Spiking Mean Field Multi-Agent Reinforcement Learning for Dynamic Resources Allocation in D2D Networks

1st Pei-Gen Ye
School of Computer Science and Cyber Engineering
Guangzhou University
Guangzhou, China
ypgmhxy@gmail.com

2th Wei Liang
School of Computer Science and Engineering
Central South University
Changsha, China
eeveeweily@gmail.com

3th Qiang Lu
School of Computer Science and Engineering
Central South University
Changsha, China
luqiang@jridge.com

4th Rong-Fang Xiao
School of Computer Science and Engineering
Central South University
Changsha, China
summerrobert2018@gmail.com

5th Zhong-Yong Guo
School of Computer Science and Engineering
Central South University
Changsha, China
bensnowguo@gmail.com

6th Kai-Xiang Sun
School of Computer Science and Engineering
Central South University
Changsha, China
sundevin806@gmail.com

***Abstract*—Device-to-device (D2D) technology has been widely used to alleviate the mobile traffic explosion due to its ability of direct communications between proximal devices. However, in practice, available spectrums are limited and the number of D2D and cellular users are rapidly increasing, which greatly decreases the efficiency of resource allocation. For this propose, we train spiking neural network (SNN) with deep reinforcement learning for channel selection and power control. At the same time, we use spatio-temporal backpropagation to accelerate SNN training. When the number of D2D users increases dramatically, the learning rate becomes intractable due to the curse of the exponential growth of action space. Therefore, we apply mean field multi-agent reinforcement learning (MFRL) to approximate interactions within the D2D users. After that, we combine different reinforcement learning algorithms with MFRL. The simulation result shows that, compared with actor-critic (AC) and proximal policy optimization (PPO), spiking actor-critic (S-AC) and spiking proximal policy optimization (S-PPO) can achieve faster convergence rate, higher access rate and better time-averaged overall throughput as well as lower collision probability even when the action space is increased. Besides, when the number of D2D users increases, our spiking mean field proximal policy optimization (SMF-PPO) can achieve better performance than AC, PPO, S-AC and S-PPO.**

***Index Terms*—Spiking neural network, Deep reinforcement learning, Multi-agent reinforcement learning, Device-to-device, Channel selection, Power control.**

I. Introduction

NOWADAYS, the rapid growth of mobile devices brings great challenge to the existing wireless communication system. Fortunately, with the development of device-to-device (D2D) technology, mobile devices can forego routing information through the base station (BS) and send the information directly to neighboring devices. Compared with traditional cellular technologies, D2D technology can not only effectively release the burden on the BS through traffic offloading [1], but also has the potential to increase the data rate and network spectrum efficiency, while reducing network latency [2] for users who are in close proximity.

Generally, the communication of D2D networks reuse the spectrum of the cellular network. In actual applications, there are two main reusing modes: namely overlay mode [3], [4] and underlay mode [5], [6]. In the overlay mode, a small part of the cellular spectrum licensed to the cellular operator is reserved, and the D2D users (D2DUs) is only allowed to transmit data on the reserved spectrum. In this mode, D2DUs and CU users (CUs) use different spectrum resources. Although few interference, it is not only difficult to allocate the spectrum reasonably for CUs and D2DUs, but also reduces the effective utilization rate of the spectrum. In the underlay mode, CUs and D2DUs are allowed to transmit on the same spectrum, and the spectrum remains indivisible, which means that D2DUs must reuse the resources allocated to the CUs, thereby improving spectrum utilization. Compared with the overlay mode, the underlay mode may have interference between the D2D pairs and the CUs, but it do not need additional cellular spectrum division. The key to this mode is how to deal with the harmful

This work was supported by the overseas joint training program for postgraduates of Guangzhou University.

978-1-6654-6667-7/21/$31.00 ©2021 IEEE
DOI 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00024

interference which has a negative impact on the performance of the D2D network when too many D2DUs reuse the same channel [7]–[10]. On the other hand, due to the limitations of energy efficiency and battery life [11], [12], the performance of D2D communication depends to a large extent on proper transmit power control. Therefore, it is important to allocate appropriate transmit channel and power for D2DUs to make a tradeoff [13] between the interference mitigation and effective energy utilization.

In recent years, many research solutions have been proposed to solve the problem of channel selection and power control in D2D networks. For example, Yuan *et al.* [14] proposed the local CSI-based distributed channel-power allocation scheme to enable D2DUs to perform channel selection and power updating independently and iteratively. In [15], a joint power and channel allocation algorithm has been proposed, where the optimal power allocation is derived on each sub-channel and the optimal channel allocation scheme is obtained by the Hungarian algorithm. In particular, Abrardo *et al.* [16] and Lyu *et al.* [17] used game theory based distributed algorithms to maximize the spatial reuse and optimize the channel allocation for the D2D network. However, the resource allocation, at any time, for one D2D user is explicitly reliant on the state of all other D2DUs in the network at that time. And this is an intractable mixed integer non-linear programming (MINLP) problem, the above methods can only find the near-optimal solutions, and there are many obstacles in the process of applying to the actual scene.

Fortunately, the emergence of multi-agent reinforcement learning (MARL) [18], a sub-field of reinforcement learning (RL), has proved to perform exceptionally well on extremely complex control tasks and MARL makes it possible to learn the best strategy for D2D networks. In MARL, each agent maximizes the cumulative return that can be obtained during the learning process, which is affected by the changes in the global state of the environment brought about by the joint actions of other agents. Li *et al.* [19] proposed a distributed spectrum allocation framework that shares global historical status and policies during centralized training to further optimize system performance. At the same time, a fully decentralized soft MARL algorithm has been proposed in [20], which extends the soft actor-critic framework to to find the optimal policy. Besides, in order to minimize the long-term system cost in energy-harvesting D2D network, Huang *et al.* [21] proposed multi-agent deep deterministic policy gradient.

