Lateral Interactions Spiking Actor Network for Reinforcement Learning

Xiangyu Chen, Rong Xiao, Qirui Yang and Jiancheng Lv

October 23, 2023

Lateral Interactions Spiking Actor Network for Reinforcement Learning

Xiangyu Chen[1], Rong Xiao[1,2(✉)], Qirui Yang[3], and Jiancheng Lv[1,2]

[1] College of Computer Science, Sichuan University, Chengdu, China
[2] Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, Chengdu, China
[3] Arizona State University
chenxiangyu@stu.scu.edu.cn
{rxiao, lvjiancheng}@scu.edu.cn
qyang30@asu.edu

Abstract. Spiking neural network (SNN) has been shown to be a biologically plausible and energy efficient alternative to Deep Neural Network in Reinforcement Learning (RL). In prevailing SNN models for RL, fully-connected architectures with inter-layer connections are commonly employed. However, the incorporation of intra-layer connections is neglected, which impedes the feature representation and information processing capacities of SNNs in the context of reinforcement learning. To address these limitations, we propose Lateral Interactions Spiking Actor Network (LISAN) to improve decision-making in reinforcement learning tasks with high performance. LISAN integrates lateral interactions between neighboring neurons into the spiking neuron membrane potential equation. Moreover, we incorporate soft reset mechanism to enhance model's functionality recognizing the significance of residual potentials in preserving valuable information within biological neurons. To verify the effectiveness of our proposed framework, LISAN is evaluated using four continuous control tasks from OpenAI gym as well as different encoding methods. The results show that LISAN substantially improves the performance compared to state-of-the-art models. We hope that our work will contribute to a deeper understanding of the mechanisms involved in information capturing and processing in the brain.

Keywords: Reinforcement Learning · Spiking Neural Networks · Lateral Interactions

1 Introduction

Deep learning has a significant impact on machine learning, particularly in reinforcement learning (RL), leading to the development of Deep Reinforcement Learning (DRL) [1]. Recent advancements in DRL have pushed its performance beyond human-level capabilities across various reinforcement learning tasks [11,8,16]. However, the resource-intensive nature of DRL combined with Deep Neural Networks (DNNs) presents challenges in applications such as mobile robots which require power-efficient and low-latency processing systems[12,9]. As a result, there is a growing need to explore energy-efficient and low-latency alternative networks for DRL.

Compared to DNNs, Spiking Neural Networks (SNNs) show great potential in simulating brain-inspired topology and functions due to the complex dynamics. By integrating spiking neurons with biologically plausible plasticity principles, complex cognitive functions can be generated. The biological brain achieves efficient computation through cell assembly which prioritizes spatial-temporal coding for memory over decision-making readout [7]. In recent works, there has been a growing interest in integrating SNN into reinforcement learning algorithms [4,19,5]. While these approaches frequently depend on reward-modulated local plasticity rules which have shown success in simple control tasks, they often face challenges when they are applied to complex robotic control tasks due to their limited optimization capabilities. To address this limitation, several approaches have emerged that integrate SNNs with DRL optimization. One notable approach involves a hybrid learning framework proposed by [17], which introduces a population coded spiking actor network (PopSAN) trained alongside a deep critic network using DRL. This approach has shown impressive performance and energy efficiency in continuous control tasks. Another approach [20] presents a multi-scale dynamic coding improved spiking actor network (MDC-SAN) for reinforcement learning, aiming to achieve effective decision-making. It combines population coding at the network scale with dynamic neurons coding and incorporates 2nd-order neuronal dynamics to enable a powerful spatial-temporal state representation.

