SNNOT: Spiking Neural Network with On-chip Training for MIMO-OFDM Symbol Detection

Honghao Zheng, *Student Member, IEEE,* Jiarui Xu, Lingjia Liu, *Senior Member, IEEE,* Yang Yi, *Senior Member, IEEE*

***Abstract*—Advancing 5G and beyond communications require innovative solutions for symbol detection in MIMO-OFDM. Our research leverages Spiking Neural Networks (SNNs), surpassing the efficiency of traditional Deep Neural Networks (DNNs) and Artificial Neural Networks (ANNs) in dynamic wireless contexts. The SNNOT architecture features a novel triplet Spike-Timing-Dependent Plasticity (STDP) learning circuit, utilizing 22FDX technology from GlobalFoundries, which overcomes STDP's supervised learning limitations and enhances dynamic learning by converting spikes into voltages. This innovation leads to substantial efficiency gains, with a minimal energy requirement of $20.92\mu W$ and a small silicon footprint of $37 \times 62\mu m^2$. Performance evaluations show the triplet STDP rule in SNNOT considerably improves image classification and symbol detection, with over 3% error rate reduction in MNIST and CIFAR-10. Additionally, it achieves a 0.07 symbol error rate (SER) at 5 dB E_b/N_o in 4 pulse amplitude modulation (4-PAM) Gaussian channel symbol detection, outperforming other methods. With 8-PAM, it surpasses other methods by at least 0.03 SER and maintains lower or comparable SERs in scenarios using the WINNER II channel model and 16 quadrature amplitude modulation (16-QAM). Importantly, our method offers up to 32% faster processing, providing a potent, efficient solution for receiver processing in cutting-edge wireless systems.**

***Index Terms*—MIMO-OFDM, symbol detection, triplet reconfigurable STDP, on-chip learning**

I. Introduction

Wireless communication, a cornerstone of modern telecommunication, involves transmitting information without physical connectors like wires, using electromagnetic waves such as radio frequencies, microwaves, or infrared. Key components include a transmitter, transmission medium (typically air or space), and a receiver. The invention of Multiple Input Multiple Output - Orthogonal Frequency Division Multiplexing (MIMO-OFDM) symbol detection marks a significant advancement in this field [1], enhancing data transmission rates and robustness against interference. MIMO uses multiple antennas at both ends of the communication system to multiply radio link capacity, while OFDM efficiently combats inter-symbol interference and fading. This complex process, employing advanced algorithms for signal processing, is pivotal in modern standards like LTE and Wi-Fi, improving throughput and reliability in challenging environments with signal scattering and fading.

ANNs have shown great potential for the symbol detection task in the MIMO-OFDM systems, but they still have practical challenges. One significant hurdle in deploying neural network (NN)-based approaches is the requirement of a vast amount of labeled data and a long training time. During the online deployment stage, the online over-the-air (OTA) labeled data in contemporary cellular networks is limited and costly. The low-latency requirement for the cellular system and the dynamic channel environment only allow for a short online training time. Attempts have been made to address this issue by performing extensive offline training with simulated data and conducting online adaptation [2]. Nevertheless, implementing these solutions in real-world situations takes time and effort. Cellular systems adopt dynamic transmission modes, such as link and rank adaptations and scheduling, performed on a subframe (1 millisecond) basis. This rapid alteration can create discrepancies between offline and online models, leading to potential model mismatches and making it challenging to apply offline weights in an online setting.

An approach called StructNet was introduced to solve the abovementioned issues of deploying NNs in practice [3]. By embedding the domain knowledge of MIMO-OFDM systems into the design, StructNet transforms the multi-class symbol classification task into multiple binary decision processes, which allows it to have less trainable weights. In addition, due to the particular design of the NN structure, StructNet augments the OTA training samples by leveraging their symmetric structure. Therefore, it can be learned with only OTA training samples and a short training time.

To utilize the abovementioned mechanism in practical environments, especially in edge computing environments, a spiking integrated circuit (IC) chip with online training capability is necessary since the shortened transmission time can significantly improve the system performance, especially in dynamic wireless communication systems.

Compared with real mammal brains, the traditional ANN has some shortcomings. The brains consume much less power than a conventional ANN computing system of the same size. They are also more stable than conventional ANNs in noisy environments. The difference between the mechanisms of ANN and brains is that biological brains utilize spikes to transfer information [4], while the ANNs use actual numerical signals. The nature of the spike-transmitting mechanism makes neurons only fire spikes when needed. The neurons can stay silent for the rest of the operating periods. In that way,

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, Grant ECCS-2128594, Grant ECCS- 2314813, and Grant CCF-1937487.