The optimality of RL, however, comes at a high-energy cost. Given that the growing complexity of environment is hard to be continuously offset by equivalent increases in on-board energy sources, there is an unmet need for low-power solutions. With the development of the third generation artificial neural network, namely spiking neural network (SNN) [22], more and more scholars begin to apply it to different fields [23]. SNN is an emerging brain-inspired alternative architecture to deep neural networks in which neurons compute asynchronously and communicate through discrete events called spikes. SNN has two main advantages: first, it can reduce energy consumption by transmitting information through a single bit; second, it has high robustness because of the high connection ratio of neurons and the existence of membrane potential threshold.

Although the above methods can get the optimal channel and power, there are still two aspects worth researching. First, the optimality of RL comes at a high-energy cost. Considering that the increasingly complex situation of D2D network is difficult to continuously offset by the same increase in on-board energy, there is an unmet need for low-power solutions for resource allocation. With the development of the third generation artificial neural network, namely spiking neural network (SNN) [22], more and more scholars begin to apply it to different fields [23].

SNN is an emerging brain-inspired alternative architecture to deep neural networks (DNN) in which neurons compute asynchronously and communicate through discrete events called spikes. SNN has two main advantages: first, it can reduce energy consumption by transmitting information through a single spike; second, it has high robustness because of the high connection ratio of neurons and the existence of membrane potential threshold. To address the limitations of SNN in solving high-dimensional continuous control problems, one approach is to combine the energy-efficiency of SNN with the optimality of deep reinforcement learning (DRL) [24]. Therefore, a popular SNN construction method [25] is to directly convert a trained deep neural network (DNN) into a SNN using weight and threshold balance. However, this method usually causes the performance of the SNN to be lower than that of the corresponding DNN, and also requires a lot of time for inference that significantly increases the energy cost. To overcome this, a hybrid learning algorithm for mapless navigation of mobile robots was proposed by Tang *et al.* [26], in which DRL is used to train SNN. They used the rate-coded inputs to learn the optimal policy in static environments. However, the optimality of the policy is highly dependent on the coding accuracy of a single spike neuron with limited representation capabilities, making the algorithm unsuitable for complex tasks.

Another aspect worth considering is that with the increase of optional power level and the number of D2DUs, the environment state and action space increase rapidly, which not only affects the convergence of the algorithm, but also makes many D2DUs focus on local optimal value. For this purpose, MADDPG [27] learns distributed policy in continuous action spaces, and COMA [28] utilizes a counterfactual baseline to address the credit assignment problem. However, the question of how to make the network maintain high performance in the case of multiple D2DUs still remains open.

In this paper, we propose to train the SNN combine with the MARL for dynamic resource allocation in D2D networks. This co-learning enabled synergistic information exchange between the SNN and MARL, allowing them to overcome each other's limitations. For this purpose, firstly, the spatio-temporal backpropagation (STBP) [29] has been applied to accelerate SNN training combining both the spatial domain (SD) and temporal domain (TD) in the training phase. Then, mean field

multi-agent reinforcement learning (MFRL) has been used to solve the problem of D2D network performance degradation for the first time. Thirdly, we combine the spiking mean field multi-agent reinforcement learning (SMFRL) with different DRL algorithms such as actor-critic (AC) [30] and proximal policy optimization (PPO) [31] to choose optimal channel and power in D2D networks. The rest of this article is organized as follows. In Section II, we introduce the system D2D model. In Section III, we describe the proposed algorithm in detail. Simulation results are provided in Section IV, followed by our conclusions in Section V.

II. SYSTEM MODEL

A. SYSTEM COMPONENTS

In this paper, we considered a decentralised network model [32] that a set of cellular users (CUs) $i \in \mathcal{I} = (1, 2, ..., I)$ and D2D users (D2DUs) $j \in \mathcal{J} = (1, 2, ..., J)$ randomly placed within the coverage area of the BS, as illustrated in Fig. 1. In this model, CUs and D2DUs use the same radio spectrum, which means that D2DUs must reuse the resources allocated to the CUs in order to use spectrum more efficiently. The decentralized setting means that each D2D user must make decisions without knowing the decisions of other D2DUs, which will cause some D2DUs to access one channel at the same time, called 'collision'. In order to simplify the simulation, we assume that in this collision process, all conflicting users do not have the capacity to transmit signal.

Each CU is allotted a single up-link channel at each time slot t, and each D2D user consisting of a transmitter (D2D-Tx) and receiver (D2D-Rx). CU and D2DUs can only reuse a single channel at any time. At time slot t, the transmit power of the ith CU is p_t^i and the transmit power of jth D2D-Tx is $p_t^{j,l}$ which has L levels denoted as

$$p_t^{j,l} = \frac{p_{max}}{L} l \tag{1}$$

where p_{max} is the maximum transmit power of D2DUs. The channel gain $g_t^{i,B}$ between CU i and the BS is defined as $g_t^{i,B} = \left|h_t^{i,B}\right|^2 \beta^{i,B}$, where $h_t^{i,B}$ is the short-scale channel coefficient of the channel between CU i and the BS, and $\beta^{i,B}$ represents the large-scale fading components including path-loss and log normal shadowing. Similarly, the channel gains over the jth D2D pair, from D2D pair j to the BS, and from CU i to D2D pair j are defined as g_t^j, $g_t^{j,B}$ and $g_t^{i,j}$, respectively. Therefor, the signal to interference plus noise ratio (SINR) of the ith CU from the BS can be defined as

$$\xi_t^i = \begin{cases} \frac{p^{i_t} g_t^{i,B}}{\sigma_t^2} & \text{if collision or } i\text{th channel not selected} \\ \frac{p^{i_t} g_t^{i,B}}{p_t^{j,l} g_t^{j,B} + \sigma_t^2} & \text{otherwise,} \end{cases} \tag{2}$$

where σ_t^2 is the additive white Gaussian noise power. And the SINR of the jth D2D pair is defined as

$$\xi_t^j = \begin{cases} 0 & \text{if collision occurs} \\ \frac{p_t^{j,l} g_t^j}{p_t^i g_t^{i,j} + \sigma_t^2} & \text{otherwise.} \end{cases} \tag{3}$$

