Anti-Jamming Communications Using Spectrum Waterfall: A Deep Reinforcement Learning Approach

Xin Liu, Yuhua Xu, *Member, IEEE*, Luliang Jia, *Student Member, IEEE*, Qihui Wu, *Senior Member, IEEE*, and Alagan Anpalagan, *Senior Member, IEEE*

***Abstract*— This letter investigates the problem of anti-jamming communications in a dynamic and intelligent jamming environment through machine learning. Different from existing studies which need to know (estimate) the jamming patterns and parameters, we use the temporal and spectral information, i.e., the spectrum waterfall, directly. First, to cope with the challenge of infinite state of spectrum waterfall, a recursive convolutional neural network is designed. Then, an anti-jamming deep reinforcement learning algorithm is proposed to obtain the optimal anti-jamming strategies. Finally, simulation results validate the proposed approach. The proposed algorithm does not need to model the jamming patterns, and naturally has the ability to explore the unknown environment, which implies that it can be widely used for combating dynamic and intelligent jamming.**

***Index Terms*— Anti-jamming, deep Q-network, deep reinforcement learning.**

I. Introduction

ANTI-JAMMING has always been an active research topic, as wireless transmissions are naturally vulnerable to jamming attacks. The mainstream anti-jamming techniques include Frequency Hopping Spread Spectrum (FHSS) and Direct-Sequence Spread Spectrum (DSSS) [1]. Recently, to address the interactions between the legitimate users and the jammers, game theory has been widely applied in the literature [2]–[6]. In methodology, these approaches need to know the jamming strategies, which implies that the legitimate users are required to estimate the jamming patterns and parameters from the observed environment. However, with the rapid development of artificial intelligence and universal software radio peripheral (USRP) [8], the jammers can easily create dynamic and intelligent jamming attacks. As a consequence, there are two limitations with regard to estimation-based anti-jamming communications: i) there may be information loss for unknown jamming patterns, and ii) if the intelligent jammer switches its strategies dynamically and intelligently, it is not possible to track and react it in real time. Thus, it is challenging and interesting to investigate anti-jamming communication approaches in dynamic and unknown environment.

To overcome the above limitations, a promising way is to design new anti-jamming approaches that utilize the spectrum information with temporal features, which is known as spectrum waterfall [9], without estimating jamming patterns and parameters. These kinds of anti-jamming approaches would avoid information loss and adapt to the dynamic environment, as can be expected. In addition, reinforcement learning (RL) is an effective way to solve the decision problems in dynamic environment. The widely used technique is Q-learning [10], which has been used in anti-jamming problems [2], [3], [7]. Unfortunately, Q-learning is not able to deal with the spectrum waterfall directly because of the infinite environment state.

Motivated by the deep reinforcement learning (DRL) technique for learning successful control policies from raw video data in [11], we investigate the anti-jamming problem in dynamic and intelligent jamming environment. First, the waterfall spectrum is defined as the state of the environment to avoid losing the jammer information. Then, a recursive convolutional neural network (RCNN) is designed to realize the direct processing of complex environmental state. Finally, an anti-jamming deep reinforcement learning algorithm (ADRLA) is proposed. Simulation results show that the proposed ADRAL achieves near optimal performance in various scenarios. The main contributions are summarized as follows:

- Based on the deep reinforcement learning technique, a smart anti-jamming communication scheme is proposed. In particular, the spectrum waterfall is defined as a state, which describes the detail features of jammer more accurately.
- The proposed algorithm is relying only on the locally observed information and does not need to estimate the jamming patterns and parameters of the jammer in advance, i.e., it is model-free, which can be widely used in various anti-jamming scenarios.

