

Article

Learning-Based Multi-Domain Anti-Jamming Communication with Unknown Information

Yongcheng Li ¹, Jinchi Wang ² and Zhenzhen Gao ^{2,*} 

¹ State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang 471003, China; lyncan@163.com

² School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China; wjc12345@stu.xjtu.edu.cn

* Correspondence: zhenzhengao@xjtu.edu.cn

Abstract: Due to the open nature of the wireless channel, wireless networks are vulnerable to jamming attacks. In this paper, we try to solve the anti-jamming problem caused by smart jammers, which can adaptively adjust the jamming channel and the jamming power. The interaction between the legitimate transmitter and the jammers is modeled as a non-zero-sum game. Considering that it is challenging for the transmitter and the jammers to acquire each other's information, we propose two anti-jamming communication schemes based on the Deep Q-Network (DQN) algorithm and hierarchical learning (HL) algorithm to solve the non-zero-sum game. Specifically, the DQN-based scheme aims to solve the anti-jamming strategies in the frequency domain and the power domain directly, while the HL-based scheme tries to find the optimal mixed strategies for the Nash equilibrium. Simulation results are presented to validate the effectiveness of the proposed schemes. It is shown that the HL-based scheme has a better convergence performance and the DQN-based scheme has a higher converged utility of the transmitter. In the case of a single jammer, the DQN-based scheme achieves 80% of the transmitter's utility of the no-jamming case, while the HL-based scheme achieves 63%.

Keywords: anti-jamming communication; game theory; Deep Q-Network



Citation: Li, Y.; Wang, J.; Gao, Z. Learning-Based Multi-Domain Anti-Jamming Communication with Unknown Information. *Electronics* **2023**, *12*, 3901. <https://doi.org/10.3390/electronics12183901>

Academic Editor: Dimitri I. Kaklamani

Received: 26 July 2023

Revised: 13 September 2023

Accepted: 13 September 2023

Published: 15 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the broadcast nature of wireless communications, wireless transmissions are vulnerable to various security threats such as eavesdropping, jamming attacks and so on. Jamming attacks could cause serious effects on the communication quality [1], therefore, the study of anti-jamming techniques has become one of the critical topics in wireless communications. Frequency hopping communication is an effective technique to combat jamming attacks and has been widely used in military and civilian communications. The traditional approaches mainly focus on changing the frequency table and frequency hopping bandwidth. Since the spectrum resources are limited, in addition to the anti-jamming techniques in the frequency domain, anti-jamming techniques in the power domain have also been investigated [2–4]. More and more research tries to design anti-jamming strategies in multiple domains jointly to enhance the wireless communication [5,6].

With the development of cognitive technology and artificial intelligence, a smart jammer, which can actively detect the legitimate communication and adaptively adjust its jamming policy [7], will cause a great threat to the current anti-jamming technologies. Since the smart jammer and the transmitter have to adjust their transmit strategies based on their sensing results, it is important to model the competition between the jammer and the transmitter. Non-cooperative game theory and the powerful concepts of equilibrium strategies are natural tools to study such problems [2,8]. In methodology, the schemes based on game theory require that the legitimate transmitter needs to know the jamming information such as jamming patterns and parameters. However, in practical wireless

networks, this requirement is difficult to be fulfilled, especially when a smart jammer exists and causes dynamic and intelligent jamming attacks.

Recently, reinforcement learning (RL) has attracted much attention in the anti-jamming research area [9,10]. The anti-jamming schemes based on RL first distinguish different jamming patterns by learning the jamming environment and then carry out the anti-jamming strategies for each jamming pattern. However, new challenges arise when a smart jammer does not obey a certain jamming pattern. To deal with smart jamming attacks, game theory has been used to model the competition between the smart jammer and the transmitter, but it is necessary to find a new approach to solve the game problem when the jamming information is unavailable. In this paper, we try to tackle this game problem by exploiting RL methods.

1.1. Related Work

In order to effectively defend the communication against the attack of the malicious jammer, many related techniques have been proposed [7,11–13].

Frequency hopping (FH) is a commonly used technique to counteract jamming attacks [14]. The optimal frequency hopping rate was investigated in [15] to resist tracking jamming, considering detection expense and detection errors. In [16], an adaptive chaotic frequency hopping scheme was proposed to enhance anti-jamming capabilities. Anti-jamming techniques in the power domain are regarded as the most direct and effective anti-jamming schemes and have been widely used [8]. In recent work, an intelligent anti-jamming algorithm based on Slot Cross Q-Learning (SCQL) was proposed in [17] to deal with the time-varying jamming environment where the jamming channel changed rapidly. In addition to the anti-jamming techniques in a certain single domain, in [5], an anti-jamming scheme in the spectrum–power domain has been proposed, which formulated a power control game and utilized a multi-armed bandit-based method to select the communication channel. An anti-jamming scheme based on the joint use of FH and the transmission rate adaptation technique was proposed in [18]. It was proved that multi-domain anti-jamming schemes perform better than single-domain anti-jamming schemes [5,18].

When resisting the malicious attack of a smart jammer, which can sense the legitimate communication and adaptively adjust its jamming policy based on the sensing results, game theory is usually used to model the competition between the legitimate devices and the jammer [19]. A power control Stackelberg game was formulated in [12] to resist a smart jammer. Further, a time–power domain anti-jamming strategy using a Stackelberg game has been proposed in [6] for wireless relay networks.

Many studies based on game theory consider the strategy with perfect information, which is not realistic in practical communication systems. A power control scheme based on a Stackelberg game was discussed in [7] to defend against intelligent jamming with observation errors. The Bayesian game theory is a common method used to deal with incomplete information [8]. In [20], a Bayesian Stackelberg game was formulated to counteract the jamming of a smart jammer with multiple antennas. Authors in [2] proposed an anti-jamming Bayesian Stackelberg game in which utility functions were defined over statistics to describe incomplete information and only distribution information was required. In [21], a multi-domain anti-jamming scheme was proposed based on a Bayesian Stakelberg game with imperfect information, which included observation errors and the bounded rationality of the jammer. By using backward induction, the closed-form solution in the time domain has been derived in [21]. When it is difficult to derive the optimal solutions of the Stackelberg game, the hierarchical learning (HL) algorithm has been used in [22–24] to find the mixed strategy of a Nash equilibrium (NE) point. Specifically, in [22,23], authors have investigated the anti-jamming problem with discrete power and found the mixed policies in the power domain by using the HL algorithm. In [24], the HL algorithm has been used to solve the capacity offloading problem over unlicensed band for two-tier dual-mode small cell networks.

When the jamming information was unavailable, authors in [25] proposed a method based on maximum likelihood estimation to obtain the parameter of jamming. RL techniques can be used to achieve an optimal communication policy via trial-and-error without being aware of the jamming and network model [26]. A two-dimensional anti-jamming mobile communication scheme has been proposed by applying RL techniques in [27] to obtain an optimal policy without any knowledge about the jamming models.

However, most of the existing work focused on the case of fixed jamming patterns, for example, a two-dimensional anti-jamming communication for a fixed jamming pattern based on the Deep Q-Network (DQN) was proposed in [28]. The existence of a smart jammer and unavailability of the jamming information could cause fatal challenges to the existing anti-jamming schemes.