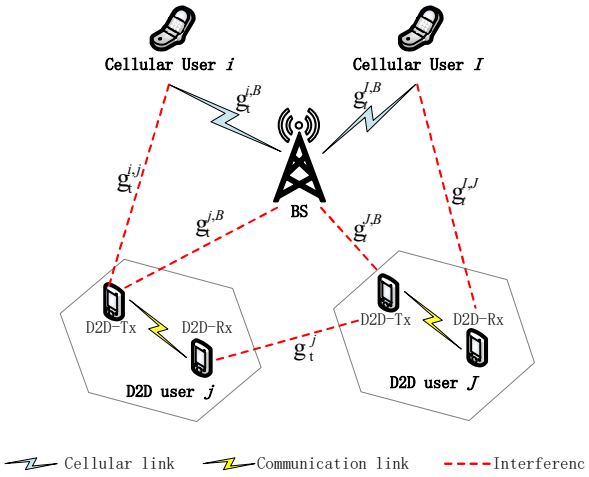

Fig. 1. D2D network model

TABLE I
SUMMARY OF MODEL NOTATION

Notation	Description
I	Number of CUs (Channels)
J	Number of D2DUs
L	Number of power levels
g_t^j	Channel gain of the jth D2D user
$g_t^{i,B}$	Channel gain from the ith CU to BS
$g_t^{j,B}$	Channel gain from the jth D2D user to BS
$g_t^{i,j}$	Channel gain from the ith CU to the jth D2D user
p^i	Transmit power of the i CU
$p_t^{j,l}$	The lth power level of the jth D2D user
p_{max}	Maximum transmit power of D2DUs
ξ_t^i	SINR of the ith CU
ξ_t^j	SINR of the jth D2D user
ξ_{min}^i	Minimum SINR of the ith CU
ξ_{min}^j	Minimum SINR of the jth D2D user
σ_t^2	Additive white Gaussian noise power
W	Channel bandwidth

B. PROBLEM FORMULATION

The channel selection and power control problem has the objective of maximizing the network throughput with minimal interference to the CUs and D2DUs to keep their SINR above certain threshold, which can be solved through each D2D-Tx selecting the optimal channel i and transmit power $p_t^{j,l}$. Consequently, the optimization problem can be formulated as

$$\max_{i \in \mathcal{I}, p_t^{j,l}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} W \left[log_2 \left(1 + \xi_t^i\right) + \mathrm{I}(i,j) log_2 \left(1 + \xi_t^j\right) \right] \tag{4}$$

$$\begin{aligned} s.t. \quad & \xi_i^t \geq \xi_{min}^i, \ \xi_j^t \geq \xi_{min}^j, \ \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \\ & \sum_i f(i,j) \leq 1, \ \sum_j f(i,j) \leq 1, \ \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \end{aligned} \tag{5}$$

where W is the channel bandwidth, ξ_{min}^i and ξ_{min}^j are the minimum SINR requirements for CU i and D2D j, respectively. $f(i,j)$ is an indicator function that equals 1 if D2DU

j reuses the ith channel and 0 otherwise. Constraints ensure that SINR minimums are kept and no channel will be used by multiple D2DUs as well as no D2DU selects multiple channels at the time slot t. The notation of our model is summarized in Table I.

III. METHODOLOGY

In this section, spiking mean field multi-agent reinforcement learning for D2D network will be illustrated in several parts. First, we describe the multi-agent environment of D2D network. Then, we illustrate how SNN is combined with DRL. In the third part, we describe spatio-temporal backpropagation method to accelerate the training of SNN. The final part describes the mean field approximation which can improve the performance under multi-agent D2D environment.

A. Multi-agent Environment

In the multi-agent D2D environment, agents are represented by the D2DUs. At that time step t, the state of the D2D network $\boldsymbol{s}_t$ is represented by the joint SINR of all CUs i.e. $\boldsymbol{s}_t = [\xi_t^1, \xi_t^2, ..., \xi_t^I]$. In the D2D environment, since the channel transmits status information to each D2D user, the status is observable for each user. At time slot t, the jth D2D user takes an action $\boldsymbol{a}_t^j$ which consists of two separate decisions: the transmit channel i and power $p_t^{j,l}$, i.e. $\boldsymbol{a}_t^j = [i, p_t^{j,l}]$. Since the number of channels is I and the number of transmit power levels is L, there are $I \times L$ actions in the D2D action space. Finally, the reward of jth D2D user are computed as

$$r_t^j = \begin{cases} -\varepsilon, & \text{if (5) are violated} \\ \sum_{i \in \mathcal{I}} log_2\left(1+\xi_t^i\right) + \mathrm{I}(i,j) log_2\left(1+\xi_t^j\right), & \text{otherwise} \end{cases} \tag{6}$$

where the punishment ε was selected over 0 to entice convergence rapidly [24]. This reward function falls in the mixed setting as the majority of the reward is identical for each agent with each agent receiving additional reward based on their own SINR.

B. The Combination of SNN and DRL

As shown in Fig. 2, the combination algorithm of SNN and DRL includes two parts: SNN based actor network and DRL based critic network, illustrated in [26]. At time slot t, the actor network generated an action $\boldsymbol{a}_t^j$ for jth D2D user according to the state $\boldsymbol{s}_t$. After that, the critic network predicted the action-value, which in turn optimized the actor network. The structure of the critic network is determined by the selected DRL algorithm.

The SNN based actor network consists of a K-layer SNN, a neural encoder, and a neural decoder [33]. The neural encoder and encoder can be seen as the first layer (input layer) and the last layer (output layer) of a multi-layered and fully-connected SNN which consists of neurons. The current-based leaky-integrate-and-fire (LIF) model is used to simulate spiking neurons in this model. There are two phases of LIF dynamics, dropping the j for notational simplicity, first, integrating the presynaptic spikes $\boldsymbol{o}_t$ into current $\boldsymbol{c}_t$. Second, integrating the current $\boldsymbol{c}_t$ into membrane voltage $\boldsymbol{v}_t$, as described lines 5 to 7 in Algorithm 1. d_c and d_v are the current and voltage decay factors, function $T(\mathbf{v}_t)$ is an indicator function that equals 1 if $\mathbf{v}_t \geq V_{th}$ and 0 otherwise. Thus, the neuron fires a spike if its membrane potential exceeds V_{th} which is set empirically.