Note that the most related work is [12], which also adopted deep reinforcement learning to investigate the anti-jamming problems. The main differences in this work are as follows: i) the environment state is presented by extracting features of signal-to-interference-plus-noise ratio (SINR) and primary user occupancy in [12], while it is the raw spectrum information in this work, and ii) it requires the jammer to have the

Manuscript received February 12, 2018; accepted March 8, 2018. Date of publication March 12, 2018; date of current version May 8, 2018. This work was supported in part by the Guang Xi Universities Key Laboratory Fund of Embedded Technology and Intelligent System (Guilin University of Technology), in part by the Natural Science Foundation for Distinguished Young Scholars of Jiangsu Province under Grant BK20160034, in part by the National Natural Science Foundation of China under Grant 61771488, Grant 61671473, and Grant 61631020, and in part by the Open Research Foundation of Science and Technology on Communication Networks Laboratory. The associate editor coordinating the review of this paper and approving it for publication was F. Wang. *(Corresponding author: Yuhua Xu.)*

X. Liu is with the College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China (e-mail: liuxin2017125@glut.edu.cn).

Y. Xu and L. Jia are with the College of Communication Engineering, Army Engineering University of PLA, Nanjing 210007, China, and also with the Science and Technology on Communication Networks Laboratory, Shijiazhuang 050002, China (e-mail: yuhuaenator@gmail.com; jiallts@163.com).

Q. Wu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: wuqihui2014@sina.com).

A. Anpalagan is with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: alagan@ee.ryerson.ca).

Digital Object Identifier 10.1109/LCOMM.2018.2815018

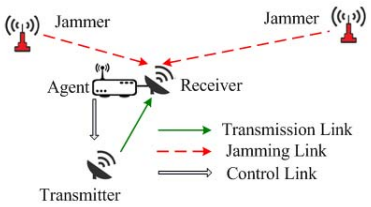

Fig. 1. System model.

same channel-slot transmission structure with the users in [12]. On the contrary, this requirement does not hold in our work, which makes the proposed approach more general.

II. System Model and Problem Formulation

We consider the transmission of one user (a transmitter-receiver pair) against one or several jammers, as shown in Fig. 1.[1] At time t, under the guidance of the agent, the user chooses an frequency, denoted by $f_t \in [f_L, f_U]$ to send signals with a given power $p_u = \int_{-b_u/2}^{b_u/2} U(f)df$, where $U(f)$ and b_u respectively denote the power spectral density (PSD) function and bandwidth of baseband signal of user, f_L and f_U respectively indicate the starting and termination frequency of the communication band of user. Jammer j can arbitrarily select frequency denoted by f_t^j and waveform denoted by baseband PSD function $J_t^j(f)$. Let g_u denote the channel power gain from the transmitter to the receiver, and g_j denote the gain from the jammer j to the receiver. The received SINR of user can be expressed as:

$$\beta(f_t) = \frac{g_u p_u}{\int_{f_t-b_u/2}^{f_t+b_u/2} \left\{ n(f) + \sum_{j=1}^{J} g_j J_t^j (f - f_t^j) \right\} df}, \quad (1)$$

where $n(f)$ is the PSD function of noise. Let β_{th} denote the required SINR threshold for successful transmission, the normalized transmission rate is defined as $\mu(f_t) = \delta(\beta(f_t) \geq \beta_{th})$, where $\delta(x) = 1$ if x is true, otherwise $\delta(x) = 0$.

The agent, which is disposed at the receiving end, continuously senses the whole communication band. Considering the coexistence of jamming and user signals, the PSD function at the receiving end can be expressed as:

$$S_t(f) = g_u U(f - f_t) + \sum_{j=1}^{J} g_j J_t^j (f - f_t^j) + n(f). \quad (2)$$

The discrete spectrum sample value is defined as $s_{i,t} = 10\log[\int_{i\Delta f}^{(i+1)\Delta f} S(f + f_L)df]$, where Δf is the resolution of spectrum analysis. Agent determines the transmission frequency based on the spectrum vector $\mathbf{s}_t = \{s_{t,1}, s_{t,2}, \ldots, s_{t,N}\}$, and sends the results to the transmitter through a reliable control link.

