with $(5M + 4)$ neurons, followed by five hidden layers, each with $f_a(i)$, $i = 1, ..., 5$ neurons, and an output layer with M neurons, which suits the number of actions. The activation function for the hidden layers is relu while it is tanh for the output layer. The critic network includes two input layers, one the same size as the observation space ($5M + 4$) and the other the size of the action space (M), a concatenation layer for the two input layers, a hidden layer with $10M$ neurons followed by F additional hidden layers with $f_c(i)$, $i = 1, ..., F$ neurons. All hidden layers have the relu activation function. A single neuron output layer determines the Q-value for the given action. We set the activation function of the output layer as "None", which describes the linear activation. Additional parameters of the NN are given in Table II. All NNs use the Adam optimizer. The learning rate for the actor network is $\beta = 0.001$ while it is $\alpha = 0.002$ for the critic network. The experience replay buffer capacity is $B = 50000$ and the batch size is $T = 64$. The action noise is a complex Gaussian process with an initial standard deviation of 0.6, which is decreased in each time step with a factor of 0.9999 until it reaches 0.05. Other variables of the network include the discount factor $\lambda = 0.95$, the soft update coefficient $\tau = 0.005$, and the reward thresholds in (15) defined as $[\eta_1, \eta_2, \eta_3] = [0, 2, 4]$ for the perfect CSI scenario and $[0, 1.2, 3]$ for the imperfect CSI case. These thresholds are set based on the long-term behaviour of the system.

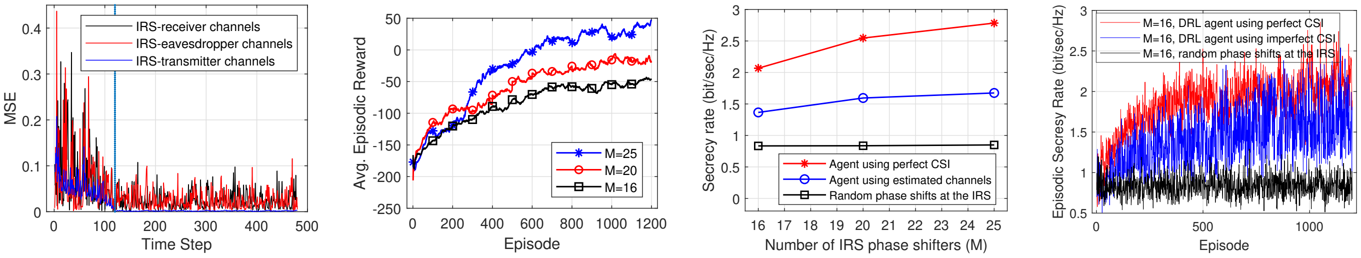
Deep MIMO dataset parameters	Value
frequency band	3.5 GHz
base station (IRS)	3
user rows (receiver and eavesdropper)	400 to 800
active user (transmitter)	row 850 column 90
user antenna shape	$(1, 1, 1)$
user antenna rotation	$[0, 30]$, $[30, 60]$, $[60, 90]$
IRS antenna shape	$(1, M_y, M_z)$
antenna spacing	0.5λ
system bandwidth	100 MHz
number of OFDM subcarriers	16
OFDM limit	1
number of paths	3

Table I: The adopted deep MIMO dataset parameters

M	Parameters of NNs	Value
16	$\{f_a(1), ..., f_a(5)\}$	$\{3M, 3M, 3M, 3M, 3M\}$
20	$\{f_a(1), ..., f_a(5)\}$	$\{3.5M, 3M, 3M, 3M, 3M\}$
25	$\{f_a(1), ..., f_a(5)\}$	$\{4M, 3M, 3M, 3M, 3M\}$
16 and 20	$\{f_c(1), ..., f_c(5)\}$	$\{125, 65, 35, 20, 10\}$
25	$\{f_c(1), ..., f_c(6)\}$	$\{200, 125, 65, 35, 20, 10\}$

Table II: Parameters of NNs

In Fig. 2.a the estimation errors of the estimated channels are depicted in the first 500 time steps. In other words, this figure depicts $\frac{1}{M} \sum_{j=1}^{M} |\mathbf{h}_i(j) - \bar{\mathbf{h}}_i(j)|^2$, where $i \in \{e, r, t\}$ displays the channel nodes. Here, an IRS with the shape $M_y = M_z = 4$ and total number of elements of $M = 16$ is used. As explained in **Algorithm 1**, it takes $\frac{M^2 - M}{2} = 120$ steps for the algorithm to fully update the correlation matrices. As depicted here, by the 120th time step, which is shown with a dotted vertical line, the correlation matrices are fully updated for the first time, and the estimation error is bounded by a certain value. Note that due to the mobility of the receiver and the eavesdropper, the IRS-transmitter channel estimation error, after time step 120, is much less than the IRS-receiver or the IRS-eavesdropper channels. Fig. 2.b shows the average episodic reward over a window of 100 episodes in a total of 1200 episodes. The IRS in each scenario is equipped with $M \in \{4 \times 4, 4 \times 5, 5 \times 5\}$ elements. Note that by increasing M, the size of our action space, observation space and NN becomes larger, thus, there is a trade-off between performance and complexity. To this end, for our system model, we chose the above-mentioned values of M, which provide enough degrees of freedom for the system to operate well and have a limited complexity. For any larger values of M, the system performance converges, while the complexity increases. This can be validated by Fig. 2.c where the average secrecy rate over the last 200 episodes is depicted versus number of IRS elements. This figure demonstrates the robustness of our



(a) Channel estimation error. (b) Average episodic reward. (c) Achievable secrecy rate. (d) Episodic secrecy rate.