The neural encoder encodes each dimension of the D2D state $\boldsymbol{s}_t$ as the activity of a neuron population $E^i, i \in 1, ..., I$. Neuron populations $\boldsymbol{E}$ has Gaussian receptive field $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ which are task-specific trainable parameters. For neuron population E^i , the receptive field is defined as (μ^i, σ^i). The encoder calculated the activity of the population in two phases: first, each state dimension ξ_t^i is transformed into the stimulation intensity $A_{E^i}^i$ of each neuron in the population E^i,

$$A_{E^i}^i = e^{-\frac{(\xi_t^i - \mu^i)^2}{2(\sigma^i)^2}} \tag{7}$$

Second, we use probabilistic encoding to generate the spikes of the neurons in $\boldsymbol{E} = [E^1, ..., E^I]$, where spikes $\boldsymbol{X}$ for all the neurons were generated at each time slot with the probabilities defined by $\boldsymbol{A_E} = [A_{E^1}^1, ..., A_{E^I}^I]$. The encoding process can be written as

$$\boldsymbol{X} = Encoder(\boldsymbol{s}_t, \boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{8}$$

where the generated spikes signal is sent to the second layer of the SNN.

The neural decoder comprised of neuron populations, which decodes the output activities into real-valued actions. The receptive field of the neurons of decoder is formed by its connection weights, which are learned as part of the training. Each neuron population D^u represented a dimension of the action a_t^u, $u \in 1, ..., U$. After every T time slots, the spikes of neurons in $\boldsymbol{D} = [D^1, ..., D^U]$ were summed up at layer K, denoted as

$$\boldsymbol{sc} = \sum_{t=1}^{T} \boldsymbol{o}_t^K \tag{9}$$

And fr^u is the uth output population firing rate of $\boldsymbol{fr}$, which is calculated by the uth dimension of $\boldsymbol{sc}$

$$fr^u = \frac{sc^u}{T} \tag{10}$$

After that, the uth dimension action a_t^u returned as the weighted sum of the fr^u (lines 13 in Algorithm 1).

C. Spatio-temporal Backpropagation for SNN Traning

In this work, we apply spatio-temporal backpropagation (STBP) algorithm [26] to train high-performance SNN based actor network for D2D system. The STBP combines the layer-by-layer spatial domain and the timing-dependent temporal domain. The goal of D2D user is to find the action policy $\boldsymbol{\pi}$, which dimension is equal to the dimension of action $\boldsymbol{a}$. Therefore, the gradient of loss relative to the policy $\nabla_\pi L$ shown in next part is used to train the parameters of neural decoder, SNN and neural encoder. The parameters of each output dimension u, $u \in 1, ..., U$ are updated independently:

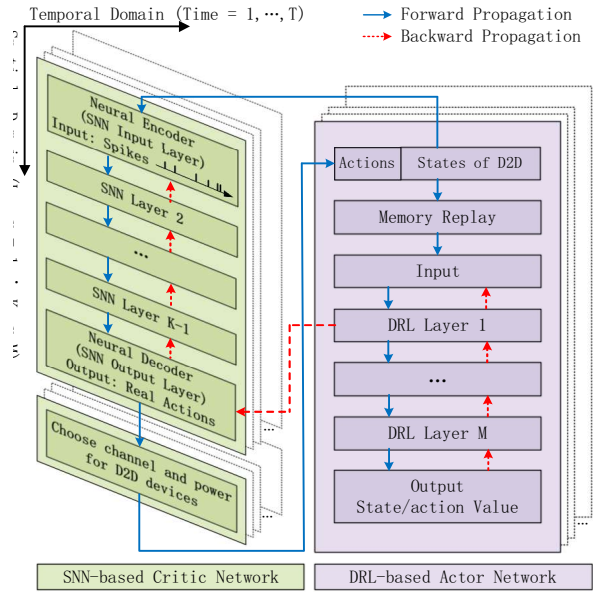

Fig. 2. The combination of SNN and DRL

$$\begin{aligned} \nabla_{\mathrm{fr}^u} L =& \nabla_{\pi^u} L \cdot \boldsymbol{W}_d^u \\ \nabla_{\boldsymbol{W}_d^u} L = \nabla_{\mathrm{fr}^u} L \cdot \mathrm{fr}^u,& \ \nabla_{\boldsymbol{b}_d^u} L = \nabla_{\mathrm{fr}^u} L \end{aligned} \tag{11}$$

where π^u is the uth dimension of policy $\boldsymbol{\pi}$. $\boldsymbol{W}_d^u$ and $\boldsymbol{b}_d^u$ are decoding weight vectors for the uth action dimension.

When the gradient of neuron decoder is updated, the gradient of each layer of SNN will continue to be updated from back to front. In order to solve the non-differentiable problem of SNN, the Gaussian cumulative distribution function [29] is chosen by us as a pseudo-gradient function to approximate the gradient of the spike:

$$z(v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v-V_{th})^2}{2\sigma}}, \tag{12}$$

where σ determines the curve steepness. We can regard the $z(v)$ as an indicator function that equals 1 if $|v - V_{th}| < \varepsilon$ and 0 otherwise. ε is the threshold window for passing the gradient. In every T time slots, the gradient derivation of K-layer SNN includes two cases: $t < T$ and $t = T$.

Case 1: for $t < T$.