Although SNNs have shown promising results in the field of RL, there are still significant avenues for exploration regarding the neuronal behavior and dynamical equations of SNNs. In recent researches, Residual Membrane Potential (RMP) spiking neurons based on soft reset [6] have been proposed to preserve the high dynamics of biological neurons. These RMP neurons have demonstrated near lossless conversion from Artificial Neural Networks (ANN) to SNN, showcasing their effectiveness on challenging datasets. Existing SNN models for reinforcement learning predominantly utilize the leaky-integrate-and-fire (LIF) neuron model [10], which effectively extracts object features. While these models incorporate inter-layer connections for feedforward processing, they often overlook the importance of intra-layer connections which is a critical mechanism in biological neural systems for object recognition. Neuroscientists have observed that lateral interactions among retina neurons can enhance the perception of visual object edges [13]. In the fields of computational neuroscience and cognitive science, Dynamic Neuronal Field (DNF) models have gained popularity as recurrent neural networks with attractor dynamics. In DNF, a neuron excites nearby neurons while inhibiting others [14,15].This mechanism effectively enhances important input regions, suppresses areas with noise, and preserves valuable information within the neuron population [3]. Therefore, further exploration of dynamical improvements in SNNs in the context of RL remains valuable and worthwhile.

In this paper, we introduce a novel Lateral Interactions Spiking Actor Network(LISAN) to enhance the state representation capabilities of our model in solving complex RL tasks. Our model integrates lateral interactions between adjacent neurons into the equation which governs the spiking neuron's membrane potential. Additionally, considering the significance of residual potentials observed in biological neurons, we incorporate soft reset mechanism to enhance our model's overall functionality. This comprehensive approach enables a robust and effective information processing capability within the network. Our approach leverages a hybrid Actor-Critic network by combining the strengths of SNN and DNN. To address the coding challenges introduced by multi-dimensional state inputs in continuous RL tasks, we employ population coding techniques [17]. The gradient loss of each output action is computed by a deep critic network, which is integrated with the Twin Delayed Deep Deterministic policy gradient algorithm (TD3). Experiment results conducted on the continuous control tasks from the OpenAI gym benchmark demonstrate the superior performance of our method in terms of rewards gained compared to state-of-the-art models. Furthermore, we conduct evaluations of our model's performance using different encoding methods. The results consistently demonstrate the robust performance of LISAN.

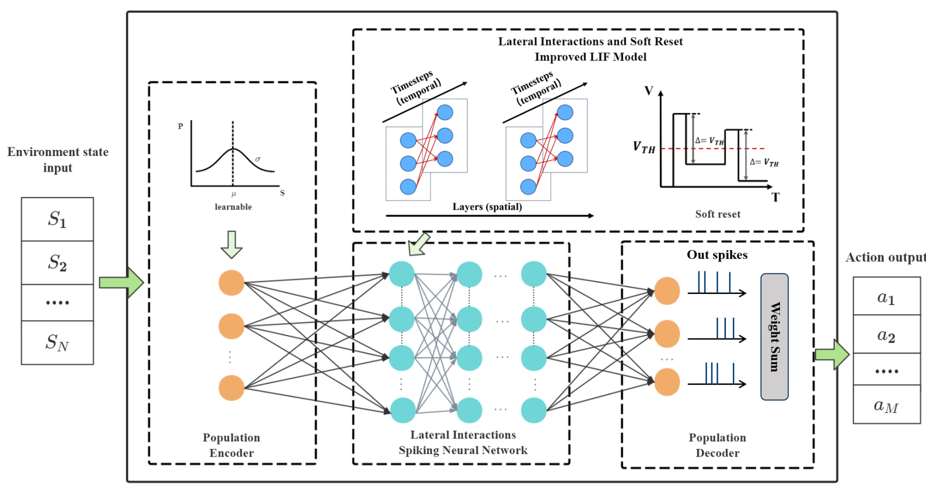


Fig. 1: The overall architecture of proposed LISAN

2 Method

In this section, we introduce the implementation details of our proposed model Lateral Interactions Spiking Actor Network (LISAN). The overall structure of LISAN is shown in Fig. 1.

2.1 Population Encoder

Population encoder refers to the encoding of information using the joint activity of multiple neurons within a population [17]. In our work, we utilize the population and deterministic coding method [20] to translate state information in RL task into spike trains that can be used as input to SNN.