Honghao Zheng, Jiarui Xu, Lingjia Liu, and Yang Yi are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA (email: zhenghh@vt.edu; jiaruixu@vt.edu; ljliu@vt.edu; yangyi8@vt.edu).

much unnecessary energy is saved compared to the traditional artificial neural networks. To use this valuable mechanism, researchers started investigating methods to mimic the spike-transmitting structure, and the coming-up neural networks are called the spiking neural networks (SNNs) [5]. In the past decades, many works of SNN have been reported to have way better power efficiency than traditional von Neumann structures and ANNs. For instance, the Intel Loihi2 chip [6] managed to integrate 128 cores in a $31mm^2$ area and can be utilized for robot arm control, modeling diffusion processes for scientific computing applications, and more. While standard CPUs and GPUs consume hundreds of watts of power for those applications, Loihi2 chip only consumes less than 1 watt.

Moreover, the SpiNNaker chip [7] also takes less than 1W of power while mimicking over one thousand spike neurons. The TrueNorth chip [8], designed by IBM, only consumes $65mW$ of power when managed to simulate over one million neurons and 256 million synapses. SNN chips can also be area-efficient due to their highly parallel computational structure. For example, the Morphic chip [9] emulated 2048 neurons and two million synapses while only taking 2.86mm2 of silicon area. The ODIN chip [10] even only takes $0.086mm^2$ of silicon area for 256 neurons and 64000 synapses.

Due to the non-differentiable nature of spikes, the spiking neural networks can not be trained with regular training algorithms. For example, normal backpropagation is unsuitable for SNNs since the spike can not be differentiated to get a gradient. Thus, researchers have come up with various training algorithms for SNN, such as spike timing dependence plasticity (STDP) [11], Hebbian learning [12], surrogate gradients [13] and NormAD [14]. Among them, the most widely commonly used is the STDP learning rule. The learning mechanism inspires it in biological neural systems. With the STDP algorithm, the synaptic weight between neurons is updated with the relative time difference of pre and postsynaptic spikes. There are many STDP rules, which means the weight can be updated in several ways. The most basic rule is the asymmetric STDP rule. The synaptic weight increases when the post-neuron spike arrives after the pre-neuron spike since it shows the relativity of two neurons.

On the contrary, the weight is decreased when the post-neuron spike arrives before the pre-neuron spike because the two neurons are not very relative. Another more advanced STDP rule is the triplet STDP [15]. It takes account of a series of spikes to tune the weight instead of only taking account of a pair of spikes. Thus, it can mimic complex biological neural mechanisms. Moreover, even for the pair-based STDP rules, there are more rules besides the asymmetric rule. Research has verified that these different rules have advantages in different engineering tasks, and researchers have proposed a circuit design that can replicate various STDP learning algorithms [16]. To combine the advantages of the triplet-based STDP rule and the reconfigurable STDP circuits, this work has implemented a triplet-based reconfigurable STDP circuit design. This circuit can apply the triplet-based STDP algorithm to various applications to improve performance.

With the STDP learning rule, SNNs can be trained for tasks such as image classification [17], motor control [18], and speech recognition [19]. However, achieving a balance between classification accuracy, on-chip learning capability, and power/silicon area efficiency remains challenging in hardware research. To broaden the application of SNNs, especially in dynamic environments, neuromorphic chips better support online training. With the limitations of hardware resources, the on-chip learning mechanism should achieve low energy and silicon area consumption and maintain relatively high classification accuracy. However, many hardware implementations of SNNs with limited resources cannot support online learning [20], and those that do often cannot achieve high training accuracy. The challenge is even more pronounced for those who use baseline STDP as the training algorithm. The baseline STDP does not support supervised learning. It can only divide data points into different groups in an unsupervised way. To address this limitation of the STDP training rule, reinforcement learning STDP should be incorporated into the final layer of neurons in the SNN [21]. In this proposed work, the last layer of neurons is connected to the penultimate layer through the reward-based STDP learning synapse. Major contributions of our work are summarized as follows:

- A spiking neural network with an online training mechanism is implemented in the GlobalFoundries 22FDX technology node.
- To the best of our knowledge, it is the first IC design of SNN with the triplet-based reconfigurable STDP learning algorithm and the on-chip training capability.
- Simulation results show that the performance of the triplet-based STDP algorithm of MNIST and CIFAR-10 achieves 3.28% and 3.63% lower error rates when compared to the pair-based STDP algorithm.
- The introduced training network layout design consumes $20.92\mu W$ of power and $37 \times 62\mu m^2$ of silicon area. An opamp-free judgmental circuit is designed to achieve reward-based STDP learning while maintaining high power and area efficiency.
- Hardware/software hybrid test result has shown that the trained neural network can achieve similar or better performance than purely software-based methods while having higher energy efficiency.