1.2. Contribution

Different from the existing related work, which assumes the availability of the jamming strategies or jamming patterns, in this paper, we try to solve the multi-domain anti-jamming problem caused by smart jammers, which adjust their jamming strategies adaptively and are not bound to a certain jamming pattern. We firstly use the game theory to model the competition between the transmitter and the jammers. Then, due to the unavailability of the jamming information, we propose anti-jamming communication schemes by exploiting reinforcement learning methods. The contributions of this paper can be summarized as follows:

- A non-zero-sum game model is used to formulate the competition between the legitimate transmitter and the smart jammers. Two learning-based schemes are proposed to solve the frequency–power domain anti-jamming communication game under the assumption that the information about the jammers is unavailable to the transmitter.
- An anti-jamming scheme based on the DQN algorithm is proposed to optimize the transmit channel and transmit power, while another anti-jamming scheme based on the HL algorithm is proposed to solve the mixed strategy for the Nash equilibrium. Simulation results show that the HL-based anti-jamming scheme has the best converge performance among the learning-based schemes and the DQN-based scheme achieves the largest utility value for the transmitter compared to the anti-jamming communication schemes.

1.3. Organization

The remainder of this paper is organized as follows. We give the problem formulation and present the anti-jamming game model in Section 2. The considered multi-domain anti-jamming issue is solved and two anti-jamming schemes based on the DQN algorithm and the HL algorithm are presented in Section 3. Simulation results and discussions are given in Section 4. Finally, Section 5 concludes this paper.

2. System Model and Anti-Jamming Game with Smart Jammers

2.1. System Model

Consider an anti-jamming communication system shown in Figure 1, where the communication between the transmitter at the source node (S) and the receiver at the destination node (D) is maliciously interfered with by N smart jammers ($N \geq 1$). In this system, the transmitter employs frequency hopping, which involves changing its operating frequency over time. In addition to the frequency hopping, the transmitter has the capability to adjust its transmit power to encounter the jamming. The jammers are smart in the sense that they can sense and analyze the transmissions and adaptively adjust their jamming strategies to achieve a better jamming effect.

As shown in Figure 1, the jammers are denoted as $J_n, n \in \{1, 2, \dots, N\}$, the distance between S and D is d_S and the distance between J_n and D is d_{J_n} . There are M channels that S can use to communicate with D . The set of all usable channels is denoted by $\mathcal{F} \in \{f_1, f_2, \dots, f_M\}$. The set of usable transmission powers at S is represented by $\mathcal{P} = \{P_1, P_2, \dots, P_{L_S}\}$, and the set of usable jamming powers can be written as

$\Phi = \{\phi_1, \phi_2, \dots, \phi_{L_J}\}$, where L_S and L_J are the number of the transmission power levels and the interference power levels, respectively. During the electronic countermeasures process, S and the jammers can adjust their strategies including the communication/jamming channel and the transmit/jamming power.

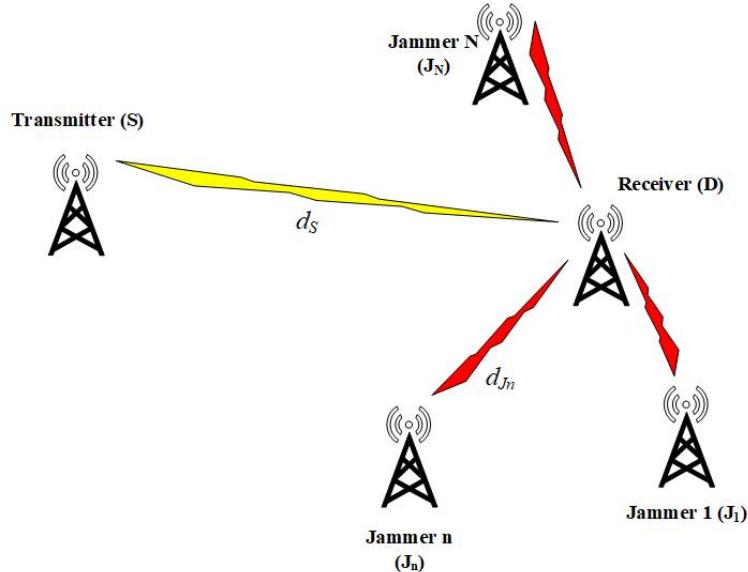


Figure 1. System model with multiple smart jammers.

2.2. Game Formulation

The competition between S and the jammers can be modeled as a game. In the game, S tries to optimize its transmit strategies in the frequency domain and the power domain to maximize the transmission utility, while the jammers try to adjust their jamming strategies in the frequency domain and the power domain adaptively to degrade the transmission utility. The signal to interference plus noise ratio (SINR) is usually used as the performance metric when formulating the utility functions [22,28] since SINR directly reflects the communication performance such as error rate and transmission rate. Considering the power cost of the transmitter, the utility function of S denoted by μ_S can be defined as

$$\mu_S = \frac{P_i h_S}{\sigma^2 + \sum_{n=1}^N \phi_{i_{J_n}} h_{J_n} f(f_m = f_{j_{J_n}})} - c_S P_i, \quad (1)$$

where $P_i \in \mathcal{P}$ is the transmit power of S and $i \in [1, L_S]$, $f_m \in \mathcal{F}$ is the transmission channel of S and $m \in [1, M]$, $\phi_{i_{J_n}} \in \Phi$ is the jamming power of J_n and $i_{J_n} \in [1, L_J]$ and $f_{j_{J_k}} \in \mathcal{F}$ is the jamming channel of J_n and $j_{J_n} \in [1, M]$. $f(\xi)$ is an indicator function, which equals 1 if ξ is true and 0 otherwise. h_S and h_{J_n} are the channel gains of the $S - D$ link and $J_n - D$ link, respectively, and σ^2 is the noise power. c_S denotes the transmission cost per unit power of S .

Given the transmit channel set \mathcal{F} and the transmit power set \mathcal{P} , the optimization problem at S can be written as:

$$\max_{f_m \in \mathcal{F}, P_i \in \mathcal{P}} \mu_S. \quad (2)$$

The goal of the jammers is to damage the normal communication between S and D . In order to obtain a better jamming effect, the jammers cooperate with each other to make jamming decisions. Therefore, the jammers share a common utility function μ_J . When the SINR

information and the channel gain information can be available at the jammers, considering the jamming power cost, we can model the utility function of the jammers as follows

$$\mu_J = -\frac{P_i h_S}{\sigma^2 + \sum_{n=1}^N \phi_{i_{j_n}} h_{J_n} f(f_m = f_{j_{j_n}})} - \sum_{n=1}^N c_J \phi_{i_{j_n}}, \quad (3)$$

where c_J represents the jamming cost per unit power of the jammers. From the perspective of the jammer $J_n, n \in \{1, N\}$, given the jamming channel set \mathcal{F} and the jamming power set Φ , the optimization problem at the jammers can be written as:

$$\max_{f_{j_{j_n}} \in \mathcal{F}, \phi_{i_{j_n}} \in \Phi} \mu_J. \quad (4)$$

The interaction of S and the jammers is now modeled as a non-zero-sum game. If S and the jammers can obtain perfect information about each other, the mixed strategy Nash equilibrium can be achieved. However, in real electronic countermeasures scenarios, it is difficult for S and the jammers to obtain each other's information.

3. Anti-Jamming Communication Scheme without Opponents' Information

When it is challenging for the transmitter to acquire the jamming information such as the jammers' strategies, the DQN algorithm has been used recently to solve the anti-jamming problem without being aware of the jamming strategies [28,29]. Based on the DQN algorithm, a two-dimensional anti-jamming communication scheme that combines the frequency domain and the spatial domain has been proposed in [28]. In [29], a frequency-power domain anti-jamming communication scheme for fixed jamming patterns is proposed based on the deep double-Q learning algorithm.