As the environment is unknown and dynamic, it is impossible to obtain the available frequencies directly from environment $\mathbf{s}_t$. Reinforcement learning shows strong learning ability against sweeping and tracking jamming [3], [7]. This is principally because the jamming actions can be obtained from the previous state in the above scenarios, and the anti-jamming problem can be modeled as Markov decision process (MDP) [3]. For more complex jamming patterns, i.e., intelligent jamming, jamming actions may be related to earlier historical state, so we define the environment state as $\mathbf{S}_t = \{\mathbf{s}_t, \mathbf{s}_{t-1}, \ldots, \mathbf{s}_{t-T+1}\}$, where T denotes the number of historical states of backtracking. $\mathbf{S}_t$ is a two-dimensional matrix with a size of $T \times N$, and the thermodynamic chart of $\mathbf{S}_t$ matrix is called spectrum waterfall, which contains both frequency and time domain information.

[1] In this letter, we mainly develop the deep reinforcement learning architecture for anti-jamming communications, and we will consider scenarios with multiple users in the near future.

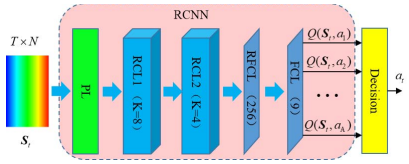

Fig. 2. The network structure of RCNN.

At last, in our anti-jamming MDP, $\mathbf{S} \in \{\mathbf{S}_1, \mathbf{S}_2, \ldots\}$ is the temporal spread environment state, $a \in \{f_1, f_2, \ldots, f_A\}$ is the frequency action of user, $P(\mathbf{S}'|\mathbf{S}, a)$ is the transition probability from the current state $\mathbf{S}$ to $\mathbf{S}'$ when taking action a, and r is the immediate reward defined as:

$$r(a_t) = \mu(a_t) - \lambda\delta(a_t \neq a_{t-1}), \quad (3)$$

where λ denotes the cost for frequency switching.

III. Anti-Jamming Communication Scheme

Since the environment state is dynamically changed according to the unknown probability $P(\mathbf{S}'|\mathbf{S}, a)$ and the size of the state-action space is very large in our anti-jamming MDP, motivated by the DRL technique in [11], we use a deep convolutional neural network (CNN) to approximate the Q-function of each state-action pair, which is the expected discounted long-term reward for state $\mathbf{S}$ and action a, i.e.,

$$Q^*(\mathbf{S}, a) = E\left\{ r + \gamma \max_{a'} Q^*(\mathbf{S}', a') \mid \mathbf{S}, a \right\}, \quad (4)$$

where $\mathbf{S}'$ is the next state if user takes action a at state $\mathbf{S}$, and γ is the discount factor.

However, traditional CNN is generally used for image processing, and its computational complexity is relatively large. Considering the recursion characteristic of the spectrum waterfall, i.e., $\mathbf{S}_{t+1} = \{\mathbf{s}_{t+1}, \mathbf{S}_t^-\}$, where $\mathbf{S}_t^- = \{\mathbf{s}_t, \mathbf{s}_{t-1}, \ldots, \mathbf{s}_{t-T+2}\}$, a recursive convolutional neural network (RCNN) is designed to reduce the computational complexity as shown in Fig. 2. Different from the CNN in [11], RCNN adds one preprocessing layer (PL), replaces the convolution layers (CL) with the recursive CL (RCL), and replaces the full connection layer (FCL) with recursive FCL (RFCL).

PL Operation: The function of PL is to filter out noise and reduce unnecessary computation, and the specific operation is expressed as:

$$\tilde{s}_{i,t} = \begin{cases} s_{i,t} & s_{i,t} \geq n_{th} \\ 0 & s_{i,t} < n_{th}, \end{cases} \quad (5)$$

where n_{th} is the noise threshhold.

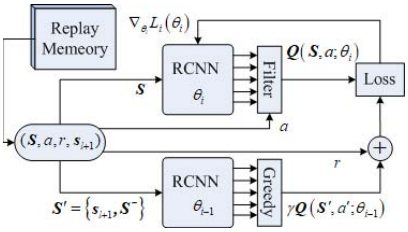

Fig. 3. Updating process of network weights.