For the rest part of this article, the background about the MIMO-OFDM system and StructNet is illustrated in Section II. The STDP training algorithms and the circuit design of the reconfigurable triplet STDP training block are demonstrated in Section III. What's more, Section IV has discussed the circuit post-layout simulation results as well as the training results for the MNIST and CIFAR-10 datasets of two different training algorithms. The testbench setup for the MIMO symbol detection application and the test results for it are shown in Section V while Section VI concludes this article.

There exist several works talking about IC design for other STDP training algorithms. In [22], the author has demonstrated an analog IC design of the asymmetric STDP rule. However, this design requires the use of transconductance amplifiers in both the upper part and lower part, which will increase the power consumption and silicon area. [23] has shown an analog circuit that can realize the reverse asymmetric

STDP rule. Compared to [22], this work is more efficient in circuitry as well as power consumption. However, it still suffers from the incapability of taking into account multiple spikes. What's more, it is also not capable of switching to other STDP algorithms. In [24], the researchers have illustrated an analog IC design of the most basic STDP learning rule, the asymmetric STDP algorithm. Similarly, it is also unable to account for various spikes and switch to other algorithms, which may limit its practical applications.

Moreover, there are also preceding studies on the circuit implementation of triplet-based STDP rule. In [25], the authors have introduced a circuit implementation of the triplet STDP training algorithm. The circuit implementation delays input signals with one clock cycle for the triplet spike weight adjustment. However, in this work, the mathematical formula shows that the STDP algorithm it uses to account for the three spikes is the basic asymmetric STDP learning rule. The inability to switch between different STDP learning rules will lead to inefficiency in specific applications. [26] has demonstrated a digital circuit design of the TSTDP learning rule. The research utilized a Field Programmable Gate Array (FPGA) board to implement the triplet STDP learning circuit. Compared with analog circuits, the FPGA implementation of the triplet STDP rule requires more power, so its efficiency will not be as good as that of analog TSTDP circuits. In [27], researchers proposed an analog training circuit with the triplet STDP rule. The circuitry is novel and power efficient. However, it requires the use of memristor. As a special device, the memristor will limit the application of the proposed circuit since memristors are not as available as standard CMOS devices for potential users. Furthermore, in our prior research [28], we introduced a design of the reconfigurable STDP training circuit. That design, however, has four time window generator blocks that takes a lot of power and silicon area as well as brings implementation complexity. In this study, we present a new, simplified design for the STDP training circuit that only requires two time window generator blocks so that the power and silicon area efficiency can be increased by almost 100%, along with a neural network architecture that incorporates this circuit for training for classification applications.

II. Background

A. Symbol detection in the MIMO-OFDM system

MIMO-OFDM is a pivotal technology in the field of wireless communications, particularly integral to the architecture of modern broadband systems like 4G, 5G, and Wi-Fi networks [1]. In the MIMO-OFDM system, transmitted information symbols are modulated in the frequency domain and then converted to the time domain through an inverse fast Fourier transform (IFFT). A cyclic prefix (CP) is inserted at the beginning of each OFDM symbol to avoid inter-symbol interference (ISI). At the receiver, the time-domain received signal is transformed to the frequency domain through the CP removal process and the fast Fourier transform (FFT). Consider a MIMO system with N_t transmit antennas and N_r receive antennas. The relationship between the transmitted symbols and received signals in the frequency domain at each subcarrier k can be written as

$$\boldsymbol{Y}(k) = \boldsymbol{H}(k)\boldsymbol{X}(k) + \boldsymbol{N}(k), \tag{1}$$

where $\boldsymbol{Y}(k) \in \mathbb{C}^{N_r \times N_s}$ is the received signal in the frequency domain at subcarrier k; $\boldsymbol{X}(k) \in \mathcal{A}^{N_t \times N_s}$ is the transmitted symbols at subcarrier k sampling from the modulation constellation alphabet set $\mathcal{A}$; $\boldsymbol{H}(k) \in \mathbb{C}^{N_r \times N_t}$ is the frequency-domain channel at subcarrier k; N_s is the number of OFDM symbols; and $\boldsymbol{N}(k)$ is the additive white Gaussian noise.

The symbol detection task in the MIMO-OFDM system involves accurately identifying and decoding the data symbols transmitted over multiple antennas and through multiple frequency subcarriers [29]. This process is essential for effectively deciphering the transmitted information in environments where signals can be distorted due to multipath fading, interference, and noise.

The landscape of symbol detection in MIMO-OFDM systems is rich and diverse, encompassing both conventional model-based and emerging learning-based approaches. Traditional model-based methods like Linear Minimum Mean Square Error (LMMSE) [30] and Sphere Decoding (SD) [31] algorithms are foundational in this domain. These approaches rely on explicit system modeling and precise Channel State Information (CSI) estimation. However, their performance can be significantly impacted when non-linear components, such as power amplifiers, are present in the system or when there are inaccuracies in CSI estimation. This limitation of conventional methods under non-ideal conditions has driven the exploration of alternative strategies, particularly those leveraging the capabilities of neural networks.