For each dimension of the state S, we adopt a neuron population to perform encoding. Within each neuron population, the state information is initially transformed into stimulation strength through a Gaussian receptive field. Subsequently, the obtained stimulation strength represents the output of the pre-synaptic neuron, which is added to the membrane potential of the post-synaptic neuron. The sum is then compared to the threshold to determine whether a spike is fired. The overall process can be mathematically formulated as follows:

$$\begin{cases} P_{i,j} = EXP(-\frac{(S_i-\mu_{i,j})^2}{2\sigma_{i,j}^2}) \\ V_{i,j}(t) = V_{i,j}(t-1) + P_{i,j} \\ O_{i,j}(t) = \begin{cases} 1, & \text{if } V_{i,j}(t) > V_{th} \\ 0, & \text{otherwise} \end{cases} \end{cases} \tag{1}$$

where S_i is the i-th dimension of the state S, μ and σ denote the mean and standard deviation within the Gaussian receptive field, $P_{i,j}$ represents the stimulation strength of the j-th neuron within the neuron population which is responsible for encoding the i-th dimension of the state, $V_{i,j}(t)$ and $O_{i,j}(t)$ correspond to the membrane potential and spike activity of the neuron at time t, V_{th} is the firing threshold. $V_{i,j}(t)$ is reset to 0 if $O_{i,j}(t)$ is 1.

The μ and σ of the Gaussian receptive field are adjustable parameters, allowing the population encoder to progressively enhance its capability in representing the state information during the iterative training of the network.

2.2 Lateral Interactions Spiking Neuron Network

This section first provides an introduction to the conventional Leaky Integrate-and-Fire (LIF) neuron model, followed by a comprehensive definition and description of our proposed improved Lateral Interactions Spiking Neuron (LISN) framework.

LIF Neuron Model. The LIF model stands as a widely adopted mathematical framework for simulating physiological processes. The orignal LIF model can be described by:

$$\tau \frac{dV(t)}{dt} = -V(t) + RI(t) \tag{2}$$

where τ is the time constant, $I(t)$ denotes the input current accumulated from synapses and integrated into $V(t)$, which represents the membrane potential.

To capture the temporal and spatial relationship among neurons, we adopt a discretized time-step approach and partition the forward propagation within spiking neurons. This iterative model is formulated as follows:

$$I_i\left(t\right) = \alpha I_i\left(t-1\right) + \sum_j W_{ij} O_j(t-1) \tag{3}$$

$$V_i(t) = \beta V_i(t-1) + I_i(t) \tag{4}$$

$$O_i(t) = \Theta(V_i(t) - V_{th}) \tag{5}$$

where $V_i(t)$ and $I_i(t)$ represent the i-th neuron's current and voltage at time t, respectively. α, β are the current and voltage decay factor, and $\sum_j W_{ij} O_j(t-1)$ is the weighted sum of the incoming spikes from the previous layer j. $O_i(t)$ is the output spike and Θ is the Heaviside step function which is described by:

$$\Theta(x) = \begin{cases} 1, & \text{if } x > V_{th} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

When the membrane potential exceeds the firing threshold V_{th}, the neuron fires a spike and resets its membrane potential to zero. This mechanism is known as "Hard Reset".

Lateral Interactions and Soft Reset Improved Model

Soft Reset. Unlike hard reset which immediately reset the membrane potential to its initial value, soft reset allows a partial reset of the membrane potential after crossing a threshold. We content that the preservation of residual voltage which follows neuron firing confers enhanced efficacy upon SNN in complex spatiotemporal representation. Hence, to implement the soft reset mechanism in our model, we make the following modification to Eq.4:

$$V_i(t) = \beta(V_i(t-1) - O_i(t-1) * V_{th}) + I_i(t) \tag{7}$$

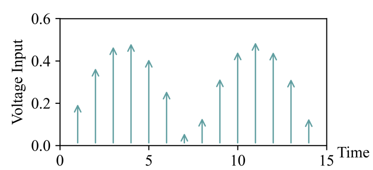
Lateral Interactions. Currently, a plethora of studies shows that incorporating internal connections within neural networks enhance computational performance in many fields [2,18]. Consequently,we propose merging the internal connections among neurons into the construction of SNN. We aim to investigate the potential enhancement of processing performance for RL tasks by introducing lateral interactions among neurons within the same layer in the spiking actor network.