Inspired by the aforementioned methods, to deal with the smart jammers, we propose a decision-making network scheme based on the DQN algorithm. Based on the non-zero-sum game model, we design a DQN decision-making network for S and each jammer, and the anti-jamming decision is made through learning the environment and the historical information.

3.1. The DQN-Based Scheme for Multi-Domain Anti-Jamming Strategies

3.1.1. The Process of the Proposed Scheme

The proposed scheme based on the DQN algorithm includes the agents for S and each smart jammer. Each agent independently trains its DQN decision-making network and makes decisions based on the local information, which means S and the jammers do not share any information about each other's strategies. Without loss of generality, we only describe the network of S here, since the agent at S and each jammer is similar. Specifically, we define s_S^k as the state of the agent of S in the k -th time slot, which is used to describe the local environment of S . The agent of S utilizes the ϵ -greedy policy to take an action a_S^k only based on its own information, s_S^k . The ϵ -greedy policy is a trade-off between exploration and exploiting, employed to balance between exploring new actions and exploiting the existing knowledge. The actions of S and each jammer are executed in parallel.

To calculate the reward, D feeds the received SINR to S by broadcasting. The agent of S obtains the SINR and the transmit power and calculates the reward r_S^k , which will be given later. The SINR broadcast by D can also be heard by the jammers, so the agent of each jammer can calculate its reward $r_J^n, \forall n \in \{1, \dots, N\}$. It should be noted that the jammers cooperate with each other to degrade the transmission utility. Due to the cooperation between the agents of the jammers, all jamming agents share a common reward r_J .

When the agent of S moves to the next state s_S^{k+1} , it obtains an experience of $e_S = (s_S^k, a_S^k, r_S^k, s_S^{k+1})$. The agent of S stores its experiences in its own experience pool M_S , and after the experience pool is full, a mini-batch is sampled from it to update the neural network, which is known as experience replay [30] and used to reduce the data

correlation. In the process of learning, the Q-function $Q(s_S^k, a_S^k, \theta_S^k)$ represents the long-term reward after the action a_S^k is executed under the state s_S^k and θ_S^k is the weight vector of the DQN. It is known that the structure of a double neural network has a better and stable performance on the training process [30]. In the proposed scheme, the agent of S has both the train DQN and the target DQN with the weight vectors θ_S^k and $\hat{\theta}_S^k$, respectively. In the k -th time slot, the agent of S randomly selects a mini-batch M_S^k with B experiences from the experience pool M_k and uses the stochastic gradient algorithm to minimize the prediction error between the train DQN and the target DQN. As a loss function, the prediction error is given as

$$L(\theta_S^k) = \frac{1}{2B} \sum_{e_S \in M_S^k} (r_S^k + \delta \max_{a_S^{k+1}} Q(s_S^{k+1}, a_S^{k+1}, \hat{\theta}_S^k) - Q(s_S^k, a_S^k, \theta_S^k))^2, \quad (5)$$

where δ is the discount factor.

Finally, by using the gradient descent optimizer to minimize the loss function, the gradients to update the weights of the train DQN are given as

$$\frac{\partial L(\theta_S^k)}{\partial \theta_S^k} = \frac{1}{B} \sum_{e_S \in M_S^k} (r_S^k + \delta \max_{a_S^{k+1}} Q(s_S^{k+1}, a_S^{k+1}, \hat{\theta}_S^k) - Q(s_S^k, a_S^k, \theta_S^k)) \nabla Q(s_S^k, a_S^k, \theta_S^k), \quad (6)$$

where $\hat{\theta}_S^k$ is updated by $\hat{\theta}_S^k = \theta_S^k$ per T_{step} . The structure of the proposed DQN algorithm is shown in Figure 2.

A similar DQN algorithm is also performed at $J_n, \forall n \in \{1, \dots, N\}$. In the k -th time slot, the agent of $J_n (\forall n \in \{1, \dots, N\})$ obtains its state, which is responsible for describing the local environment. Then, each jamming agent individually chooses an action according to the local information, known as state, and they execute in parallel. The only difference is that all the jamming agents calculate a common reward by using the feedback SINR of D as well as the jamming powers shared among the jammers. The DQN algorithm of each jamming agent is the same as shown in Figure 2, and we will not describe it again. The process of the proposed DQN-based scheme is shown in Algorithm 1.

Algorithm 1 The pseudocode of the proposed DQN-Based Scheme

Without loss of generality, the DQNs of S and $J_n, \forall n \in \{1, \dots, N\}$ are illustrated in the following:

Set up DQNs for both S and J_n , set empty experience pools M_S and M_{J_n}

Initialize the train DQN with random weights for S and J_n

Initialize the target DQN with weights $\hat{\theta}_S^k = \theta_S^k$ and $\hat{\theta}_{J_n}^k = \theta_{J_n}^k$ for S and J_n , respectively

Agent of S/J_n chooses an action randomly

Agent of S/J_n stores its experience e_S/e_{J_n} into M_S/M_{J_n} , respectively, until full

Repeat

Agent of S/J_n observes its states $s_S^k/s_{J_n}^k$, respectively, in the k -th time slot

Agent of S/J_n chooses an action $a_S^k/a_{J_n}^k$, respectively

Agent of S calculates reward r_S^k according to the feedback of D

Agent of J_n calculates reward $r_{J_n}^k$ according to the feedback of D and the jamming powers shared among the jammers

Agent of S/J_n obtains the next state $s_S^{k+1}/s_{J_n}^{k+1}$, respectively

Agent of S/J_n stores its experience e_S/e_{J_n} into M_S/M_{J_n} , respectively

Agent of S/J_n samples a mini-batch $M_S^k/M_{J_n}^k$ from M_S/M_{J_n} , respectively

Agent of S/J_n updates the weights of the train DQN $\theta_S^k/\theta_{J_n}^k$, respectively

Agent of S/J_n updates the weights of the target DQN $\hat{\theta}_S^k/\hat{\theta}_{J_n}^k$ with $\theta_S^k/\theta_{J_n}^k$ per T_{step} , respectively

Until convergence

In Algorithm 1, the notation A/B means A or B , the actions $a_S^k, a_{J_n}^k$, the rewards $r_S^k, r_{J_n}^k$ and the states $s_S^k, s_{J_n}^k$ of S and the jammer $J_n, \forall n \in \{1, \dots, N\}$ will be given in detail in the following subsection.

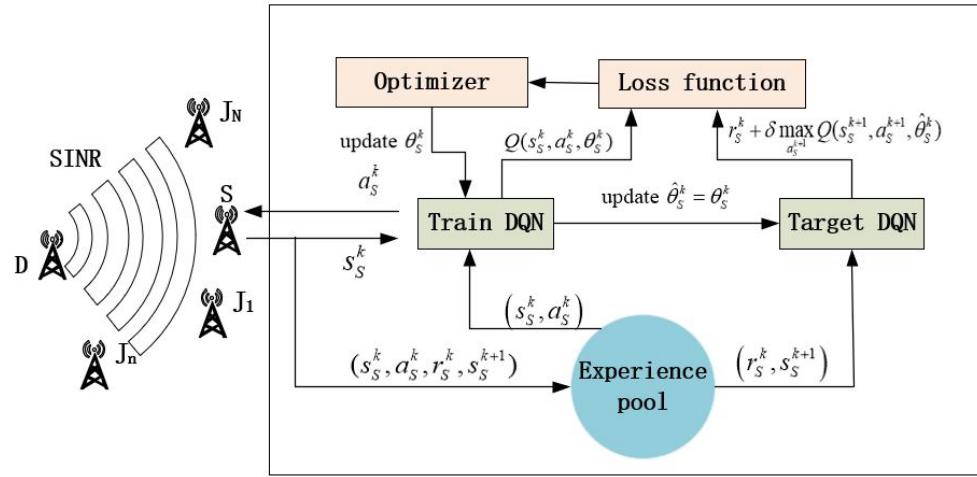


Figure 2. Illustration of the proposed DQN-based scheme.