Learning-based approaches, especially those utilizing neural networks, have gained traction as a means to address the shortcomings of traditional model-based methods. These methods capitalize on the pattern recognition strengths of NNs, offering potential improvements in environments with non-linear distortions or imperfect CSI. However, the implementation of these approaches in wireless communication presents its challenges. The requirement for extensive training data and prolonged training durations is a significant hurdle. Existing works, such as MMNet [32], have explored training large neural network models offline, which raises issues when these models are deployed in real-world scenarios. The discrepancy between the offline training data and the real-time online data distribution can lead to performance degradation. An emerging area of interest is integrating offline learning with online adaptation approaches, which aim to mitigate this issue by continuously updating the model with over-the-air training data on a subframe basis. This approach represents a promising direction for learning-based symbol detection, offering a more adaptive and robust solution for the dynamic environments encountered in wireless communications. Additionally, the development of reservoir computing (RC)-based approaches, including RC-Struct [33] and RC-AttStructNet-DF [3], further illustrates the ongoing evolution and refinement of learning-based strategies in symbol detection for MIMO-OFDM systems.

Recent literature on symbol detection in MIMO-OFDM systems underscores the diversity of approaches and challenges in this field. Zhou Zhou introduced a reservoir computing structure that can significantly improve the symbol detection performance and effectively mitigate model mismatch effects using very limited training symbols [34]. Aswathy K Nair has proposed channel sparsity to design a pilot-based compressed sensing method for channel estimation and integrated a BiLSTM approach for symbol detection for improved performance [35]. Seung-Jin Choi presented a novel MIMO detection algorithm with low complexity and good error performance [36]. Moreover, this method is also expected to decrease the computational complexity of the massive MIMO systems.

B. StructNet

StructNet [3] is a frequency domain classification neural network for the MIMO-OFDM symbol detection task, which incorporates the symmetric structure of the modulation constellation to improve the training efficiency. The design of StructNet originates from the atomic decision neuron network (ADNN) [37]. The ADNN in [37] assumes the perfect channel knowledge and utilizes one learned binary classifier to perform multi-class detection by leveraging the symmetric modulation constellation pattern. For ease of discussion, we refer to this approach as ADNN-GT. In practice, perfect channel knowledge is not directly accessible. Therefore, later on, the ADNN with linear minimum mean square error (LMMSE) estimated channel (ADNN-LMMSE) is introduced for the frequency-domain classification and combined with the time-domain neural network to perform online subframe-based symbol detection [33]. One drawback of ADNN-LMMSE is that the LMMSE estimated channel can not accommodate changing channel environments. To address this issue, StructNet is designed by adopting a linear layer to estimate the channel and dynamically track the channel changes. It is noteworthy that such an NN-based channel estimation process does not require ground truth channels for training.

StructNet consists of two main components: the linear layer for channel estimation and a multilayer perception (MLP) network that works as a binary classifier. The LMMSE estimated channel first initializes the parameters of the linear layer and is then updated along with the binary classification task. As the multi-class classification problem is transformed into multiple binary decision processes with the same binary classifier, StructNet can have more efficient training processes with a limited amount of training data and shorter training time. Moreover, utilizing the channel estimation layer allows it to dynamically track channel variations without assuming knowledge of the ground truth channel.

III. On-chip Learning Spiking Neural Network Design.

A. Baseline STDP

In the realm of SNNs, several algorithms are employed to facilitate their training. Among these algorithms, STDP has emerged as a promising approach. This algorithm adjusts synaptic weights according to the temporal dynamics of spikes. One of the most widely implemented examples is the asymmetric STDP rule [11].

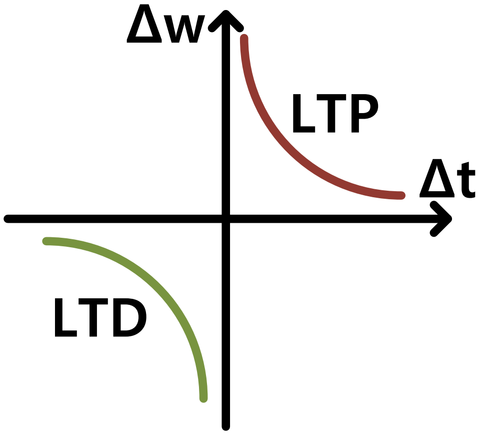


Fig. 1. Diagram of the relation between the time difference of pre-and post-spikes and weight voltage adjustment in the conventional asymmetric STDP learning rule.

