2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937639

A Lightweight Spiking GAN Model for Memristor-centric Silicon Circuit with On-chip Reinforcement Adversarial Learning

Min Tian[1], Jing Lu[1,*], Haoran Gao[1], Haibing Wang[1], Jianyi Yu[1], Cong Shi[1]

[1]School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

Email: jinglu@cqu.edu.cn

***Abstract*—As a powerful generative model, Generative Adversarial Network (GAN) is widely studied to automatically generate high-quality new data to greatly enhances the capabilities of artificial intelligence (AI) technology. However, the unique training process of GAN comes at a very high computational complexity and high cost of memory accesses. In this work, a memristor-based spiking-GAN neuromorphic hardware system is proposed to address the challenges. Both the generator and discriminator of GAN are in the form of spiking neural network (SNN) to improve the computational performance, and the memristor synapse circuit with 1 memristor and 4 transistors (1M4T) is proposed as Computing in Memory (CIM) to avoid the cost of memory accesses. The reinforcement learning rule (i.e., reward-modulated spike-timing dependent plasticity, or R-STDP) is used to train both discriminator and generator networks, with a new backpropagation method for the reward/punishment signal. Tests on the MNIST and Fashion-MNIST datasets showed that the proposed GAN can efficiently generate data samples. The results demonstrate the great potential of this memristor-based spiking-GAN for high-speed energy-efficient data augmentations.**

Keywords—Spiking neuron, Generative adversarial network, Memristor, Reinforcement learning, Reward-modulated STDP, On-chip learning, Neuromorphic systems

I. Introduction

Generative adversarial networks (GANs) [1] are one of the most promising researched deep learning topics, and have been extensively used in various image processing applications [2,3]. While GAN comes at the cost of high computational complexity, conventional CPU and GPU platforms are not optimal for this intensive training process due to the well-known power wall [4] and memory wall [5] phenomena. Most importantly, these platforms cannot achieve real-time performance which limited the application of GANs. There are also some hardware accelerators for Deep Neural Network (DNN) already been fabricated such as DaDianNao [6], Eyeriss [7], which greatly accelerates the inference speed of DNN. However, these chips do not support on-chip learning. The large consumption of computation and storage resources of GANs expose great challenges to the on-chip real-time application on edge computing and embedded systems.

On the other side, spiking neural networks (SNNs), which mimic the cognitive mechanism of the human brain cortex by encoding and processing information via sparse spikes, are demonstrated to be highly suited for low-power hardware implementation [8]. Compared to conventional DNNs with intensive computations, SNNs significantly reduce the computation resources thanks to their event-driven sparsity. To further improve the energy efficiency, it is urgent to reduce storage consumptions. The emerging memristors have now attracted an increasing interest and become a promising candidate for hardware accelerator implementations of neural network [9]. Such devices compute exactly where the data are stored, thus avoiding the communication bottleneck. Therefore, combining SNNs with memristors is very promising for the realization of lightweight (i.e., low-cost) and high energy-efficient GANs.

This work proposes an approach to integrate memristors into the spiking-GAN hardware implementation, with on-chip learning capability. Both the generator and discriminator of GAN are a single layer SNN, whose synaptic weights are trained using the reinforcement learning rule, specifically, the reward-modulated spike-timing dependent plasticity (R-STDP) rule [10]. We propose a novel method of backpropagating the reward/punishment signals from the discriminator to the generator. A synapse circuit with 1 memristor and 4 transistors is designed to realize the R-STDP rule, and is integrated into the generator and discriminator of spiking-GAN as synapse.

The rest of this paper is organized as follows. Section II describes the background of the GAN with spiking form and memristors. Section III presents the algorithm of spiking-GAN. Section IV describes the memristor based hardware realization. Section V demonstrates the experimental results of data samples generated through memristor based spiking-GAN. Finally, Section VI concludes this paper.

II. Background

A. GAN and Spiking Neuron

GANs typically consist of two subnetworks, a Generator (G) and a Discriminator (D). The discriminator is a classifier which is trained to distinguish real data from the generated ones, and the target of generator is to produce data that can deceive the discriminator. Both networks compete with each other, and are trained simultaneously. The training process of GANs include three stages: 1) the discriminator is feeding to a real sample and is trained as real. 2) the discriminator is feeding to a false sample generated by the generator and is trained as false. 3) the discriminator is feeding to a false sample produced from the generator and remain unaltered, but return a “real” loss to train the generator. However, both G and D generally are modeled as deep neural networks, which consume lots of computing resources.