Furthermore, we make the following enhancements to Eq.7 in our study:

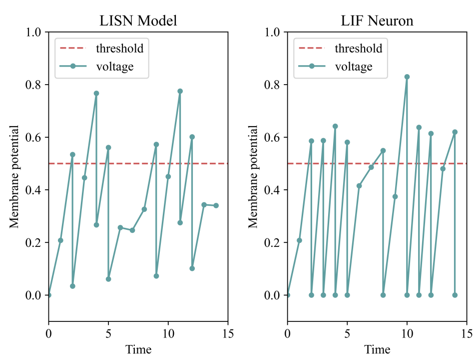
$$V_i(t) = \beta(V_i(t-1) - O_i(t-1) * V_{th}) + I_i(t) + (O_j(t-1) * W_N) \tag{8}$$

where $O_j(t-1)$ is the spiking output of the neighbors in the same layer at time $t-1$, W_N is a learnable matrix which characterizes the interconnections among neurons within the same layer to increase network flexibility.

In conclusion, we refer to the constructed model as Lateral Interactions Spiking Neuron (LISN). To observe the difference in information representation ability between LISN model and LIF model, We present sinusoidal wave signals to simulate the voltage inputs (weighted sum of pre-synaptic neuron outputs) to the neurons. We also simulate four neighboring neurons in the LISN model. The lateral interaction weight matrix W_N is initialized with random values from a normal distribution with mean 0 and standard deviation 0.05. Under identical inputs, the contrasting dynamics between LISN and traditional LIF neuron are depicted in Fig. 2. It is evident that under identical voltage inputs, the LISN and LIF neurons exhibit distinctive voltage dynamics, leading to disparate spike patterns. Specifically, the LISN neuron model produces a total of 6 output spikes, while the LIF model generates 9 spikes (see Fig. 2b). This observation highlights that the LISN neuron, with the incorporation of soft reset and interconnection, is less susceptible to continuous fluctuations in input voltages. Furthermore, for the sake of clarity in presentation, we maintain fixed parameters for the W_N. However, it is important to note that in actual experimental settings, W_N is subject to learning during the training iterations which enables LISN to robustly process intricate input information.



(a) Simulation of voltage input with sinusoidal wave variations



(b) Temporal Evolution of Membrane Potential. (Left) LISN Neuron (Right) LIF Neuron

Fig. 2: Contrasting dynamics between LISN and traditional LIF neurons with the same voltage inputs and threshold.

2.3 Surrogate Gradient

The non-differentiable nature of the firing function (Eq.8) poses challenges in training SNNs by backpropagation. To overcome the issue of gradient vanishing caused by the non-differentiable property of firing function, researchers have introduced the concept of surrogate gradient to facilitate gradient propagation in deep SNNs. In our work, we use the rectangular function equation to approximate the gradient of a spike.

$$d(x) = \begin{cases} 1, & \text{if } |x - V_{th}| < 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where d is the pseudo-gradient, x is membrane voltage, and V_{th} is the firing threshold.

2.4 Population Decoder

In the population decoding process, spikes generated at the output layer are accumulated within a predefined time window to calculate the average firing rate. Subsequently, the output action is obtained by applying a weighted sum to the computed average firing rate.

3 Experiment

In this section, we adopt LISAN as the model for the actor network with DNN as the critic networks. We evaluate LISAN in different environments and compare it with the mainstream models.

3.1 Benchmarking LISAN against mainstream models

Environment. We choose four classic MuJoCo continuous robot control tasks from the OpenAI gym (see Fig. 3) as our RL environment, the dimension information of each environment is shown in Table 1.