In the output population layer K, the gradient flow through the SNN can be shown as

$$\nabla_{\mathbf{sc}} L = \frac{1}{T} \cdot \nabla_{\mathbf{fr}} L, \quad \nabla_{\mathbf{o}_t^K} L = \nabla_{\mathbf{sc}} L \tag{13}$$

where the $\mathbf{o}_t^K$ is the presynaptic spikes, i.e., the population output of layer K. Then, for each layer from $k = K$ to 1:

$$\begin{aligned} \nabla_{\mathbf{v}_t^k} L &= z(\mathbf{v}_t^k) \cdot \nabla_{\mathbf{o}_t^k} L + d_v(1 - \mathbf{o}_t^k) \cdot \nabla_{\mathbf{v}_{t+1}^k} L \\ \nabla_{\mathbf{c}_t^k} L &= \nabla_{\mathbf{v}_{t+1}^k} L + d_c \nabla_{\mathbf{c}_{t+1}^k} L \\ \nabla_{\mathbf{o}_t^{k-1}} L &= \boldsymbol{W}^k \cdot \nabla_{\mathbf{c}_t^k} L \end{aligned} \tag{14}$$

where $\boldsymbol{W}^k$ denotes the weight matrices of SNN layer k.

Case 2: for $t = T$.

Calculate the gradient loss of the SNN parameters for each layer k by collecting the gradients backpropagated from all time slots. In the output population layer K:

$$\nabla_{\boldsymbol{W}^k} L = \sum_{t=1}^{T} \mathbf{o}_t^{k-1} \cdot \nabla_{\mathbf{c}_t^k} L, \quad \nabla_{\boldsymbol{b}^k} L = \sum_{t=1}^{T} \nabla_{\mathbf{c}_t^k} L \tag{15}$$

where $\boldsymbol{b}^k$ denotes the bias of SNN layer k.

After the SNN is updated, the neuron encoder will be updated by directly backpropagating the $A_{E^i}^i$ of each input population:

$$\begin{aligned} \nabla_{\mu^i} L &= \nabla_{A_{E^i}^i} L \cdot A_{E^i}^i \cdot \frac{\xi^i - \mu^i}{(\sigma^i)^2} \\ \nabla_{\sigma^i} L &= \nabla_{A_{E^i}^i} L \cdot A_{E^i}^i \cdot \frac{(\xi^i - \mu^i)^2}{(\sigma^i)^3} \end{aligned} \tag{16}$$

All the parameters of SNN based actor network will be updated after every T timesteps and the detailed backpropagation derivation can be found in [33].

D. Mean Field Approximation for D2D network

A D2D user needs the status and rewards of all other CUs and D2DUs in the D2D network to complete their own resource allocation, which greatly reduces the performance of D2D network. In order to solve this problem, mean field approximation (MFA) [34] has been used in this D2D model. The aim of MFA is replacing the effect of all other agents with an mean effect, which convert many agent interactions into two agent interactions. The Q-function of the jth D2D user is decomposed using only the pairwise local interactions:

$$Q^j(\boldsymbol{s}, \boldsymbol{a}^j) = \frac{1}{N} \sum_{k \in \mathcal{N}(j)} Q^j(\boldsymbol{s}, \boldsymbol{a}^j, \boldsymbol{a}^k) \tag{17}$$

where $\mathcal{N}(j)$ is the index set of the neighboring agents of the jth agent and $N = |\mathcal{N}(j)|$ is the number of neighbors.

We use the one-hot encoding to indicate one of the $I \times L$ possible D2D actions: $\boldsymbol{a}^j \triangleq [\boldsymbol{a}_1^j, ..., \boldsymbol{a}_{I\times L}^j]$. The mean action $\bar{\boldsymbol{a}}^j$ is calculated based on the neighborhood $\mathcal{N}(j)$ of D2D user j

$$\bar{\boldsymbol{a}}^j = \frac{1}{N} \sum_{k \in \mathcal{N}(j)} \boldsymbol{a}^k \tag{18}$$

where $\boldsymbol{a}^k$ is consisted of $\bar{\boldsymbol{a}}^j$ and a fluctuation δ as $\boldsymbol{a}^k = \bar{\boldsymbol{a}}^j + \delta$. Besides, $\bar{\boldsymbol{a}}^j \triangleq [\bar{\boldsymbol{a}}_1^j, ..., \bar{\boldsymbol{a}}_{I\times L}^j]$ can be regarded as the empirical distribution of the actions that taken by neighbors of jth D2D user. According to the above formula and Taylor's theorem, an approximate expression [34] of $Q^j(\boldsymbol{s}, \boldsymbol{a}^j)$ is proved

$$Q^j(\boldsymbol{s}, \boldsymbol{a}^j) \approx Q^j(\boldsymbol{s}, \boldsymbol{a}^j, \bar{\boldsymbol{a}}^j) \tag{19}$$

At time slot t, given an experience $\boldsymbol{e}_t^j = (\boldsymbol{s}_t, \boldsymbol{a}_t^k, r_t^j, \boldsymbol{s}_t')$, the mean field Q-function is updated by MFA in a recurrent way as

$$Q^j_{t+1}(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t) = (1-\alpha)Q^j_t(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t) + \alpha[r^j_t + \gamma v^j_t(\boldsymbol{s}'_t)] \quad (20)$$

where α denotes the learning rate and γ denotes the discount factor which represents the uncertainty of the sender about the future rewards. And v^j_t is the mean field value function for D2D user j defined as

$$v^j_t(\boldsymbol{s}'_t) = \sum_{\boldsymbol{a}^j_t} \boldsymbol{\pi}^j_t(\boldsymbol{a}^j_t | \boldsymbol{s}'_t, \overline{\boldsymbol{a}}^j_t) \mathbb{E}_{\overline{\boldsymbol{a}}^j_t}[Q^j_t(\boldsymbol{s}'_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t)] \quad (21)$$

where $\boldsymbol{\pi}^j_t$ denotes the action policy for D2D user j. In spiking mean field actor-critic (S-MFAC),the D2D user j is trained by minimizing the loss function:

$$\mathcal{L}(\phi^j) = (y^j - Q^j_t(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t))^2 \quad (22)$$

where ϕ^j represents the weights in SNN based actor network. $y^j_t = r^j_t + \gamma v^j_t(\boldsymbol{s}'_t)$ is the mean field value calculated with the weights ϕ^j_- for target network. The policy $\boldsymbol{\pi}^j_t$ parameterized by θ^j in DRL based critic network is trained by

$$\nabla_{\theta^j} \mathcal{L} \approx \nabla_{\theta^j} \log \boldsymbol{\pi}^j_t Q^j_t(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t)|_{\boldsymbol{a}^j_t = \boldsymbol{\pi}^j_t(\boldsymbol{s}_t)} \quad (23)$$

The proof of convergence can be found in [34] and the pseudocode of S-MFAC under the D2D environment is shown in Algorithm 1.