3.1.2. The Definition of the Action, Reward and State

As described in Section 2, S expects to obtain the transmission channel and transmission power, which can be denoted as $(f_m, P_i) \in \mathcal{F} \times \mathcal{P}$ according to the decision-making network's output, where \times represents the Cartesian product. Therefore, the action of the agent of S needs to indicate the channel and power selection. Denote an action of S in the k -th time slot as a_S^k , which represents a certain choice of (f_m, P_i) . The size of the action space $\mathcal{F} \times \mathcal{P}$ is ML_S , thus an action has ML_S possible choices, and $a_S^k \in [1, 2, \dots, ML_S]$. If the action of the agent of S is $a_S^k = i$, it indicates that S needs to select the transmission channel and power according to the i -th element in $\mathcal{F} \times \mathcal{P}$. Similarly, the action of J_n denoted as $a_{J_n}^k \in [1, 2, \dots, ML_J]$ can be defined in a similar way.

To solve the anti-jamming game in Section 2.2 without requiring the opponents' information, the DQN algorithm is designed to find the optimal transmission strategy, which maximizes the long-term reward of S . Therefore, we utilize the game utility of S as the reward of the agent, that is

$$r_S^k = SINR_D^k - c_S P_i \quad (7)$$

where $SINR_D^k$ is the SINR feedback from D at the k -th time slot. c_S denotes the transmission cost per unit power of S , and P_i is the transmission power chosen by the action a_S^k . The reward of the jammers, r_J^k , is similar to S .

Referring to [28], the state of S is composed of the previous SINR at D and previous actions of S , which is denoted as

$$s_S^k = \{SINR_D^{k-w}, a_S^{k-w}, SINR_D^{k-w+1}, a_S^{k-w+1}, \dots, SINR_D^{k-1}, a_S^{k-1}\}, \quad (8)$$

where $SINR_D^k$ represents the SINR feedback from D during the k -th time slot and w is the number of previous SINRs or actions.

Similarly, we use the game utility as the reward r_J^k of the agent of J_n , which can be calculated as

$$r_J^k = -SINR_D^k - \sum_{n=1}^N c_J \phi_{i_{J_n}}, \quad (9)$$

where c_J denotes the cost per unit power of the jammers and $\phi_{i_{J_n}}$ is the jamming power chosen by J_n 's action $a_{J_n}^k$. The jammers share the jamming powers to calculate the reward r_J^k . In the same way, the state of J_n is defined as

$$s_{J_n}^k = \{SINR_D^{k-w}, a_{J_n}^{k-w}, SINR_D^{k-w+1}, a_{J_n}^{k-w+1}, \dots, SINR_D^{k-1}, a_{J_n}^{k-1}\}. \quad (10)$$

3.2. Hierarchical Learning-Based Scheme for Mixed Strategies

When the information of the opponent is available, it has been proved that the mixed strategy for a Nash equilibrium can be obtained for a game with finite players and a finite-size strategy set [31]. The mixed strategy of S denoted by \mathbf{q} is given as

$$\mathbf{q} = (q_1, q_2, \dots, q_{ML_S}), \quad (11)$$

where q_i represents the probability that S selects the action $a_S^k = i$, which is defined in Section 3.1. Similarly, the mixed strategy of J_n is defined as

$$\boldsymbol{\rho}_n = (\rho_{n,1}, \rho_{n,2}, \dots, \rho_{n,ML_J}), \quad (12)$$

where $\rho_{n,i}$ is the probability that J_n chooses the i -th action from the action space $\mathcal{F} \times \mathcal{P}$. The N jammers cooperate with each other to attack the communication between S and D as much as possible, so we define the mixed strategies for the jamming attack as

$$\boldsymbol{\rho} = (\rho_1, \dots, \rho_n, \dots, \rho_N), \quad (13)$$

which consists of the mixed strategies of the N jammers.

If a pair of mixed strategies $(\mathbf{q}^*, \boldsymbol{\rho}^*)$ constitutes a Nash equilibrium, the mathematical expression for the Nash equilibrium is as follows

$$\mu_S(\mathbf{q}^*, \boldsymbol{\rho}^*) \geq \mu_S(\mathbf{q}, \boldsymbol{\rho}^*), \quad (14)$$

$$\mu_J(\mathbf{q}^*, \boldsymbol{\rho}^*) \geq \mu_J(\mathbf{q}^*, \boldsymbol{\rho}). \quad (15)$$

However, it is impossible to derive the optimal mixed strategies of S when the information of the jammers is unavailable. For this case, we can use learning methods to find the mixed strategies of S to maximize the utility without being aware of the opponent's action spaces.

A hierarchical learning method has been proposed in [22] to obtain the mixed strategies on the power control for a legitimate transmitter and a single jammer. Inspired by the aforementioned work, we proposed an HL-based scheme to obtain the mixed strategies of the players by learning a Q-function for each action and exploiting the participant's own utility feedback.

An agent is designed at S to learn a Q-function for each action and then obtain the mixed strategy \mathbf{q} . For the jammers, an agent can be designed to learn the strategy $\boldsymbol{\rho}$ and instruct all jammers to cooperate in the electronic countermeasures. However, a centralized learning algorithm in this case results in a large action space and much information overhead, which is challenging and complicated in the real wireless environment. Therefore, we set up a local agent for each jammer to have their own strategy $\boldsymbol{\rho}_n$ by using a common feedback to achieve the cooperation among the jammers.

In the proposed HL-based scheme, we firstly use the Q-learning algorithm to obtain the Q-function of the agent's actions, which is used to evaluate the importance of each action and then obtain the mixed strategy through the acquired Q-function. The following is the detailed procedure of the algorithm.

We create agents for S and each jammer. In order to learn the Q-functions of S and the jammer $J_n, \forall n \in \{1, \dots, N\}$, denoted by Q_S and Q_{J_n} , respectively, every agent has a Q-function table to record the Q-value of all possible actions. Meanwhile, the agent of S has a mixed strategy \mathbf{q} and agents of all jammers also have their own strategies $\boldsymbol{\rho}_n, n = 1, \dots, N$. In the k -th time slot, the strategies for S and J_n can be denoted as \mathbf{q}^k and $\boldsymbol{\rho}_n^k$, while the Q-functions are denoted as Q_S^k and $Q_{J_n}^k$. In the k -th time slot, S chooses its action a_S^k according to the strategy \mathbf{q}^k , i.e., the probability distribution for the actions of S . J_n also chooses an action $a_{J_n}^k$ according to its strategy $\boldsymbol{\rho}_n^k$. Then, D feeds the received

SINR by broadcasting to help S in calculating the reward r_S^k in this time slot. Meanwhile, the jammers can also obtain the SINR from the broadcasting. They share their powers with each other so that they can obtain a common reward r_J^k . By using the SINR and the shared information about jamming powers, each jammer can calculate its reward independently. The definition and calculation of the rewards are the same as mentioned in Section 3.1.