SNN is composed of biological spiking neurons, e.g., the commonly employed Leaky Integrate-and-Fire (LIF) neuron model, that communicate with each other using discrete spikes. A neuron has multiple synapses, and incoming spikes are integrated through their respective synapses onto the neuron membrane overtime. The membrane potential undergoes an exponentially decay when there is no input spike. If the membrane potential reaches a threshold, the neuron fires (i.e., generating an output spike) and resets to a resting potential. In order to reduce the computing resources, this work propose spiking implementation of a simple Generative Adversarial Network.

B. Memristor

Implementing learning by software on CPUs or GPUs inevitably incurs severe performance and energy bottlenecks, as these von Neumann computers adopt separated processing and memory components with bandwidth limitation. Using customized accelerating circuits based on standard CMOS technology alleviates this problem, but the logic and memory cells are still separated. Therefore, a new hardware paradigm that tightly coupled computing and memorizing in a single device is desired. Among these emerging devices, the memory resistor, or memristor for short, is the most popular one [11]. It is an electrical nanoscale device that can retain a state of internal conductance based on the history of applied voltage and current [12]. Its conductance ideally serves as the synaptic weight, and can be tuned by the applied voltage or current, realizing in-memory computing (learning plasticity). Therefore, to reduce the power of memory access, this work integrates memristors into the spiking-GAN to accelerate training efficiently and realize on-chip learning.

III. Algorithm

A. Spiking-GAN Model

As shown in Fig. 1, the spiking GAN model consists of the Generator and Discriminator networks, an Encoder and Decoder for pixel-to-spikes and spikes-to-pixel conversion, respectively. Both G and D subnetworks are a single layer spiking neural network. The G takes a random spike noise as input. The generator outputs a false spike image whose dimension is the same as the real input encoded spike images, and the false spike image is decoded to present the pixel image. The discriminator takes the encoded real spike image and false spike image (produced by the generator) as inputs. It has two output neurons (true '1' and false '-1'), which determine the classification result of the discriminator.

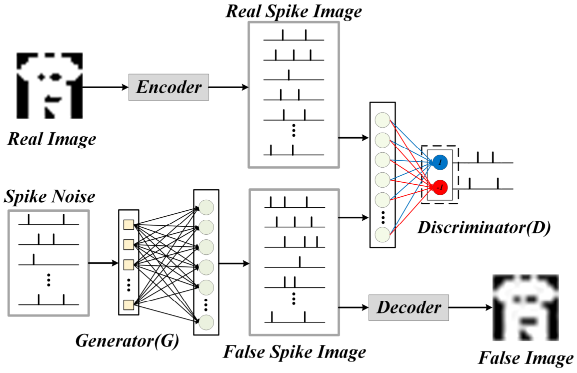


Fig. 1. The spiking-GAN architecture.

B. Reinforcement Learning rule for spiking-GAN

The spike-timing dependent plasticity (STDP) learning rule is a popular method for training spiking neurons, which adjusts the weight of one synapse depending on the time difference of pre and post synaptic spikes [13]. However, this learning rule suffers low recognition accuracy [14]. The supervised STDP variant R-STDP is an efficient reinforcement learning mechanism that can achieve high accuracy for single-layer SNNs [15]. Each neuron is preassigned an object category. If the neuron fires correctly (i.e., the category of the input pattern matches that of the neuron), the neuron is rewarded to perform a normal-STDP process on its synapses: when a presynaptic spike comes earlier (later) than the postsynaptic one, the synapse is strengthened (weakened). Otherwise, the neuron is punished and performs an anti-STDP process, which updates the synaptic weight in the direction opposite to the normal-STDP.

In this work, the spiking-GAN are trained using reinforcement learning rule. Both the weights of generator and discriminator are adjusted by R-STDP rule. Once the generator and discriminator neurons get their Reward/Punish signal and can train their weights through R-STDP learning rule.