Benchmarking. We first compare LISAN to existing models DAN and PopSAN [17]. To investigate how soft reset and lateral interactions works on LIF neurons, We integrate them to PopSAN for comparison: PopSAN+SR(soft reset), PopSAN+LI(Lateral interactions). All models are trained using the TD3 algorithm combined with deep critic networks of the same structure. The hyperparameter configurations of these models are as follows:

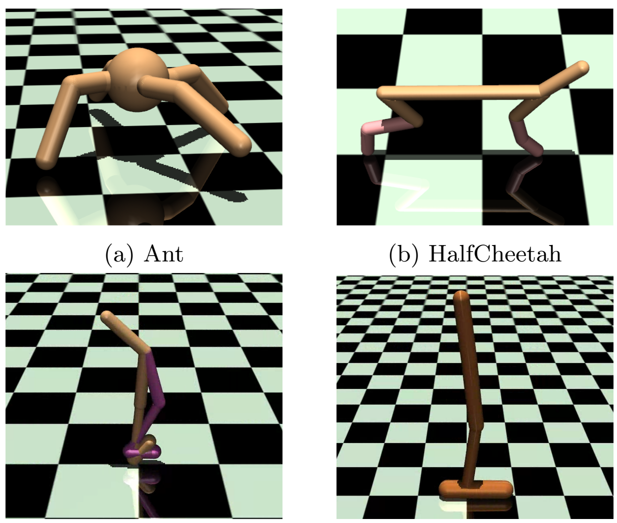
• *DAN*: Actor network (256, relu, 256, relu, tanh), learning rate $= 1e-3$; critic network (256, relu, 256, relu, linear), learning rate $= 1e-3$; mini-batch size $n = 100$; reward discount factor $\gamma = 0.99$; soft target update factor $\mu = 0.005$; policy delay factor $d = 2$; maximum size of replay buffer is $1M$.

• *PopSAN*: Actor network (Population Encoder, 256, LIF, 256, LIF, Population Decoder); input population size for single state dimension is 10; output population size for single action dimension is 10; firing threshold $V_{th} = 0.5$; current decay factor $\alpha = 0.5$; voltage decay factor $\beta = 0.5$; SNN time window $T = 5$; the remaining parameters are the same as DAN.

• *LISAN*: Actor network (Population Encoder, 256, LISN, 256, LISN, Population Decoder); the remaining parameters are the same as PopSAN.

• *PopSAN+SR*: PopSAN improved with Soft Reset.

• *PopSAN+LI*: PopSAN improved with Lateral Interactions.



(a) Ant (b) HalfCheetah (c) Walker2d (d) Hopper

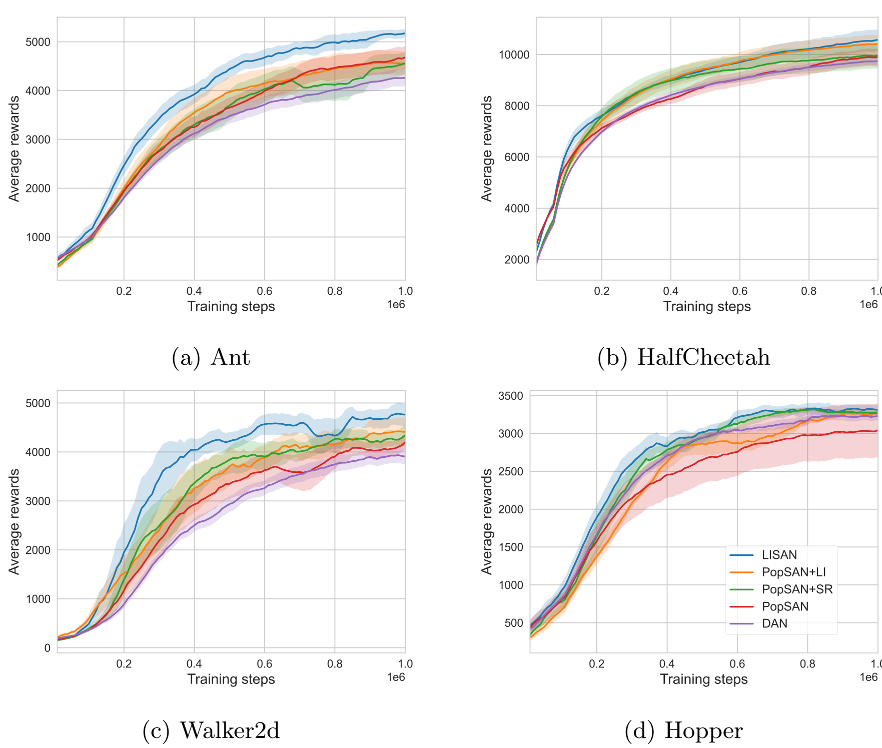
Fig. 3: The overview of four OpenAI gym tasks in the simulation environment: (a) Ant: make a quadruped crawling robot run as fast as possible; (b) HalfCheetah: make a biped robot walk as fast as possible; (c) Walker2d: make a biped robot walk quickly; (d) Hopper: make a 2D robot hop forward.