As demonstrated in Fig.1, in the asymmetric STDP rule, the synaptic weights are increased when spike timing aligns with the direction of spike propagation, a process known as Long-Term Potentiation (LTP). Conversely, when the post-neuron spike is initiated before the pre-neuron spike—a scenario indicative of a weaker correlation between the two neurons—their synaptic weight undergoes a decrement, a phenomenon referred to as Long-Term Depression (LTD).

The following equation defines the relationship between weight modification and the time difference:

$$\Delta W = \begin{cases} A^{+}e^{-(t_{post}-t_{pre})/\tau}, & t_{post}-t_{pre}>0 \\ -A^{-}e^{(t_{post}-t_{pre})/\tau}, & t_{post}-t_{pre}<0. \end{cases} \tag{2}$$

In the equation, t_{pre} and t_{post} denote the pre- and post-neurons' firing times, respectively. A^{+} and A^{-} signify the maximum values of potentiation and depression. Both A^{+} and A^{-} have the same magnitude but opposite signs. τ represents the time constant, dictating the rate at which the potentiation and depression diminish over time.

Notably, the potentiation and depression values exhibit an exponential relationship with the temporal differences between spikes. This relationship ensures that the algorithm significantly impacts the weight when the spikes are temporally proximate. However, when the spikes are temporally distant, the rule's impact on the weight is minimal. This feature allows the STDP rule to adaptively modify synaptic weights based on the precise timing of pre- and post-neuronal activities, making it a powerful tool for training SNNs.

The asymmetric STDP rule, therefore, has a crucial role in shaping the learning dynamics of SNNs. Its ability to tune synaptic weights based on the relative timing of spikes allows for the precise encoding of temporal information. This ensures that the SNNs can adapt and learn from the changing patterns of spikes. It is the foundation for the temporal processing capabilities of SNNs, which sets them apart from traditional artificial neural networks and brings them closer to the biological realism of the human brain.

B. Triplet STDP

As outlined in Section III.A, the Pair-based Spike-Timing-Dependent Plasticity (PSTDP) rule uses two spikes to adjust the synaptic weight according to their temporal differences. On the contrary, the Triplet-based Spike-Timing-Dependent Plasticity (TSTDP) rule takes into account three spikes [38]. The combination of these spikes can either be pre-post-pre or post-pre-post. These spike combination models illustrate how the temporal differences between spikes are employed to adjust synaptic weights. Compared to the PSTDP rule, the TSTDP can mimic and realize higher-order spiking patterns in the process of training.

The mathematical model for the baseline TSTDP learning rule [15] is expressed as follows:

$$\Delta W = \begin{cases} A_1^+ e^{(\frac{-\Delta t_1}{\tau_1^+})} + A_2^+ e^{(\frac{-\Delta t_2}{\tau_2^+})} e^{(\frac{-\Delta t_1}{\tau_1^+})} \\ -A_1^- e^{(\frac{\Delta t_1}{\tau_1^-})} - A_2^- e^{(\frac{-\Delta t_3}{\tau_2^-})} e^{(\frac{\Delta t_1}{\tau_1^-})}. \end{cases} \tag{3}$$

In this equation, A_1 and A_2 are the potentiation and depression parameters, while Δt_1 denotes the time difference between pre- and post-neuron spikes. Δt_2 equates to $t_{post}(n){-}t_{post}(n-1)$, indicating the temporal gap between two consecutive post-neuron spikes. Here, n signifies a specific time step, and $n-1$ represents the immediately preceding post spike. Similarly, Δt_3 denotes the time gap between two pre-neuron spikes. When comparing the TSTDP mathematical model to the PSTDP formula, it is clear that the TSTDP incorporates a higher-order term in both the depression and potentiation equations. Thus, the time difference between two pre- or post-spikes is also taken into account.

Physiological experiments have illustrated that the TSTDP more accurately emulates biological mechanisms. Initially, researchers identified that the PSTDP rule could not entirely account for the results observed in biological synaptic weight-changing experiments [15]. Moreover, the TSTDP has been shown to accurately reproduce the frequency-dependent effects observed in experimental settings, where the amplitude of potentiation increases with the spike firing rate.

Beyond these advantages, the most significant attribute of TSTDP is its ability to reproduce the behavior of the Bienenstock–Cooper–Munro (BCM) model [39]. With this capability, neurons demonstrate input selectivity when receiving multiple inputs. This behavior closely mirrors the adaptability of biological neurons, bringing SNNs a step closer to achieving the complexity and versatility of the human brain.

The TSTDP rule, thus, represents an essential advancement in the field of SNNs. Its ability to capture higher-order spiking patterns and accurately reflect various biologically observed phenomena makes it a powerful tool in developing and training more sophisticated and biologically realistic neural networks.

C. Reconfigurable STDP

As mentioned in Section III.B, STDP is a biological process that adjusts the strength of synaptic connections between neurons based on the relative timing of their spikes. The STDP rule has become popular in engineering applications because it provides a way to train SNNs to learn from their environment in a biologically plausible way.