If the neuron is rewarded, then

$$\Delta w = \begin{cases} a_r^+ e^{\frac{-\Delta t}{\tau_+}} & if\ \Delta t = t_{post} - t_{pre} \geq 0 \\ a_r^- e^{\frac{\Delta t}{\tau_-}} & if\ \Delta t = t_{post} - t_{pre} < 0 \end{cases} \quad (1)$$

Otherwise, if the neuron is punished

$$\Delta w = \begin{cases} a_p^+ e^{\frac{\Delta t}{\tau_-}} & if\ \Delta t = t_{post} - t_{pre} < 0 \\ a_p^- e^{\frac{-\Delta t}{\tau_+}} & if\ \Delta t = t_{post} - t_{pre} \geq 0 \end{cases} \quad (2)$$

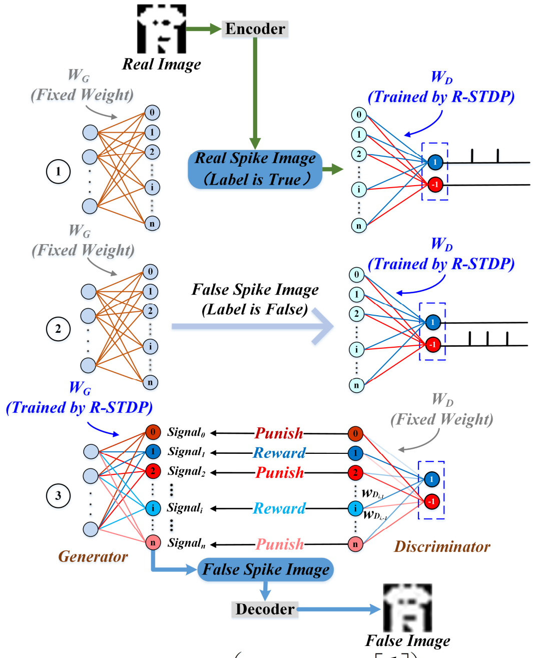
Among them, Δw is the amount of weight change for the synapse, t_{post} is postsynaptic spike time, t_{pre} is presynaptic spike time. a_r^+, a_r^-, a_p^+ and a_p^- scale the magnitude of weight change. Specifically, a_r^+, $a_p^+ > 0$, a_r^-, $a_p^- < 0$, and $a_r^- \ll a_r^+, a_p^+ \ll a_p^-$, $\tau_{+,} \tau_-$ is the time constant.

The training process of spiking-GAN consists of 3 stages (as shown in Fig.2). At the first stage, the discriminator is feeding to encoded real spike image, and the label is preassigned 'true'. The weights of the generator are fixed and only the discriminator is trained by R-STDP rule, that is, the weights of true neuron ('1') and false neuron ('-1) of discriminator are adjusted by normal-STDP and anti-STDP rule, respectively. During the second stage, the generator is feeding to spike noise and outputs spikes (namely, the false image spikes) which is then feeding to the discriminator, so the label is preassigned 'false'. The generator still remains unaltered, but the discriminator implements R-STDP rule according to its false label (the weights of false neuron '-1' and true neuron '1' of discriminator are adjusted by normal-STDP and anti-STDP rule, respectively). At the third stage, the weights of discriminator are fixed and only the generator is trained using R-STDP rule. It requires a Reward/Punish signal for the neurons of the generator to determine whether the neuron is rewarded or punished. We propose a new method of backpropagation of the Reward/Punish signal through the discriminator to the generator. The spike image produced by the generator is preferred to be judged by the discriminator as 'true' (the neuron '1' backpropagates Reward signal, and neuron '-1' backpropagates Punish signal). The Reward/Punish signal of the generator neurons (*signal$_i$*) are determined by the weight and Reward/Punish signal of the discriminator by the formula of (3).

$$signal_{\ i} = Sign\left(\left[W_{Di,1}\ W_{Di,-1}\right] \times \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) \quad (3)$$

If *signal$_i$* > 0, the *i*th neuron of the generator is rewarded.
If *signal$_i$* < 0, the *i*th neuron of the generator is punished.