Table 1: Dimension information of four OpenAI gym environments

Environment	State Dimension	Action Dimension
Ant	111	8
HalfCheetah	17	6
Walker2d	17	6
Hopper	11	3

Training In our experiment, we perform ten independent trainings on each model, using the same 10 random seeds to ensure consistency. During our training procedure, we train the models in each environment for one million steps, and we test every 10K steps. During each test, we calculate the average reward across 10 episodes with each episode capped at 1K steps.

Result The experimental results are shown in Fig. 4 and Table 2. LISAN achieves the best performance in both simple and complex continuous control tasks. On the other had, DAN only performs well in simple environments, and fails to perform well with reinforcement learning tasks that involve high-dimensional environments and actions. Furthermore, we notice that the enhancement in model performance resulting from the soft reset mechanism is inconsistent. The residual voltage generated by soft reset proves advantageous to the model only in specific scenarios where it captures more dynamic information and leads to improved performance. And it is also obvious that LISAN exhibits a notable performance improvement even in the absence of soft reset.



(a) Ant

(b) HalfCheetah

(c) Walker2d

(d) Hopper

Fig. 4: Learning curves for different algorithms in the MuJoCo environment

Table 2: The maximum average return over 10 random seeds. LISAN achieves the best performance marked in bold.

Actor Network	Ant	HalfCheetah	Walker2d	Hopper
DAN	5038	10588	4745	3685
PopSAN	5736	10899	5605	3772
PopSAN+SR	5799	11295	5379	3590
PopSAN+LI	5811	11170	5816	3675
LISAN	**5965**	**11893**	**5901**	**3781**

3.2 Discussion of Different Input Codings for LISAN

We employ four different population coding methods [20] to encode the input state information, namely pure population coding (C_{pop}), population and Poisson coding ($C_{pop}+C_{poi}$), population and deterministic coding ($C_{pop}+C_{det}$), and population and uniform coding ($C_{pop}+C_{uni}$). We apply these coding methods to LISAN to test the performance of the Hopper task. Fig. 5 shows the four integrated population-based coding methods with LISAN. All of them achieve good performance in the Hopper environment. Notably, the population and deterministic coding achieves the most rewards thanks to its faster convergence speed while the pure population coding achieves the fewest rewards. These observations highlight the robust adaptability and processing capabilities of our LISAN model across diverse coding methods.

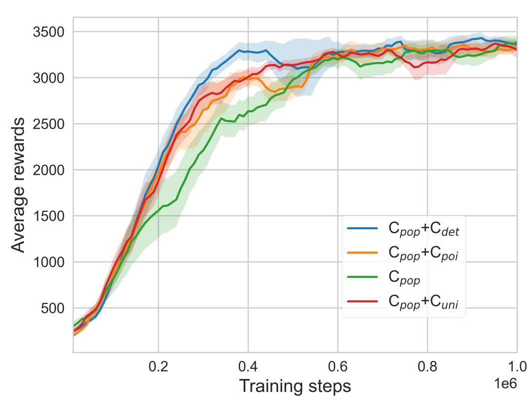


Fig. 5: Comprehensive comparison of the impact of various input coding methods on Hopper task