The basic asymmetric STDP rule is widely used for unsupervised learning applications in SNNs, where it adjusts the weights of synaptic connections based on the relative timing of pre- and post-synaptic spikes. Typically, when the post-synaptic spike occurs after the pre-synaptic spike, the synaptic weight is increased (potentiation). When the post-synaptic spike occurs before the pre-synaptic spike, the synaptic weight is decreased (depression).

However, the basic asymmetric STDP rule has some limitations, and its application in various engineering applications is only sometimes suitable. Thus, researchers have proposed several other STDP algorithms with different shapes and modified synaptic weights differently based on the relative timing of pre- and post-synaptic spikes.

One such STDP algorithm is the anti-STDP learning rule, which increments the synaptic weight when the post-synaptic spike arrives before the pre-synaptic spike and vice versa. As the name suggests, the anti-STDP learning rule is the opposite of the basic asymmetric STDP rule and is used in some supervised learning applications [40].

Other STDP rules have been developed, such as DPD (Depression-Potentiation-Depression), PP (Potentiation-Potentiation), DD (Depression-Depression), and PD (Potentiation-Depression), where D denotes depression and P denotes potentiation. These STDP rules modify synaptic weights differently based on the relative timing of pre- and post-synaptic spikes and have advantages for different applications. For example, the symmetric STDP rule (DPD) is effective for associative learning [41]. In contrast, the potentiating rule (PP) and depressive rule (DD) are effective in liquid state machine (LSM) [42] and some classification projects.

In conclusion, the selection of the STDP rule depends on the specific application requirements. It is essential to consider both the advantages and limitations of various STDP rules to select the most appropriate one for the specific task. Thus, to broaden the application of a spiking neural network, the capability of switching between different STDP training rules is important. The reconfigurability of the STDP rules of the training circuit can be achieved by employing digital control signals so that the circuit can adapt to various tasks.

D. Reward-based STDP training algorithms

A well-known disadvantage of the conventional STDP training algorithm is that it does not support supervised learning. Rather than classifying data points into known categories, a traditional STDP training circuit can only group data points into clusters, a process referred to as unsupervised learning [43]. However, in real-world applications, supervised learning could be very crucial. With the help of labeled data, supervised learning can achieve much higher accuracy than unsupervised learning. Supervised learning is also more suitable for classification and regression applications such as spam detection, image classification, and weather forecasting, which are more widely used. Moreover, supervised learning models often have less complexity since the outputs are already known, which makes the training procedure more straightforward. To realize the supervised learning of SNN with the STDP circuit, a reward-based STDP training rule is utilized in the proposed neural network.

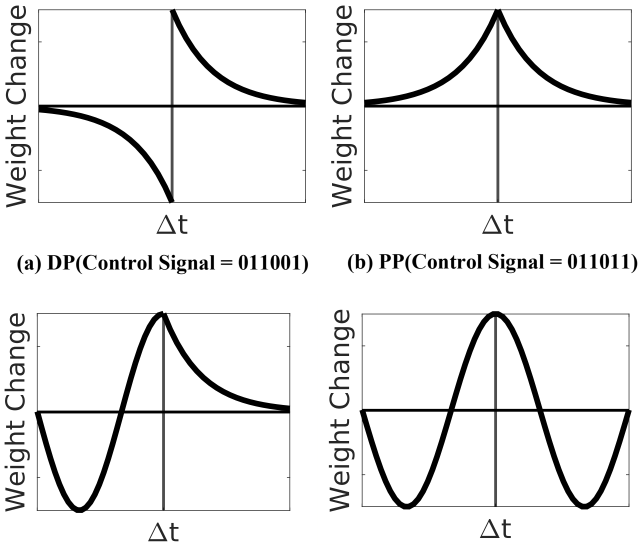
The reward-based STDP algorithm uses the reinforcement learning concept to implement supervised learning for the STDP circuit. As discussed earlier, the weights adjust based on the temporal relationship between pre- and post-spikes. In the reward-based STDP algorithm, the adjustment of synaptic weights will be reversed if the classification result is incorrect [44]. However, if the prediction result for a given cycle is correct, the traditional STDP weight adjustment mechanism will be maintained to reinforce the learning. In this manner, the reward and punishment mechanisms from reinforcement learning are integrated into the originally unsupervised STDP rule, effectively transforming it into a supervised learning rule.

E. Circuit design of the triplet reconfigurable STDP circuit

In MIMO-OFDM systems, the need for on-chip training in symbol detection arises from several key aspects. Firstly, the wireless signal environment in MIMO-OFDM systems often changes, requiring receivers to adapt in real time. On-chip training enables these receivers to learn and adjust to variations in signal characteristics internally, thus enhancing detection accuracy and efficiency. Additionally, on-chip training enhances computational efficiency by reducing reliance on external processing resources and minimizing data transmission needs and delays. This is particularly crucial for applications that demand quick response, such as real-time communication or mobile telecommunications.