The whole spiking-GAN training process repeats the 3 stages, in each turn, real spike image and spike noise are fed to the discriminator and generator, respectively. In the whole training process, both the discriminator and generator are trained by the R-STDP reinforcement learning rule, only need a Reward/Punish signal, thus avoiding complex gradient calculation to update the weights. This reinforcement learning rule is quite suitable for on-chip hardware realization.



$$Signal_i = Sign\left(\begin{bmatrix} W_{D_{i,1}} & W_{D_{i,-1}} \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right)$$

If $Signal_i$=1, the ith neuron is rewarded
If $Signal_i$=-1, the ith neuron is punished

Fig. 2. The training algorithm of spiking-GAN

IV. Memristor based Spiking-GAN Accelerator Circuit Design

A. Memristor based R-STDP Synapse Circuit Design

Memristor is a two-terminal device with polarity characteristic. It can be used as a plastic neural synapse by tuning its conductance with a pair of temporally varying voltages in dedicate shapes (e.g., exponential, triangular or rectangular) on its two terminals. The increase or decrease of memristor conductance is depending on the polarity of the voltage difference [16]. By converting the relative timing between the pre- and postsynaptic neuron spikes to differences in voltage amplitude or duration on the memristor, STDP mechanism can be implemented [17]. We implement a 1M4T (1 memristor and 4 transistors) synapse circuit to realize R-STDP learning rule [18]. As shown in Fig.3 (a), the transistors guide the memristor to carry out either normal-STDP or anti-STDP rule, according to the external Reward/Punish signal.

In this work, we select the widely used VTEAM memristor model [12]. Simulation have been carried out using Cadence Virtuoso as the simulator, and the memristor model was written in Verilog-A. The key parameters are set as: k_{on} = -1e-4, k_{off} = 1e-4, α_{on} = 3, α_{off} = 3, v_{on} = -0.01, v_{off} = 0.01, w_{on} = 0, w_{off} = 3. For more details about the device model and the parameter definitions, refer to [12]. We have designed a spike shape as shown in Fig. 3(a) to realize approximated R-STDP rule (Fig. 3(b)) when the spikes are applied to the two terminals of the 1M4T synapse circuit. In order to realize the reinforcement learning of spiking GAN algorithm and further minimize the complexity of memristor circuit design, we reduced a_r^- in (1) a_p^+ in (2) down to 0 (as $a_r^- \ll a_r^+, a_p^+ \ll a_p^-$), which means it only works when t_{post} - t_{pre} > 0 (Fig.3 (b)).

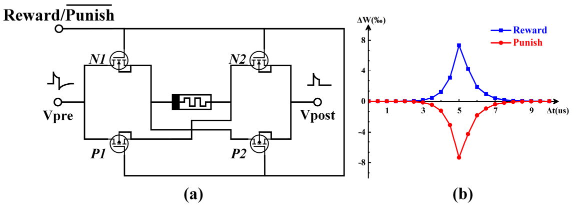


Fig. 3. (a) R-STDP circuit based on 1 memristor and 4 transistors. (b) approximated R-STDP curve.

B. Fully Hardware Implementation of Memristor based spiking-GAN

We employed two memristor-based single layer SNNs into a GAN architecture and demonstrate a system for generating images. The block diagram of the system is shown in Fig. 4. Both the generator and discriminator consist of single layer of neurons fully interconnected by 1M4T synapses in a crossbar fashion, in which the neuron is based on LIF neuron model. The input layer of the generator is applied spike train noise which is generated under gaussian distribution. These spike train noise generates current through the connected 1M4T synapses circuit and causes accumulation in the neurons of generator, once the membrane potential of the neurons exceed their threshold value, they would fire spikes. These neurons' spikes consist the generated false image spikes. The discriminator is a binary classification which aims to distinguish whether the spike image is real or false (produced by the generator).