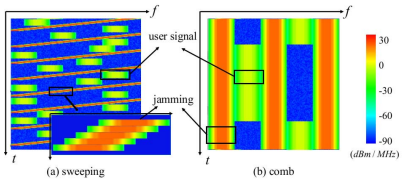

Fig. 4. Graphical state in the presence of fixed jamming pattern.

RCL Operation: Let $\mathbf{X}_t$, $\mathbf{Y}_t$ and $\mathbf{W}_C$ denote the input, ouput and weights of CL respectively, then $\mathbf{Y}_t = \mathbf{X}_t \otimes \mathbf{W}_C$. Taking into account the recursive characteristic of the input, that is, $\mathbf{X}_{t+1} = \{\mathbf{x}_{t+1}, \mathbf{X}_t^-\}$, hence $\mathbf{Y}_{t+1} = \{\mathbf{x}_{t+1}, \mathbf{X}_t^-\} \otimes \mathbf{W}_C$. According to the operation of convolution, the recursive formula of RCL is given by:

$$\mathbf{Y}_{t+1} = \{\{\mathbf{x}_{t+1}, \mathbf{x}_t, \ldots, \mathbf{x}_{t-K+2}\} \otimes \mathbf{W}_C, \mathbf{Y}_t^-\}, \tag{6}$$

where K is the kernel size of convolver. Hence, RCL has only K/T of the ordinary CL computation.

RFCL Operation: Let $\mathbf{X}_t$, $\mathbf{Y}_t$ and $\mathbf{W}_F$ denote the input, ouput and weights of FCL respectively, then $\mathbf{Y}_t = \mathbf{W}_F \mathbf{X}_t$ and $\mathbf{Y}_{t+1} = \mathbf{W}_F \mathbf{X}_{t+1}$. Let $\Delta\mathbf{X}_{t+1} = \mathbf{X}_{t+1} - \mathbf{X}_t$, then the recursive formula of RFCL can be derived as:

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t + \mathbf{W}_F \Delta\mathbf{X}_{t+1}. \tag{7}$$

According to the recursive characteristic of the input, $\Delta\mathbf{X}_{t+1} = \{\mathbf{x}_{t+1} - \mathbf{x}_t, \mathbf{x}_t - \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-T+2} - \mathbf{x}_{t-T+1}\}$, which reflects the changes of spectrum of adjacent time. As long as some frequency points remain unchanged at some time, the computational complexity of FCL can be reduced.

As the above derivation is based on the fact that $\mathbf{W}_C$ and $\mathbf{W}_F$ are constant, the recursive operation in the training phase needs to be closed, but the normal convolution and full connection operation is adopted. As shown in Fig. 3, experiences $e_t = (\mathbf{S}_t, a_t, r_t, \mathbf{s}_{t+1})$ at each time-step t is stored in data set $D_t = (e_1, \ldots e_t)$, and target values are constructed as $\eta_i = r + \gamma \max_{a'} Q(\mathbf{S}', a'; \theta_{i-1})$ by randomly choosing elements in a uniform distribution $e \sim U(D)$, where θ_i is the parameter of Q-network at iteration i. By assuming that η_i is the expected output of RCNN with network weight θ_i when the input is $\mathbf{S}$, we calculate the difference between real output $Q(\mathbf{S}, a; \theta_i)$ and target value η_i to determine the update of network parameters. Finally, the gradient of the loss function with respect to the weight is given by [11]:

$$\nabla_{\theta_i} L_i(\theta_i) = E_e\left[(\eta_i - Q(\mathbf{S}, a; \theta_i))\, \nabla_{\theta_i} Q(\mathbf{S}, a; \theta_i)\right]. \tag{8}$$

Therefore, the network weight θ_i can be updated based on the gradient descent algorithm. At last, the proposed algorithm for anti-jamming communication based on deep reinforcement learning is presented in Algorithm 1.