4 Conclusion

In this paper, we propose the Lateral Interactions Spiking Actor Network (LISAN) for deep reinforcement learning. LISAN utilizes the lateral interaction neuron model and incorporates soft reset mechanism for efficient training, which enables it to handle complex information while maintaining energy efficiency. Through extensive experiments conducted on four continuous control tasks from the OpenAI Gym, we demonstrate that LISAN outperforms the state-of-the-art deep neuron model as well as the same hybrid architecture SNN model. Additionally, we evaluate the performance of LISAN under different encoding methods, and show that LISAN achieves promising results across all of them. We hope that our work can serve as a fundamental building block for the emerging field of Spiking-RL and can be extended to a wide range of tasks in future research.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 62206188), the China Postdoctoral Science Foundation (Grant No. 2022M712237), Sichuan Province Innovative Talent Funding Project for Postdoctoral Fellows and the 111 Project under grant B21044.

References

1. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866 (2017)
2. Cheng, X., Hao, Y., Xu, J., Xu, B.: Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition. In: IJCAI. pp. 1519–1525 (2020)
3. Evanusa, M., Sandamirskaya, Y., et al.: Event-based attention and tracking on neuromorphic hardware. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
4. Florian, R.V.: Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. Neural computation **19**(6), 1468–1502 (2007)
5. Frémaux, N., Sprekeler, H., Gerstner, W.: Reinforcement learning using a continuous time actor-critic framework with spiking neurons. PLoS computational biology **9**(4), e1003024 (2013)
6. Han, B., Srinivasan, G., Roy, K.: Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13558–13567 (2020)
7. Harris, K.D., Csicsvari, J., Hirase, H., Dragoi, G., Buzsáki, G.: Organization of cell assemblies in the hippocampus. Nature **424**(6948), 552–556 (2003)
8. Hu, S., Zhu, F., Chang, X., Liang, X.: Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. arXiv preprint arXiv:2101.08001 (2021)
9. Lahijanian, M., Svorenova, M., Morye, A.A., Yeomans, B., Rao, D., Posner, I., Newman, P., Kress-Gazit, H., Kwiatkowska, M.: Resource-performance tradeoff analysis for mobile robots. IEEE Robotics and Automation Letters **3**(3), 1840–1847 (2018)

10. Memmesheimer, R.M., Rubin, R., Ölveczky, B.P., Sompolinsky, H.: Learning precisely timed spikes. Neuron **82**(4), 925–938 (2014)
11. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
12. Niroui, F., Zhang, K., Kashino, Z., Nejat, G.: Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. IEEE Robotics and Automation Letters **4**(2), 610–617 (2019)
13. Ratliff, F., Hartline, H.K., Lange, D.: The dynamics of lateral inhibition in the compound eye of limulus. i. Studies on Excitation and Inhibition in the Retina: A Collection of Papers from the Laboratories of H. Keffer Hartline p. 463 (1974)
14. Sandamirskaya, Y.: Dynamic neural fields as a step toward cognitive neuromorphic architectures. Frontiers in Neuroscience **7**, 276 (2014)
15. Schöner, G., Spencer, J.P.: Dynamic thinking: A primer on dynamic field theory. Oxford University Press (2016)
16. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484–489 (2016)
17. Tang, G., Kumar, N., Yoo, R., Michmizos, K.: Deep reinforcement learning with population-coded spiking neural network for continuous control. In: Conference on Robot Learning. pp. 2016–2029. PMLR (2021)
18. Wu, Z., Zhang, Z., Gao, H., Qin, J., Zhao, R., Zhao, G., Li, G.: Modeling learnable electrical synapse for high precision spatio-temporal recognition. Neural Networks **149**, 184–194 (2022)
19. Yuan, M., Wu, X., Yan, R., Tang, H.: Reinforcement learning in spiking neural networks with stochastic and deterministic synapses. Neural computation **31**(12), 2368–2389 (2019)
20. Zhang, D., Zhang, T., Jia, S., Xu, B.: Multi-sacle dynamic coding improved spiking actor network for reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 59–67 (2022)