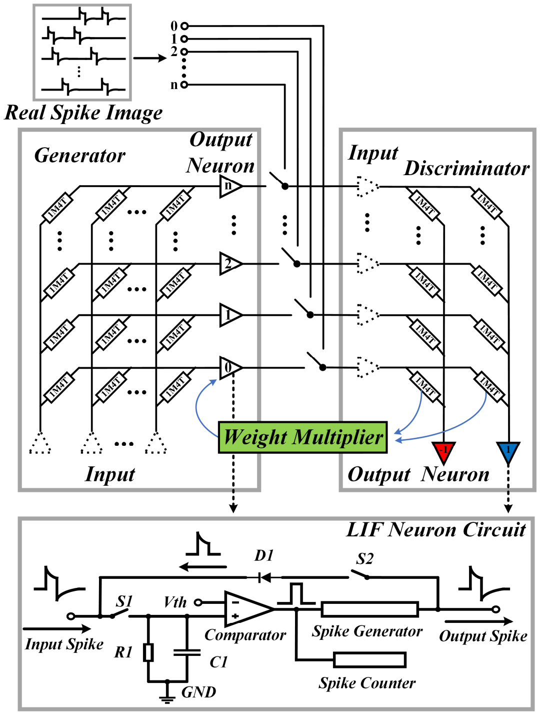


Fig. 4. The spiking-GAN Hardware topology implemented with memristor.

Now we investigate how the above proposed memristor based circuit can realize reinforcement learning for spiking GAN. When the discriminator is trained, the true and false output neurons would be rewarded or punished according to the input spike images, and the corresponding weights can be adjusted by R-STDP rule. For the training process of generator, the reward/punish signal is calculated from $signal_i$ as in section III. We implemented the formula by Verilog-A for behavior method, and the formula can also be realized in a vector-matrix form of memristor circuit (not shown here). The weights of generator can therefore be adjusted by R-STDP rule.

V. Full System Simulation Results

The entire system is built with Verilog-AMS for behavioral simulation. MNIST (a standard handwritten digital dataset) and Fashion-MNIST (the same size with MNIST, but it consists more complicated and challenging image patterns of fashion objects) datasets are used in this system for verification. The images are pre-processed to decrease the number and intensities of pixels. They were downsampled from 28 × 28 to 12 × 12 resolution and then binarized. Each pixel of the pre-processed image is randomly encoded into spike trains. The foreground pixel (i.e., the pixel with the value of 1) emits 2-4 spikes randomly within the time window of 0-40 us, and the background pixel (i.e., the pixel with the value of 0) remains silent. Therefore, the generator and discriminator are 12 × 144 and 144 × 2 single layer SNN, respectively. All the weights of 1M4T synapses (i.e., the memristor conductance) of generator and discriminator were randomly initialized. We trained our network on the preprocessed MNIST and Fashion-MNIST dataset for each individual image. The training process followed the three stages as illustrated in section III and repeated several epochs until the generated image looks like the real one. The generated spikes by the generator are feed to spike counters and decoded to image pixels for visualization. Fig. 5 exhibits the original preprocessed MINIST and Fashion-MNIST sample images and the generated images by the generator. The MINIST and Fashion-MINST samples are successfully produced by the trained generator. The generated samples look like the real images, it may be caused by the high learning rate during the training process. However, our main purpose is to validate the memristor realization of spiking GAN, not the perfect algorithm, and the results demonstrate the potential of the memristor-based spiking GAN.

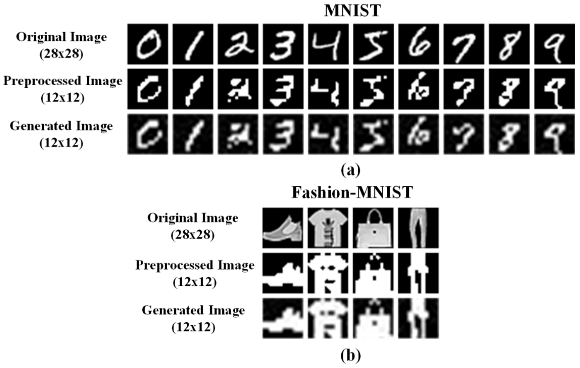


Fig. 5. Comparison between the training real images and the generated samples by the memristor based spiking GAN. (a) MNIST. (b) Fashion-MNIST

Prior works have used memristors for GAN implementation, such as LerGAN [19], ReGAN [20]. These works take advantage of the memristor's processing in memory technology and greatly improves the performance to accelerate GAN training. However, these GANs are based on traditional deep learning neural networks which evolves real-number complex computation and consumes much energy. Our proposed memristor based spiking GAN is the first spiking neural network implemented with memristor realization. It has several advantages. First, our GAN is based on spiking neural network, both the generator and discriminator are single layer SNN, which greatly reduce the network complexity and computation energy. Second, our spiking GAN model can be trained using R-STDP learning rule, this reinforcement learning only need a reward/punish signal for the discriminator and generator to train their weights, thus avoiding the complexity of gradient calculation. Third, our memristor-based spiking GAN can realize on-chip learning, combining both the spiking computational efficiency and memristor's computing in memory advantages, it demonstrates the great potential for high-speed energy-efficient applications.