Algorithm 1: Anti-jamming Deep Reinforcement Learning Algorithm (ADRLA)

Initialize: Set $D = \emptyset$, $i = 0$, θ_0 with random weights, Taining=**True**, $\mathbf{S}_1 = O(T \times N)$, Close recursive operation.
For $t = 1, 2, \cdots, \infty$ **do**
  If Training **then**
    Choose a_t via the ϵ-greedy algorithm
  Else
    Open recursive operation
    Select $a_t = \arg\max_a Q(\mathbf{S}_t, a; \theta)$
  End If
  Execute action a_t and compute r_t and sense $\mathbf{s}_{t+1}$
  Store $(\mathbf{S}_t, a, r, \mathbf{s}_{t+1})$ in D, update $\mathbf{S}_{t+1} = \{\mathbf{s}_{t+1}, \mathbf{S}_t^-\}$
  If $Sizeof(D) > \mathcal{N}$ **and** Training
    Sample random minibatch of transistions $(\mathbf{S}, a, r, \mathbf{s}_{l+1})$ from D, $\mathbf{S}' = \{\mathbf{s}_{l+1}, \mathbf{S}^-\}$
    Compute $\eta = r + \gamma \max_{a'} Q(\mathbf{S}', a'; \theta_i)$
    Compute $\nabla_{\theta_i} L(\theta_i)$, update θ_i, and $i = i + 1$
    If $i > Iterations_{Max}$ **then** Training=**Fasle**.
  End If
End For

IV. Numerical Results and Discussions

In the simulation setting, the user and the jammer combat with each other in a frequency band of 20MHz. The user performs a full band sensing every 1ms with $\Delta f = 100$kHz and retains the spectrum data within the 200ms. Hence, the size of matrix $\mathbf{S}_t$ is 200×200. The bandwidth of user signal is 4MHz, and the center frequency is allowed to change in each 10ms with the step of 2MHz, which means $A = 9$. Both signal and jamming are raised cosine waveforms with roll-off factor $\alpha = 0.5$, in which jamming power is 30dBm and signal power is 0dBm. The demodulation threshold β_{th} at all frequency is set to be 10dB, and the cost of action change λ is set to be 0.2. According to the above setting, the peak calculation requirement of ADRLA is 4.456×10^9 floating-point operations per second (FLOPS). The ordinary multi-core CPU can basically meet this requirement, not to mention the GPU that can reach the capability of 10^{12} FLOPS.

Four kinds of jamming scenarios are considered for simulation: i) Sweeping jamming (sweeping speed is 1GHz/s); ii) Comb jamming (three fixed frequency signals at 2MHz, 10MHz, and 18MHz); iii) Dynamic jamming (selects sweeping and comb jamming pattern randomly); iv) Intelligent comb jamming (selects the frequencies of comb jamming by counting the probability of user's actions in past 100ms). For all jamming patterns, the instantaneous bandwidth of the each jamming tone is set to be 4MHz.

For illustration and presentation, we first give the converging thermodynamic chart of state in the presence of fixed jamming patterns in Fig. 4. It is seen from the figure that the sweeping jamming appears as a diagonal line due to the linear frequency

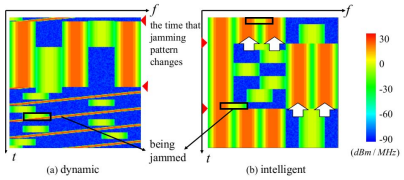

Fig. 5. Graphical state in the presence of dynamic and intelligent jamming.

TABLE I

THE PERFORMANCE COMPARISON BETWEEN THE ADRLA AND THE Q-LEARNING

Scenario	ADRLA		Q-Learning		Optimal
	Throughput	Iterations	Throughput	Iterations	
Sweeping	0.77	5000	0.78	12000	0.8
Comb	0.94	5000	0.94	12000	1.0
Dynamic	0.81	10000	0.75	∞	0.9
Intelligent	0.85	10000	0.59	∞	1.0

change and the comb jamming appears as several vertical stripes corresponding to the several fixed frequency tones. In addition, the user signal was presented by a rectangular block, which reflects the selected frequency of the user at different times. It is noted that there is no overlap between the user signal and the jamming signal, which shows the anti-jamming ability of the proposed ADRLA in the fixed jamming mode.