Secondly, energy efficiency is a paramount concern in mobile and edge computing devices, which often have strict energy consumption limits. On-chip training optimizes the energy efficiency of algorithms by reducing the need for data transmission outside the chip, thereby lowering overall energy consumption. Furthermore, on-chip training allows for customization and flexibility, as different MIMO-OFDM systems may require optimization for their specific signal characteristics and operational environments. This self-optimization feature improves the adaptability and performance of the system. Lastly, conducting data processing and training on the chip enhances data security, as it reduces the risk of data leakage by avoiding external data transmission. In summary, on-chip training in MIMO-OFDM symbol detection is crucial for improving the system's adaptability, efficiency, and security, especially in scenarios with limited resources and high real-time requirements.

To utilize the advantages of the triplet STDP learning rules on various tasks, especially their most suitable tasks, the triplet reconfigurable STDP circuit is in high demand. As discussed in Section III.C, in addition to the baseline asymmetric STDP learning rule, there are numerous other STDP learning rules, such as PD, DPD, and PDP. Research has demonstrated that each rule has unique applications. To increase the versatility of the proposed circuit, the circuit needs to acquire the feasibility of switching between shapes. Thus, the circuit is controlled by six digital controlling signals. The first three signals control the shape when $\Delta t > 0$, and another three control it when $\Delta t < 0$. For instance, as demonstrated in Fig.2(a), DP, the conventional asymmetric STDP learning rule's control signals are 011001. 011 controls the right half of the waveform and 001 controls the left half where $\Delta t < 0$. When the first three signals and the second three signals are the same, as shown in Fig.2(b), the shape of the learning rule will be symmetric. What's more, the STDP learning rule shape can be more complicated than Fig.2(a) and (b). As illustrated in Fig.2(c), when the control signals are 010, the time and weight changing relation will look like a sine wave. Similarly, when the first and second three signals are both 010, the shape becomes symmetric, as depicted in Fig. 2(d).



(a) DP(Control Signal = 011001) **(b) PP(Control Signal = 011011)**

(c) D1P(Control Signal = 010011) **(d) D1PD1(Control Signal = 010010)**

Fig. 2. STDP learning rule shapes with various control signals.

The circuit can be divided into three main blocks: the Central OTA Core, the Time Window Generator for LTD, and the Time Window Generator for LTP. The Time Window Generator is shown in Fig. 3. It compares the time difference between the pre-and post-neuron spikes and generates voltage differences. The generator for LTP creates voltage differences that result in an increased weight voltage, while the generator for LTD causes the weight voltage to decrease. The generator for LTP will have pre-spikes for the Spike1 input and post-spikes for the Spike2 input. For example, when the control signals are set to 011001, which means the Vctr1 = 0, Vctr2 = 1, Vctr3 = 1 and Vctr1B = 0, Vctr2B = 0, and Vctr3B = 1, the training circuit represents the DP learning rule. As discussed in the last paragraph, Vctr1, Vctr2, and Vctr3 control the right side of the STDP learning rule shape, where pre-spike arrives before post-spike. As for Vctr1B, Vctr2B, and Vctr3B, they control the left side of STDP shape, where post-spike arrives before pre-spike. Since Vctr1 = 0, the comparator in the LTP generator outputs ground signal, and Vcont equals

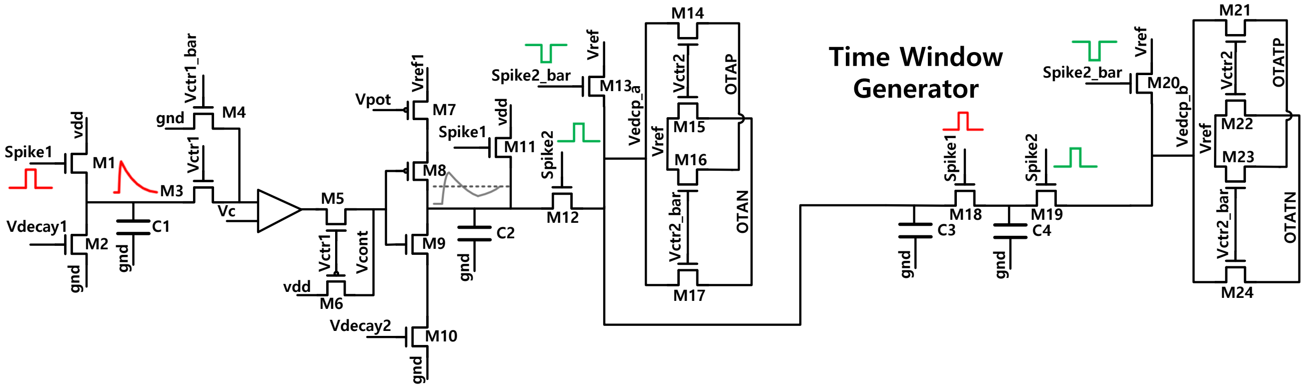


Fig. 3. Schematic of the time window generator circuit.