After this, the agent of S updates its Q-function Q_S^k to Q_S^{k+1} by the following equation

$$Q_S^{k+1}(a_S^k) = (1 - \alpha^k)Q_S^k(a_S^k) + \alpha^k r_S^k, \quad (16)$$

where $\alpha^k \in [0, 1]$ is the learning rate.

Subsequently, S updates its mixed strategy to q^{k+1} for the next time slot by the equation as follows

$$q_i^k = \frac{e^{\frac{Q_S^k(i)}{\tau_S}}}{\sum_{m=1}^{Mn_S} e^{\frac{Q_S^k(m)}{\tau_S}}}, \quad (17)$$

where τ_S is a parameter to balance exploration and exploitation.

Similarly, the agents of jammers update their Q-function as follows

$$Q_{J_n}^{k+1}(a_{J_n}^k) = (1 - \alpha^k)Q_{J_n}^k(a_{J_n}^k) + \alpha^k r_J^k, \quad n = 1, 2, \dots, N, \quad (18)$$

and the mixed strategies of jammers are updated by the equation as follows

$$\rho_{n,i}^k = \frac{e^{\frac{Q_{J_n}^k(i)}{\tau_J}}}{\sum_{m=1}^{Mn_J} e^{\frac{Q_{J_n}^k(m)}{\tau_J}}}, \quad n = 1, 2, \dots, N, \quad i = 1, \dots, ML_J, \quad (19)$$

where τ_J is a parameter to balance exploration and exploitation. The parameters τ_S and τ_J are important and can influence the performance of the HL-based scheme. Therefore, we will discuss the influence of them in detail in the simulation section.

The process of the proposed HL-based scheme is shown in Algorithm 2.

Algorithm 2 The pseudocode of the HL-Based Scheme

Set up Q-functions $Q_S^k(a_S)$ and $Q_{J_n}^k(a_{J_n})$ for S and J_n , $\forall n \in \{1, 2, \dots, N\}$

Set up mixed strategies q^k and ρ_n^k for S and J_n , respectively

Initialize $Q_S^k(a_S) = 0$ and $Q_{J_n}^k(a_{J_n}) = 0$ for all actions of S and J_n

Initialize q^k and ρ_n^k so that every action can be chosen with equal probability

Repeat

S chooses action a_S^k by q^k in the k -th time slot

J_n chooses action $a_{J_n}^k$ by ρ_n^k in the k -th time slot

S calculates the reward r_S^k according to the feedback SINR of D

J_n calculates the reward r_J^k according to the feedback SINR of D as well as the jamming powers shared among the jammers

S and J_n update their Q-functions via (16) and (18), respectively

S and J_n update their mixed strategies via (17) and (19), respectively

Until convergence

4. Simulation Results

In the following simulations, the channel gains are modeled by the path-loss model, which has been widely used in wireless communications, and the channel gains h_S and h_J can be written as [32]

$$h_l = \left(\frac{c}{4\pi f_0 d_0} \right)^2 \cdot \left(\frac{d_0}{d_l} \right)^\gamma, \quad (20)$$

where $l \in \{S, J_1, \dots, J_N\}$, c is the speed of light, f_0 is the central frequency of the wireless signal, d_0 is the far-field reference distance of the antenna, γ is the path-loss exponent, d_l is the distance of the $S - D$ link for $l = S$ and the distance of the jammers link for $l = J_n, \forall n \in \{1, \dots, N\}$.

There are $M = 5$ available channels that can be used by S . The set of transmit powers at S is $\mathcal{P} = \{1 \text{ W}, 2 \text{ W}, 3 \text{ W}\}$, while the set of jamming powers is $\Phi = \{3 \text{ W}, 5 \text{ W}\}$. To make as successful of an attack as possible, the jammers usually have a larger jamming power. Here, we consider the case that the jammers do not cherish the power as the transmitter and set the transmission cost of S as $c_S = 1$ and the the jamming cost of J_n as $c_J = 0.1$. The details of other simulation parameters can be found in Table 1.

Table 1. The main parameters in the simulation.

Parameter	Value
the distance between S and D , d_S	1000 m
the distance between J_1 and D , d_{J_1}	300 m
the distance between J_2 and D , d_{J_2}	500 m
the distance between J_3 and D , d_{J_3}	500 m
the distance between J_4 and D , d_{J_4}	600 m
the number of usable channels, M	5
the set of usable transmission power, \mathcal{P}	{1 W, 2 W, 3 W}
the set of jamming power for jammer, Φ	{3 W, 5 W}
the noise power, σ^2	-114 dBw
the number of transmission power levels, L_S	3
the number of jamming power levels, L_J	2
the discount of long-term reward, δ	0.5
the far-field reference distance in (20), d_0	20 m
the path-loss exponent in (20), γ	3
the central frequency in (20), f_0	900 MHz

The software and hyperparameters adopted in the DQNs are as follows. The proposed DQN-based scheme was implemented using Python (Version 3.8) and Keras (Version 3.6.0). The number of the neurons of the input layer is $2w$, where the value of w will be discussed later. The numbers of the neurons of the output layer for S and $J_n \forall n \in \{1, \dots, N\}$ are, respectively, $L_S M$ and $L_J M$. The DQNs of S and J_n have two hidden layers, which have 128 and 64 neurons, respectively. The ReLU activation function is applied to each hidden layer. The sizes of experience pool and mini-batch are 500 and 32, and we use the RMSprop optimizer with a learning rate of 0.01. The learning rate decreases during the training process with a learning rate decay of $\frac{1}{1+e^{-4}}$. The parameter that balances the exploration and exploiting in the ϵ -greedy policy of the DQN-based scheme is 0.6, and it decreases with a decay rate of $\frac{1}{1+e^{-4}}$ during the network training until it reaches the minimum value of 0.01. The hard update interval is $T_{\text{step}} = 100$.

4.1. Comparison Schemes

In order to verify the performance of the proposed schemes based on the DQN algorithm and the HL algorithm, the effectiveness of the proposed schemes in learning to maximize their utilities and make decisions on transmit channels and power selecting is validated by comparing it with some other benchmark schemes.

- Q-learning scheme: in addition to the DQN algorithm, Q-learning is another common method used to estimate the value of each action and make a decision by recording the Q-values of all actions in a local table.
- random strategy: randomly choosing all actions with the same probability is a classic and commonly used method to defend against jamming.
- no-jamming: the transmitter and receiver work in the environment without malicious interference from the jammer. This scheme works as the upper bound of the transmitter's utility.

4.2. Discussion on the Parameters of the HL-Based Scheme

In the HL-based scheme, the parameters τ_S and τ_J are utilized to make a balance between exploration and exploitation for S and the jammers to update their mixed strategies. Thus, it is crucial to select an appropriate value for τ_S and τ_J . In the following, we will discuss the influence of the parameters τ_S and τ_J on the performance of the HL-based scheme. We set $N = 1$ and perform extensive simulations for different values of τ_S and τ_J .

Table 2 presents the results of the average utility of the transmitter and the jammers in the scheme based on the HL algorithm with different values of τ_S and τ_J . The results show, when the values are small, such as $\tau_S = 0.3$ and $\tau_J = 0.3$, that the utility of both the transmitter and the jammers fluctuates significantly, where u_S fluctuates between 2.5 and 5 in ten experiments, indicating that the performance of the HL-based scheme is not stable when the parameters τ_S and τ_J are small. When the values of τ_S and τ_J are large, such as $\tau_S = 5$ and $\tau_J = 3$, the average utility of S and the jammers remains stable in ten experiments. Nonetheless, it is questionable that the stable utility values of S and the jammers are good enough.