Algorithm 1: Spiking Mean Field Actor-Critic (S-MFAC) for D2D Network

Input: Randomly initialize $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{W}_d$, $\boldsymbol{b}_d$, Q_{ϕ^j}, $Q_{\phi^j_-}$, $\boldsymbol{\pi}_{\theta^j}$, $\boldsymbol{\pi}_{\theta^j_-}$, $\overline{\boldsymbol{a}}^j_{t=0}$, and $\boldsymbol{s}_{t=0}$ for $j \in 1, ..., J$.
Input: Initialize $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, N, spikes $\boldsymbol{X} = Encoder(\boldsymbol{s}_{t=0}, \boldsymbol{\mu}, \boldsymbol{\sigma})$.
for *t=0, 1...* **do**
  for *j=1,...,J* **do**
    Drop j for simplicity.
    for *k=1,...,K* **do**
      Update LIF neurons in layer k:
      $\boldsymbol{c}^k_t = d_c \cdot \boldsymbol{c}^k_{t-1} + \boldsymbol{W}^k \boldsymbol{o}^{k-1}_t + \boldsymbol{b}^k$
      $\boldsymbol{v}^k_t = d_v \cdot \boldsymbol{v}^k_{t-1} \cdot (1 - \boldsymbol{o}^k_{t-1}) + \boldsymbol{c}^k_t$
      $\boldsymbol{o}^k_t = T(\boldsymbol{v}^k_t)$
    end
    if *t % T==0* **then**
      Sum up the spikes of output: $\boldsymbol{sc} = \sum_{t=1}^{T} \boldsymbol{o}^K_t$
      for *i=1,...,U* **do**
        Compute the uth dimension of firing rate: $fr^u = \frac{sc^u}{T}$
        Compute the uth dimension of action: $a^u = \boldsymbol{W}^u_d \cdot fr^u + \boldsymbol{b}^u_d$
      end
      Get action $\boldsymbol{a}^j_t$ for D2D user j.
    end
  end
  Compute all mean actions $\overline{\boldsymbol{a}}_t = [\overline{\boldsymbol{a}}^1_t, ..., \overline{\boldsymbol{a}}^J_t]$.
  Take the joint action $\boldsymbol{a}^*_t = [\boldsymbol{a}^1_t, ..., \boldsymbol{a}^J_t]$.
  Compute the joint reward $\boldsymbol{r}^*_t = [\boldsymbol{r}^1_t, ..., \boldsymbol{r}^J_t]$.
  Get the next state $\boldsymbol{s}'_t = [\xi^1_t, \xi^2_t, ..., \xi^I_t]$.
  Store $\boldsymbol{ex} = (\boldsymbol{s}_t, \boldsymbol{a}^*_t, \boldsymbol{r}^*_t, \boldsymbol{s}'_t, \overline{\boldsymbol{a}}_t)$ in replay buffer $\mathcal{R}$.
  if $|\mathcal{R}| \geq N$ **then**
    for *j=1,...,N* **do**
      Sample M experience $\boldsymbol{ex}$ from $\mathcal{R}$.
      Set $y^j_t = r^j_t + \gamma \boldsymbol{v}^{MF}_{\phi^j_-}$
      Update the SNN based actor network by:
      $\mathcal{L}(\phi^j) = (y^j - Q_{\phi^j}(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t))^2$
      Update the DRL based critic network by:
      $\nabla_{\theta^j} \mathcal{L} \approx \nabla_{\theta^j} \log \boldsymbol{\pi}^j_t Q_{\phi^j}(\boldsymbol{s}_t, \boldsymbol{a}^j_t, \overline{\boldsymbol{a}}^j_t)|_{\boldsymbol{a}^j_t = \boldsymbol{\pi}^j_t(\boldsymbol{s}_t)}$
      Update the parameters of target networks:
      $\phi^j_- = \tau_\phi \phi^j + (1 - \tau_\phi)\phi^j_-$
      $\theta^j_- = \tau_\theta \theta^j + (1 - \tau_\theta)\theta^j_-$
    end
  end
end

IV. SIMULATION RESULTS

In this section, we conduct simulations to examine the proposed schemes. In our model, D2DUs are randomly located in a circular space with a radius of 500m and the maximum D2D transmit power p_{max} and noise power ^2_N are respectively set to be 30 dBm and -174 dBm, the transmit power of CUs is 25dBm. The SINR minimum for CUs ξ^i_{min} and D2DUs ξ^j_{min} is 3dB and the bandwidth W is 10MHz. Except where noted, the number of channels I and the number of D2DUs J are set to be 30 and 10,respectively. The actual effects of finite-precision digital processing is taken into account, which truncates the received SINR by a maximum value of 30 dB [35]. Each simulation result is the average of the results obtained from 20 randomly initialized independent experiments, including topology and channel gains.

As for the parameters of the SNN based actor network and DRL based critic network are similar to [33] and [34], except for the following. The hidden layer of critic network is 32 and the learning rates α, τ_ϕ, τ_θ are $1e{-}4$, $1e{-}3$, $1e{-}3$, respectively. The punishment ε of reward r is 0.2. The number of neighbors N is 5 and the capacity M of replay buffer $\mathcal{R}$ is 10.