Secondly, the converging thermodynamic chart of state in the presence of dynamic and intelligent jamming scenarios are given in Fig. 5. The red triangles indicate the time that jamming pattern changes, and the black rectangular blocks are the user signal being jammed. It is shown that although the jamming pattern changes dynamically and intelligently, the user can also avoid jamming effectively, which again validates that the proposed ADRLA has excellent learning ability in dynamic environment. In particular, the intelligence of ADRLA exhibits more incisively and vividly, when again with intelligent jamming. As shown in the white arrow in Fig. 5(b), the jamming frequency in the next period is inclined to the positions where the user signal appears often in the previous period, according to the intelligent jamming pattern. Although this jamming rule is not modeled and estimated by the proposed ADRLA in advance, the agent seems to understand it by learning, and switchs to positions being selected less in the previous period. More interestingly, the new actions of user not only avoid jamming, but also guide the actions of jammer at the next moment as much as possible by occupying multiple jamming frequency points, which can increase the prediction probability of jamming action.

Finally, the performance comparison between the ADRLA and the Q-learning in [7] in four scenarios are given in Table I. For Q-learning, the state is defined as $\{C_1, C_2, C_3, C_4, C_5\}$, where $C_n \in \{0, 1\}$ represents the occupancy of jamming in each channel, and the action set and immediate rewards are the same as ADRLA. As shown in Table I, the column indicated by throughput shows the average throughput performance after convergence (if the algorithm does not converge, the average throughput after 20000 iterations is taken), the column indicated by iterations shows the minimum number of iterations required for convergence (∞ indicates that the corresponding algorithm can not converge) and the last column shows the optimal performance when jamming actions are completely known. Although the performance of Q-learning is almost the same as that of ADRLA except convergence rate in the fixed jamming mode, the performance gap becomes obvious in dynamic or intelligent jamming mode. Q-learning can not converge in those senarios. That is, ADRLA has better learning ability compared with Q-learning, and can achieve near optimal results in various environments.

V. CONCLUSION

In this letter, we investigated the anti-jamming problem in dynamic and intelligent jamming environment. Aiming at employing the waterfall spectrum information directly, we constructed a recursive convolutional neural network to handle the complex interactive decision-making problem with infinite number of states. Then, an anti-jamming deep reinforcement learning algorithm was proposed. Using the proposed learning algorithm, the user is able to learn the best anti-jamming strategy by constantly trying various actions and sensing the spectrum environment. Simulation results in various scenarios are presented to validate the proposed anti-jamming communication approach. Future work on designing multi-user anti-jamming deep reinforcement learning algorithms is ongoing.

REFERENCES

[1] L. Zhang, Z. Guan, and T. Melodia, "United against the enemy: Anti-jamming based on cross-layer cooperation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5733–5747, Aug. 2016.

[2] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.

[3] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 4–15, Jan. 2012.

[4] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2247–2259, Sep. 2016.

[5] L. Xiao, T. Chen, J. Liu, and H. Dai, "Anti-jamming transmission Stackelberg game with observation errors," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 949–952, Jun. 2015.

[6] L. Jia, F. Yao, Y. Sun, Y. Niu, and Y. Zhu, "Bayesian Stackelberg game for antijamming transmission with incomplete information," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1991–1994, Oct. 2016.

[7] S. Machuzak and S. K. Jayaweera, "Reinforcement learning based anti-jamming with wideband autonomous cognitive radios," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–5.

[8] H. Zhu, C. Fang, Y. Liu, C. Chen, M. Li, and X. S. Shen, "You can jam but you cannot hide: Defending against jamming attacks for geo-location database driven spectrum sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2723–2737, Oct. 2016.

[9] W. Chen and X. Wen, "Perceptual spectrum waterfall of pattern shape recognition algorithm," in *Proc. IEEE ICACT*, Jan./Feb. 2016, pp. 382–389.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[11] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[12] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE ICASSP*, Mar. 2017, pp. 2087–2091.