Table 2. The average utility of the transmitter with different values of τ_S and τ_J .

(τ_S, τ_J)	Experiment									
	1	2	3	4	5	6	7	8	9	10
μ_S	(0.3, 0.3)	5.03	2.52	2.52	5.03	2.51	2.52	2.52	5.03	5.03
	(5, 3)	4.11	4.08	4.07	4.05	4.09	4.04	4.04	4.02	4.04
μ_J	(0.3, 0.3)	-7.37	-3.89	-4.02	-7.37	-4.02	-3.92	-3.82	-7.40	-7.53
	(5, 3)	-6.76	-6.72	-6.70	-6.68	-6.73	-6.67	-6.67	-6.64	-6.67

Figure 3 gives more insights on the influence of the parameters τ_S and τ_J . The mixed strategies of S and the jammers with different values of τ_S and τ_J are shown in the figure. The results show that when τ_S and τ_J are small, for example, $\tau_S = 0.3$ and $\tau_J = 0.3$, the HL-based scheme tends to favor deterministic strategies, indicating that the probability of a certain action is equal to or close to 1 while probabilities of other actions are close to 0. However, when the parameters are larger, such as $\tau_S = 5$ and $\tau_J = 3$, the probability distribution of actions tends to become a uniform distribution, indicating that the mixed strategies obtained by the HL-based scheme become similar to the random strategy. These observations of the HL-based scheme can be explained as follows. When τ_S and τ_J are small, the effect of feedback on the mixed strategies is amplified, leading the HL-based scheme to exploit existing strategies more than exploration of the environment. When τ_S and τ_J are large, the effect of feedback on the strategies decreases and results in exploring new action more than exploiting the known information.

Based on the previous experimental results, it can be observed that when the parameters τ_S and τ_J are small, the HL-based scheme tends to lean towards deterministic strategies, which leads to a decrease in the stability of the utilities. Conversely, when the parameters τ_S and τ_J are large, the mixed strategies tend to follow uniform probability distribution, approaching the random strategy. To select suitable values for τ_S and τ_J , we conduct a search within a certain range. Figure 4 illustrates the average utility values of the transmitter in 20 experiments when τ_S ranges from 0.7 to 2.5 and τ_J ranges from 0.3 to 2.5. It can be observed that the transmitter achieves higher utility when $\tau_S \in [1.2, 1.4]$. Taking into account the previous analysis results, we ultimately choose $\tau_t = 1.2$ and $\tau_j = 0.8$ in the following simulations.

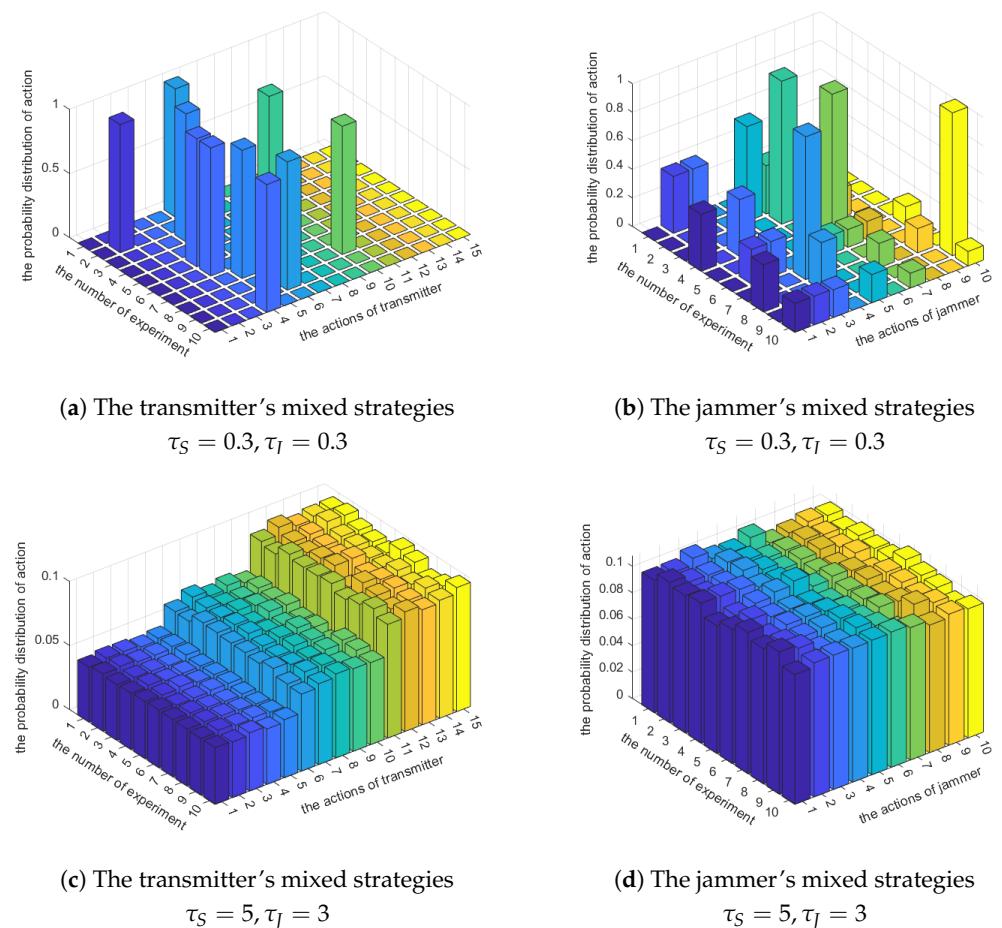


Figure 3. Mixed strategies of the transmitter and the jammer with different values of τ_S and τ_J .

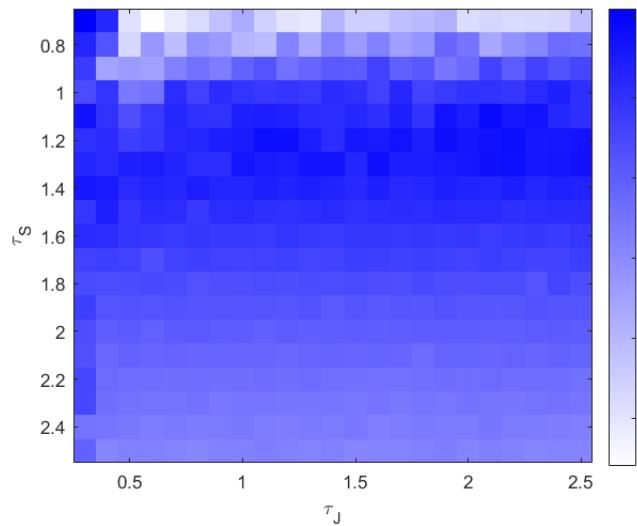


Figure 4. The effect of the parameters τ_S and τ_J on the utility values.

4.3. Performance Comparisons of Different Schemes

In this section, the proposed DQN-based scheme and the HL-based scheme are compared to the comparison schemes in Section 4.1 to validate the effectiveness of the proposed anti-jamming communication schemes with different numbers of jammers. During the simulations, we find out that when $w > 3$ in (8) and (10), the transmitter's utility does not increase significantly, but the complexity increases significantly as w increases. Therefore, in the following simulations, we set $w = 3$.