A. Simulation under small action space

In this section, firstly, in terms of the average reward, we compare the actor-critic (AC), proximal policy optimization (PPO), spiking AC (S-AC) and spiking PPO (S-PPO) with spiking mean field AC (S-MFAC) and spiking mean field proximal policy optimization (S-MFPPO) under the same environment, where $I = 30$, $J = 10$, and $L = 30$. It is noteworthy that AC, PPO, S-AC, and S-PPO do not contain any multi-agent principles like inter-agent communication or credit assignment. D2DUs learn solely based on their own observations of the environment. In order to improve readability, each plot is smoothed. As can be found in Fig. 3(a),

S-PPO and S-AC get higher average rewards, reaching 5.19 and 4.39 respectively at time slot 800, while PPO and AC are only 2.94 and 2.78 respectively. After applying mean field approximation, S-MFPPO and S-MFAC not only have higher average reward, but also nearly converge at time slot 450, while S-PPO and PPO close to convergence at time slot 850 and 1150 respectively. Secondly, we verify the performance of S-MFPPO and S-MFAC by detecting three indicators of D2D network [36], namely, colligation probability, access rate and network throughput. As shown in Fig. 3(b)-(d), S-MFPPO and S-MFAC also got better performance.

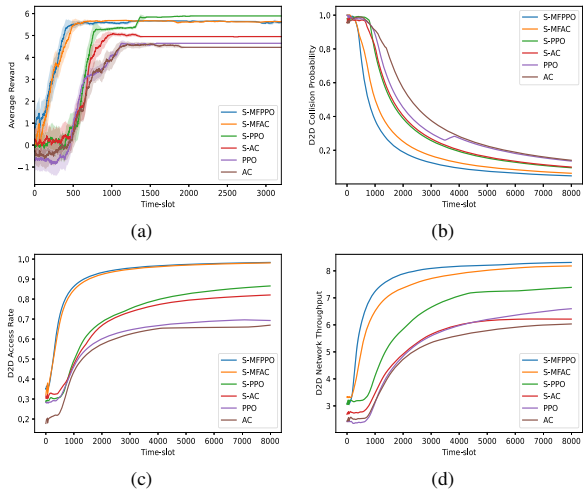

Fig. 3. Simulation results under 30 CUs, 10 D2DUs, and 30 power levels. (a)-(d) represents the performance of algorithms in terms of average reward, collision probability, access rate and D2D network throughput respectively.

B. Simulation under large action space

It can be found from Fig. 3(a) that there is no significant difference in convergence rate between S-PPO and PPO when the action space is small. In order to test the performance of proposed algorithm under large action space [37] and demonstrate the representation ability of neural encoder and decoder, we set $L = 900$ (other settings remain unchanged) for D2D network. As shown in Fig. 4(a), S-PPO and S-AC converge faster than PPO and AC, and also can obtain higher average reward. As for the colligation probability, access rate and network throughput, S-MFPPO is still perform better than others.

C. Simulation under more D2DUs

In this section, the performance of the proposed algorithm has been tested in the case of a large number of D2D pairs ($J = 30$) [38]. Fig. 5(a) shows that the D2D network can still maintain high performance in the case of 30 D2DUs by applying MFA. Due to the increase of the number of D2DUs, the collision probability of each algorithm is close to 1. In order to express clearly, the collision probability has been replaced by the specific number of collisions as shown in Fig. 5(b). It can be seen in Fig. 5(a)-(d) that the S-MFPPO can still

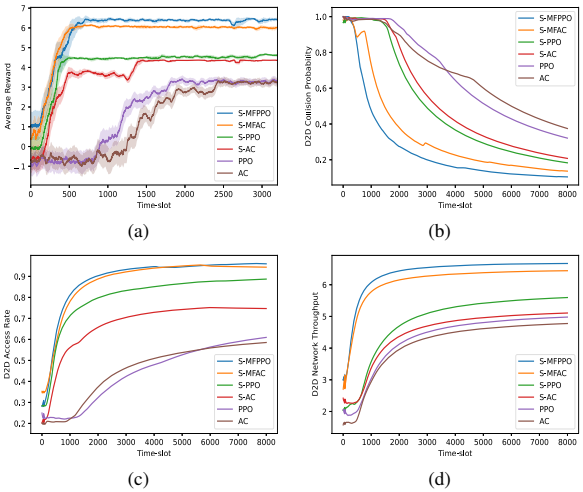

Fig. 4. Simulation results under 30 CUs, 10 D2DUs, and 900 power levels. (a)-(d) represents the performance of algorithms in terms of average reward, collision probability, access rate and D2D network throughput respectively.

maintain high efficiency in the face of more D2DUs, which proves that it is feasible to solve the D2D resource allocation problem by combining SNN and MFRL.

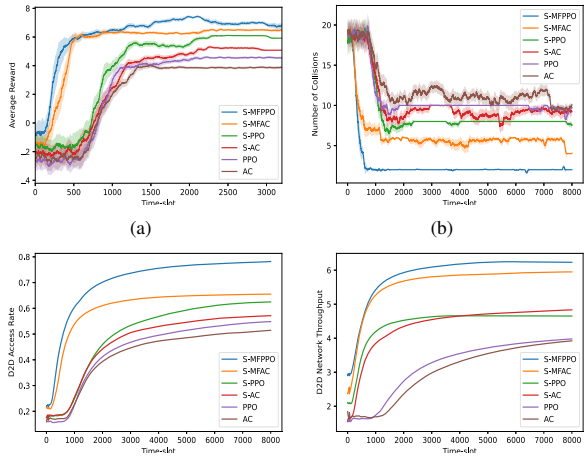

Fig. 5. Simulation results under 30 cellular users, 30 D2DUs, and 30 power levels. (a)-(d) represents the performance of algorithms in terms of average reward, number of collisions, access rate and D2D network throughput respectively.

V. CONCLUSIONS

In this paper, we combine the SNN with DRL to investigate the joint channel selection and power control problem of D2D network for the first time. However, as the number of D2D devices increases, calculating the reward of each device one by one will bring a large performance loss. To overcome this issue, we use the mean field multi-agent reinforcement learning to simplify the influence of multiple D2D devices on each other, and further improve the efficiency of the algorithm.