VI. Conclusion

This paper proposes a lightweight spiking GAN model with memristor implementation. It significantly reduces the computational complexity and support on-chip R-STDP learning. A prototype single-layer spiking GAN is constructed with 1M4T synapses, and is simulated successfully on the MNIST and Fashion-MNIST dataset samples. This is the first time that the memristor completes hardware spiking GAN, and exhibits great potential for high-speed energy-efficient applications.

Acknowledgment

This work was funded in part by the National Key Research and Development Program of China (Grant No. 2019YFB2204303), in part by the Chongqing Natural Science Foundation (Postdoctoral Foundation) (Grant No. cstc2021jcyj-bsh0126), in part by the National Natural Science Foundation of China (Grant No. U20A20205), and in part by the Key Project of Chongqing Science and Technology Foundation (Grant No. cstc2019jcyj-zdxmX0017, cstc2021ycjh-bgzxm0031).

References

[1] I. Goodfellow, J. Pouget-Aradie, M. Mirza, et al., "Generative Adversarial Nets," *Advances in neural information processing systems (NIPS)*, 2014

[2] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[3] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv:1809.11096*, 2018.

[4] X. Guo, E. Ipek, and T. Soyata, "Resistive computation: avoiding the power wall with low-leakage, STT-MRAM based computing," *ACM SIGARCH computer architecture news*, vol. 38, no. 3, pp. 371-382, 2010.

[5] W. Wulf, and S. A. McKee. "Hitting the memory wall: Implications of the obvious." *ACM SIGARCH computer architecture news*, vol. 23, no. 1, pp. 20-24, 1995.

[6] Y. Chen, T. Luo, S. Liu, et al. "Dadiannao: A machine-learning supercomputer." *47th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE*, 2014.

[7] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, 2017.

[8] M. Bouvier, A. Valentian, T. Mesquida, et al. "Spiking Neural Networks Hardware Implementations and Challenges: A Survey," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no.

2, pp. 1-35, 2019.

[9] P. Yao, H. Wu, B. Gao, et al. "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641-646, 2020.

[10] N. Frémaux and W. Gerstner, "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules," *Frontiers in Neural Circuits*, vol. 9, no. 85, 2016.

[11] S. Choi, J. Yang, and G. Wang. "Emerging Memristive Artificial Synapses and Neurons for Energy-Efficient Neuromorphic Computing." *Advanced Materials,* vol. 32, no. 51, pp.2004659, 2020.

[12] S. Kvatinsky, M. Ramadan, E. G. Friedman, et al. "VTEAM: A General Model for Voltage-Controlled Memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786-790, 2015.

[13] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919-926, 2000.

[14] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, p. 99, 2015.

[15] M. Mozafari, S. R. Kheradpisheh, T. Masquelier, et al. "First-Spike-Based Visual Categorization Using Reward-Modulated STDP," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 29, no. 12, pp. 6178-6190, 2018.

[16] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain Spike-Time-Dependent-Plasticity in Neural Synapses," *Nature Precedings*, 1-1, 2009.

[17] S. Saïghi, C.G. Mayr, T. Serrano-Gotarredona, et al. "Plasticity in memristive devices for spiking neural networks." *Frontiers in neuroscience* vol. 9, pp. 51, 2015.

[18] C. Shi, J. Lu, Y. Wang, et al. “Exploiting Memristors for Neuromorphic Reinforcement Learning,” *IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS),* 2021

[19] H. Mao, M. Song, T. Li, et al. "Lergan: A zero-free, low data movement and pim-based gan architecture." *51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 2018.

[20] F. Chen, L. Song and Y. Chen. "Regan: A pipelined reram-based accelerator for generative adversarial networks." *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2018.