Figure 5 illustrates the transmitter's utility of different anti-jamming schemes for a single jammer case. The “no-jamming” line represents the maximum utility of the communication system when there is no jamming. As shown in the figure, the DQN-based scheme has the highest utility value among the anti-jamming communication schemes. The utility of the HL-based scheme is higher than that of the commonly used random strategy. Compared with the Q-learning scheme, the HL-based scheme achieves comparable utilities as the Q-learning scheme but has a much faster convergence speed. The HL-based scheme converges after 10,000 time slots, while the Q-learning scheme does not reach convergence until after the 20,000th time slot. The reason is that the size of the Q-value table of the HL-based scheme is only related to the number of the actions that can be used, while the size of the Q-value table of the Q-learning scheme is not only related to the number of possible actions but also related to the state space, which results in a much larger Q-value table than the HL-based scheme and a slower convergence speed.

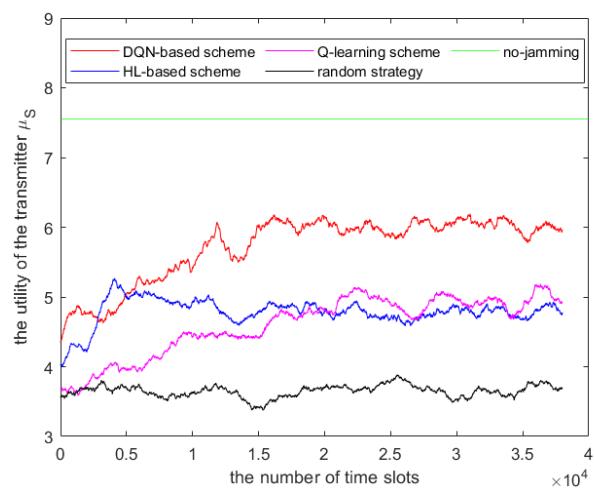


Figure 5. The transmitter's utility comparison of different schemes with a single jammer.

The transmitter's utilities of different anti-jamming communication schemes with two and four jammers are shown in Figure 6. Similar to the case of a single jammer, among the anti-jamming communication schemes, the random strategy has the smallest utility, the DQN-based scheme has the largest utility and the HL-based scheme has the fastest convergence. In addition, as the number of jammers increases, the transmitter's utility decreases for all the schemes. This is because more smart jammers have better jamming capabilities to decrease the transmitter's utility.

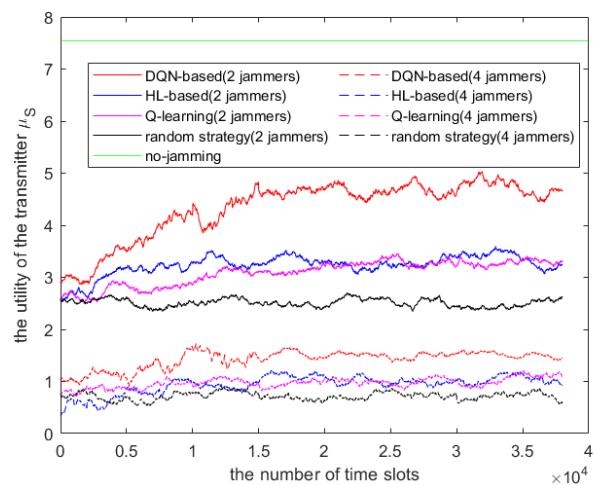


Figure 6. The transmitter's utility comparison of different schemes with multiple jammers.

Table 3 presents the average utility of these five schemes from the 25,000th time slot to the 35,000th time slot. As shown in the table, the random strategy has the smallest average utility. For the single jammer case, the DQN-based scheme achieves 80% of the utility of the case without jamming attack, while the HL-based scheme achieves 63% of the utility of the no-jamming case. Compared to the Q-learning scheme, the utility of the HL-based scheme is slightly less and can achieve more than 97% of the utility of the Q-learning scheme. For all the cases, the DQN-based scheme has the largest average utility among the anti-jamming communication schemes, but the transmitter's utility decreases as the number of the jammers increases.

Table 3. The average utilities of different schemes.

	No-Jammer	DQN-Based Scheme	HL-Based Scheme	Q-Learning Scheme	Random Strategy
$\mu_S(1 \text{ jammer})$	7.55	6.03	4.76	4.89	3.66
$\mu_S(2 \text{ jammers})$	7.55	4.75	3.36	3.28	2.48
$\mu_S(4 \text{ jammers})$	7.55	1.50	0.99	0.97	0.88

4.4. Complexity Analysis of Different Learning-Based Schemes

The structure of the DQN is important because too many neurons will lead to problems such as excessive computational complexity, slow convergence and overfitting, and too few neurons will result in the decrease of performance. We use a fully connected neural network (FCN) for our DQN structure. The neuron numbers of the input and output layers of the network are denoted as N_s and N_a , i.e., the number of elements the DQN's state contains and the size of the action space. There are two hidden layers in the network, which contain 128 and 64 neurons, respectively, because of our focus on the situation with a small action space. The numbers of the hidden layers are designed to avoid overfitting and optimize the performance and the convergence speed. In each time slot, one forward propagation and one backward propagation are required, and we analyze the complexity of the DQN-based scheme by calculating the computational complexity of the forward propagation and the backward propagation. The computational complexity of the forward propagation is mainly related to the number and the size of the hidden layers, and the complexity of the backward propagation is the same as the forward propagation. As mentioned above, the time complexity of the DQN-based scheme is about $O(2^7 N_s + 2^6 N_a + 2^{13})$. As for the HL-based scheme, in each time slot, the HL-based scheme only needs to update the Q-value of the selected action and the mixed strategy, thus the time complexity of the HL-based scheme can be represented as $O(2N_a + 1)$.

For the compared schemes, the random strategy is a fixed strategy, which is the simplest strategy. In each time slot, the Q-learning scheme needs to update the Q-value of the selected state-action pair and transform the state into the index, which is used to look up the Q-value table. Updating the Q-value in each time slot is a fixed procedure, and the complexity is determined by transforming the state into the index related to the Q-value table. Thus, the complexity of the Q-learning scheme is decided by the number of elements of the state and can be represented as $O(N_s + 1)$.

Although the complexity of the DQN-based scheme is relatively high, it has the best performance on the transmitter's utility. The HL-based scheme not only has an affordable complexity but also has a comparable performance to the Q-learning scheme and a faster convergence.

5. Conclusions

In this paper, we considered the anti-jamming problem caused by smart jammers, which can sense the legitimate transmission and adaptively adjust the jamming channel and jamming power. The interaction between the smart jammers and the transmitter has been modeled as a non-zero-sum game in this paper. Since the transmitter is unable to

obtain any information about the smart jammers' jamming strategies or jamming patterns, two anti-jamming communication schemes have been proposed based on DQN and HL algorithms to solve the game problem. The DQN-based scheme solved the anti-jamming strategies in the frequency and power domain, while the HL-based scheme solved the mixed strategy, which gives the probabilities for the actions in the frequency and power domain. The performance of the proposed schemes are investigated carefully through simulations. Compared to the random strategy and the Q-learning scheme, the DQN-based scheme achieves the largest utility of the transmitter, while the HL-based scheme has the fastest convergence among the learning-based schemes. When a single jammer exists, compared to the case of no jamming, the DQN-based scheme achieves 80% of the utility obtained by the no-jamming case.