Figure 2: Simulation Results.

design, by comparing the performance of the agent in case perfect CSI is available with the case where CSI is obtained via **Algorithm 1**. Furthermore, it is shown that a secure wireless communication system can be achieved by only optimizing the passive beamforming at the IRS, regardless of the active beamforming at the transmitter. Note that the transmitter can also have a secure physical layer beamforming individually which enhances the system performance even more. The achieved performance here is in case the transmitter uses an insecure beamforming. Finally, Fig. 2.d demonstrates the rising trend of the secrecy rate in each episode, which shows that the agent is learning to perform the best action over time. The agent behaves the same when $M = 20$ or $M = 25$.

V. Conclusion

In this paper, secure wireless communication with an standalone IRS is investigated. To do so, the IRS is equipped with two active elements, which can acquire a partial channel between the IRS and the transmitter, receiver, or eavesdropper nodes. We then map the partial channels to full channels by using the channels' correlation matrix. Finally by designing a DRL-based framework using the DDPG technique, we design a reflection matrix that maximizes the secrecy rate.

References

[1] C. Pan, H. Ren, K. Wang, W. Xu, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Multicell MIMO Communications Relying on Intelligent Reflecting Surfaces," *IEEE Transactions on Wireless Communications (Early Access)*, pp. 1–1, May 2020.

[2] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep Reinforcement Learning-Based Intelligent Reflecting Surface for Secure Wireless Communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, Jan. 2021.

[3] M. Cui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface ," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1410–1414, Oct. 2019.

[4] Y. Omid, S. M. M. Shahabi, C. Pan, Y. Deng, and A. Nallanathan, "A Trellis-based Passive Beamforming Design for an Intelligent Reflecting Surface-Aided MISO System ," *IEEE Communications Letters*, MAr. 2022.

[5] Y. Omid, S. M. Shahabi, C. Pan, Y. Deng, and A. Nallanathan, "Low-Complexity Robust Beamforming Design for IRS-Aided MISO Systems with Imperfect Channels," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1697–1701, May 2021.

[6] G. Zhou, C. Pan, H. Ren, K. Wang, and A. Nallanathan, "Intelligent Reflecting Surface Aided Multigroup Multicast MISO Communication Systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3236–3251, 2020.

[7] B. Zheng and R. Zhang, "Intelligent Reflecting Surface-Enhanced OFDM: Channel Estimation and Reflection Optimization," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 518–522, Apr. 2020.

[8] Z. Wang, L. Liu, and S. Cui, "Channel Estimation for Intelligent Reflecting Surface Assisted Multiuser Communications: Framework, Algorithms, and Analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6607–6620, Oct. 2020.

[9] S. Liu, Z. Gao, J. Zhang, M. D. Renzo, and M. S. Alouini, "Deep Denoising Neural Network Assisted Compressive Channel Estimation for mmWave Intelligent Reflecting Surfaces," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9223–9228, Aug. 2020.

[10] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling Large Intelligent Surfaces With Compressive Sensing and Deep Learning," *IEEE Access*, vol. 9, pp. 44 304–44 321, 2021.

[11] ——, "Deep Learning for Large Intelligent Surfaces in Millimeter Wave and Massive MIMO Systems," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2019.

[12] A. Taha, Y. Zhang, F. B. Mismar, and A. Alkhateeb, "Deep Reinforcement Learning for Intelligent Reflecting Surfaces: Towards Standalone Operation," *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.

[13] S. Zhang, S. Zhang, F. Gao, J. Ma, and O. A. Dobre, "Deep Learning Optimized Sparse Antenna Activation for Reconfigurable Intelligent Surface Assisted Communication," *IEEE Transactions on Communications*, July 2021.

[14] S. M. Kay, "Fundamentals of statistical signal processing," *Prentice Hall Signal Processing Series*, p. 303, 1993.

[15] M. Gui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1410–1414, Oct. 2019.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, feb 2015.

[17] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 1057–1063, 2000.

[18] 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," *[Online]. Available: arXiv:1910.02398*, 2015.

[19] A. Alkhateeb, "DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications," *Proc. of Information Theory and Applications Workshop (ITA), San Diego, CA*, pp. 1–8, Feb. 2019. [Online]. Available: https://www.deepmimo.net/.