At the same time, we use spatial-temporal backpropagation to accelerate the training of SNN. Simulation results show that spiking mean field proximal policy optimization (S-MFPPO) achieves better performance than actor-critic (AC), proximal policy optimization (PPO), and spiking proximal policy optimization (S-PPO) on average reward, access rate, colligation probability, and throughput.

REFERENCES

[1] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to-device communications in cellular networks," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, 2014.

[2] L. Zhang, M. Xiao, G. Wu, M. Alam, Y.-C. Liang, and S. Li, "A survey of advanced techniques for spectrum sharing in 5g networks," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 44–51, 2017.

[3] G. D. Swetha and G. R. Murthy, "Selective overlay mode operation for d2d communication in dense 5g cellular networks," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 704–709.

[4] S. Xiao, X. Zhou, D. Feng, Y. Yuan-Wu, G. Y. Li, and W. Guo, "Energy-efficient mobile association in heterogeneous networks with device-to-device communications," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5260–5271, 2016.

[5] Y. Jiang, Q. Liu, F. Zheng, X. Gao, and X. You, "Energy-efficient joint resource allocation and power control for d2d communications," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6119–6127, 2016.

[6] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[7] S. Hakola, T. Chen, J. Lehtomki, and T. Koskela, "Device-to-device (d2d) communication in cellular network - performance analysis of optimum and practical communication mode selection," in *2010 IEEE Wireless Communication and Networking Conference*, 2010, pp. 1–6.

[8] A. Asheralieva and Y. Miyanaga, "Qos-oriented mode, spectrum, and power allocation for d2d communication underlaying lte-a network," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9787–9800, 2016.

[9] R. Yin, C. Zhong, G. Yu, Z. Zhang, K. K. Wong, and X. Chen, "Joint spectrum and power allocation for d2d communications underlaying cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2182–2195, 2016.

[10] D. Feng, G. Yu, C. Xiong, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Mode switching for energy-efficient device-to-device communications in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 6993–7003, 2015.

[11] H. S. Wang and N. Moayeri, "Finite-state markov channel-a useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, 1995.

[12] Y. A. Al-Gumaei, N. Aslam, and A. M. Al-Samman, "Non-cooperative power control game in d2d underlying networks with variant system conditions," *Electronics*, vol. 8, no. 10, 2019.

[13] L. Xiao, Y. Li, C. Dai, H. Dai, and H. V. Poor, "Reinforcement learning-based NOMA power allocation in the presence of smart jamming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3377–3389, 2018.

[14] Y. Yuan, T. Yang, H. Feng, and B. Hu, "An iterative matching-stackelberg game model for channel-power allocation in d2d underlaid cellular networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7456–7471, 2018.

[15] Y. Wang, M. Chen, N. Huang, Z. Yang, and Y. Pan, "Joint power and channel allocation for d2d underlaying cellular networks with rician fading," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2615–2618, 2018.

[16] A. Abrardo and M. Moretti, "Distributed power allocation for d2d communications underlaying/overlaying ofdma cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1466–1479, 2017.

[17] J. Lyu, Y. H. Chew, and W.-C. Wong, "A stackelberg game model for overlay d2d transmission with heterogeneous rate requirements," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8461–8475, 2016.

[18] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[19] L. Li, Y. Xu, J. Yin, W. Liang, X. Li, W. Chen, and Z. Han, "Deep reinforcement learning approaches for content caching in cache-enabled d2d networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 544–557, 2020.

[20] Y. Yan, B. Zhang, C. Li, and C. Su, "Cooperative caching and fetching in d2d communications - a fully decentralized multi-agent reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 095–16 109, 2020.

[21] B. Huang, X. Liu, and S. Wang, et al., "Multi-agent reinforcement learning for cost-aware collaborative task execution in energy-harvesting d2d networks," *Computer Networks*, vol. 195, no. 6, pp. 710–722, 2021.

[22] W. MAASS, "Networks of spiking neurons : The third generation of neural network models," *Neural networks*, 1997.

[23] M. Al-Yasari and N. Al-Jamali, "Modified training algorithm for spiking neural network and its application in wireless sensor network," *IARJSET*, vol. 5, pp. 33–42, 10 2018.

[24] V. Mnih, K. Kavukcuoglu, and D. Silver, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[25] D. Patel, H. Hazan, D. J. Saunders, H. T. Siegelmann, and R. Kozma, "Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to atari breakout game," *Neural Networks*, vol. 120, pp. 108–115, 2019.

[26] G. Tang, N. Kumar, and K. P. Michmizos, "Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware," *CoRR*, vol. abs/2003.01157, 2020.

[27] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *CoRR*, vol. abs/1706.02275, 2017.

[28] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[29] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *CoRR*, vol. abs/1706.02609, 2017.

[30] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, vol. 12, 2000.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[32] J. Mass, C. Chang, and S. N. Srirama, "Wiseware: A device-to-device-based business process management system for industrial internet of things," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016, pp. 269–275.

[33] G. Tang, N. Kumar, R. Yoo, and K. P. Michmizos, "Deep reinforcement learning with population-coded spiking neural network for continuous control," *CoRR*, vol. abs/2010.09635, 2020.

[34] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 5571–5580.

[35] J. Tan, Y.-C. Liang, L. Zhang, and G. Feng, "Deep reinforcement learning for joint channel selection and power control in d2d networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1363–1378, 2021.

[36] Z. Li and C. Guo, "Multi-agent deep reinforcement learning based spectrum allocation for d2d underlay communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1828–1840, 2020.

[37] Y. Lu, H. Lu, L. Cao, F. Wu, and D. Zhu, "Learning deterministic policy with target for power control in wireless networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[38] B. Chen, J. Zheng, and Y. Zhang, "A time division scheduling resource allocation algorithm for d2d communication in cellular networks," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 5422–5428.