Author Contributions: Conceptualization, Y.L.; methodology, Z.G. and J.W.; software simulation, J.W.; analysis and validation, J.W. and Z.G.; writing—original draft preparation, J.W.; writing—review and editing, Y.L. and Z.G.; visualization, Y.L. and J.W.; supervision, Z.G.; project administration, Y.L.; funding acquisition, Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Open Research Found of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE) under Grant 2022K0202A and by the National Natural Science Foundation of China under Grant 62071367.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Due to institutional data privacy requirements, our data are unavailable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FH	Frequency hopping
DQN	Deep Q-Network
HL	Hierarchical learning
SINR	Signal to interference plus noise ratio

References

1. Zou, Y.; Zhu, J.; Wang, X.; Hanzo, L. A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. *IEEE Trans. Commun.* **2016**, *10*, 1727–1765. [[CrossRef](#)]
2. Jia, L.; Yao, F.; Sun, Y.; Niu, Y.; Zhu, Y. Bayesian Stackelberg Game for Anti-jamming Transmission With Incomplete Information. *IEEE Commun. Lett.* **2016**, *20*, 1991–1994. [[CrossRef](#)]
3. Li, Y.; Xiao, L.; Liu, J.; Tang, Y. Power control Stackelberg game in cooperative anti-jamming communications. In Proceedings of the 2014 5th International Conference on Game Theory for Networks, Beijing, China, 25–27 November 2014.
4. Xu, Y.; Ren, G.; Chen, J.; Luo, Y.; Jia, L.; Liu, X.; Yang, Y.; Xu, Y. A One-Leader Multi-Follower Bayesian-Stackelberg Game for Anti-Jamming Transmission in UAV Communication Networks. *IEEE Access* **2018**, *6*, 21697–21709. [[CrossRef](#)]
5. Jia, L.; Xu, Y.; Sun, Y.; Feng, S.; Yu, L.; Anpalagan, A. A Multi-Domain Anti-Jamming Defense Scheme in Heterogeneous Wireless Networks. *IEEE Access* **2018**, *6*, 40177–40188. [[CrossRef](#)]
6. Li, Y.; Bai, S.; Gao, Z. A Multi-Domain Anti-Jamming Strategy Using Stackelberg Game in Wireless Relay Networks. *IEEE Access* **2020**, *8*, 173609–173617. [[CrossRef](#)]
7. Xiao, L.; Chen, T.; Liu, J.; Dai, H. Anti-Jamming Transmission Stackelberg Game With Observation Errors. *IEEE Commun. Lett.* **2015**, *19*, 949–952. [[CrossRef](#)]
8. Jia, L.; Xu, Y.; Sun, Y.; Feng, S.; Anpalagan, A. Stackelberg Game Approaches for Anti-Jamming Defence in Wireless Networks. *IEEE Wirel. Commun.* **2018**, *25*, 120–128. [[CrossRef](#)]
9. Naparstek, O.; Cohen, K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 310–323. [[CrossRef](#)]
10. Liu, X.; Xu, Y.; Cheng, Y.; Li, Y.; Zhao, L.; Zhang, X. A heterogeneous information fusion deep reinforcement learning for intelligent frequency selection of HF communication. *China Commun.* **2018**, *15*, 73–84. [[CrossRef](#)]
11. D’Oro, S.; Galluccio, L.; Morabito, G.; Palazzo, S.; Chen, L.; Martignon, F. Defeating Jamming With the Power of Silence: A Game-Theoretic Analysis. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 2337–2352. [[CrossRef](#)]

12. Yang, D.; Xue, G.; Zhang, J.; Richa, A.; Fang, X. Coping with a Smart Jammer in Wireless Networks: A Stackelberg Game Approach. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4038–4047. [[CrossRef](#)]
13. D’Oro, S.; Ekici, E.; Palazzo, S. Optimal Power Allocation and Scheduling Under Jamming Attacks. *IEEE/ACM Trans. Netw.* **2017**, *25*, 1310–1323. [[CrossRef](#)]
14. Noori, H.; Vilni, S.S. Defense Against Intelligent Jammer in Cognitive Wireless Networks. In Proceedings of the 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 30 April–2 May 2019; pp. 1309–1314.
15. Zhu, Y.; Yu, L.; Zhu, Y.; Jia, L. Optimal Frequency Hopping Rate for Anti-follower Jamming with Detection Error. *J. Signal Process.* **2018**, *34*, 824–832.
16. Mansour, A.E.; Saad, W.M. Adaptive Chaotic Frequency Hopping. In Proceedings of the 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 23–24 December 2015.
17. Niu, Y.; Zhou, Z.; Pu, Z.; Wan, B. Anti-jamming Communication using Slotted Cross Q learning. *Electronics* **2023**, *12*, 2879. [[CrossRef](#)]
18. Hanawal, M.K.; Abdel-Rahman, M.J.; Krunz, M. Game Theoretic Anti-jamming Dynamic Frequency Hopping and Rate Adaptation in Wireless Systems. In Proceedings of the 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Hammamet, Tunisia, 12–16 May 2014; pp. 247–254.
19. Han, Z.; Niyato, D.; Saad, W.; Basar, T.; Hjorungnes, A. *Game Theory in Wireless and Communication Networks*; Cambridge University Press: Cambridge, UK, 2012.
20. Shen, Z.; Xu, K.; Xia, X. Beam-Domain Anti-Jamming Transmission for Downlink Massive MIMO Systems: A Stackelberg Game Perspective. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2727–2742. [[CrossRef](#)]
21. Li, Y.; Li, K.; Gao, Z.; Zheng, C. A Multi-Domain Anti-Jamming Scheme Based on Bayesian Stackelberg Game With Imperfect Information. *IEEE Access* **2022**, *10*, 132250–132259. [[CrossRef](#)]
22. Jia, L.; Yao, F.; Sun, Y.; Xu, Y.; Feng, S.; Anpalagan, A. A Hierarchical Learning Solution for Anti-Jamming Stackelberg Game With Discrete Power Strategies. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 818–821. [[CrossRef](#)]
23. Yao, F.; Jia, L.; Sun, Y.; Xu, Y.; Feng, S.; Zhu, Y. A Hierarchical Learning Approach to Anti-jamming Channel Selection Strategies. *Wirel. Netw.* **2019**, *25*, 201–213. [[CrossRef](#)]
24. Sun, Y.; Shao, H.; Qiu, J.; Zhang, J.; Sun, F.; Feng, S. Capacity Offloading in Two-tier Small Cell Networks over Unlicensed Band: A Hierarchical Learning Framework. In Proceedings of the 2015 International Conference on Wireless Communications & Signal Processing (WCSP), Nanjing, China, 15–17 October 2015; pp. 1–5.
25. Wu, Y.; Wang, B.; Liu, K.J.R.; Clancy, T.C. Anti-Jamming Games in Multi-Channel Cognitive Radio Networks. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 4–15. [[CrossRef](#)]
26. Adem, N.; Hamdaoui, B. Jamming Resiliency and Mobility Management in Cognitive Communication Networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
27. Xiao, L.; Jiang, D.; Xu, D.; Zhu, H.; Zhang, Y.; Poor, H.V. Two-Dimensional Anti-jamming Mobile Communication Based on Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2018**, *67*, 9499–9512. [[CrossRef](#)]
28. Han, G.; Xiao, L.; Poor, H.V. Two-dimensional Anti-jamming Communication based on Deep Reinforcement Learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
29. Nguyen, P.K.H.; Nguyen, V.H.; Do, V.L. A Deep Double-Q Learning-based Scheme for Anti-Jamming Communications. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–22 January 2021.
30. Mnih, V.; Kavukcuoglu, K.; Silver, D. Human-level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
31. Fudenberg, D.; Tirole, J. *Game Theory*; MIT Press: Cambridge, MA, USA, 1991; pp. 24–30.
32. Tse, D.; Viswanath, P. *Fundamentals of Wireless Communication*; Cambridge University Press: Cambridge, UK, 2005; pp. 10–48.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.