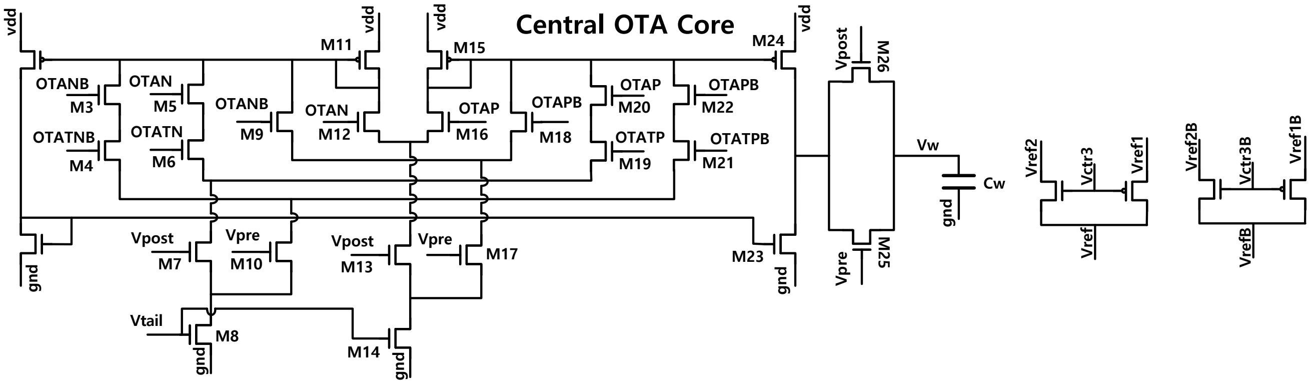


Fig. 4. Schematic of the central OTA core circuit.

supply voltage vdd. With that, the voltage across C2 will be increased to vdd when the pre-spike fires and slowly decrease after the pre-spike at a rate controlled by Vdecay2. Similarly, if Vctr1 = 1, the voltage across C1 will work just like C2 voltage and then be compared with Vc. The comparison result will be output to Vcont in this situation. When the post-spike fires, Vedcp_a will equal the C2 voltage; otherwise, it will equal Vref, which is controlled by Vctr3 of whether it equals Vref1 or Vref2. In this case, since Vctr3 = 1, Vref equals Vref2. After that, Vctr2 will switch the voltages OTAN and OTAP between Vedcp_a and Vref. From that, we can notice that the only difference between the OTAN and OTAP will be the C2 voltage during post-neuron spikes. By comparing these two voltages, we can realize the first-order relation of formula 2.

Moreover, Vedcp_a will also be stored in the capacitor C3. When the next pre-spike occurs, the voltage will be stored in C4 and will cause a switch in Vedcp_b between Vref and the voltage across C4. Thus, the pre-spike is postponed to the next pre-spike and compared with the reference voltage. Also, with the same switch mechanism of Vctr2 between OTATP and OTATN, the second-order relationship between the pre-post-pre spike train is realized.

Similarly, the LTD generator follows the relationship between the post-pre-neuron spike train. The pre-spike and post-spike are switched in this block. In this situation, Vctr2B = 0. Thus, OTAPB equals VrefB, and OTANB equals voltage Vedcp_c that switches between C6 voltage and VrefB. For the second-order relationship between the post-pre-post spike train in Formula 2, the LTD generator follows a similar mechanism as the LTP generator. The only difference is that the post- and pre-spikes are switched.

Looking into the Central OTA Core, as demonstrated in Fig. 4, it is noticeable that the outputs of each generator are all compared in pairs. For instance, the OTAP and OTAN are compared, and OTATN and OTATP are also compared. With each compared pair, the difference will be transferred to current signals through M23 and M24. It will either inject current into or draw current from the weight capacitor at the firings of pre-and post-spikes, triggering the increasing and decreasing of the weight voltage.

There are several difficulties when designing this IC implementation of the triplet-based reconfigurable STDP learning rule. The first one is realizing the functionality of switching between different STDP shapes. To solve that issue, the time window generator is designed. The time windows can be divided into four situations, the post-spike arrives after the pre-spike, the post-spike arrives before the pre-spike, the pre-post-pre spike train when accounting for three spikes and the post-pre-post spike train when accounting for three spikes. Thus, four time window generators are needed for the circuit implementation. To simplify the circuit implementation, signals from the two-spike time window generators are used as the inputs of the three-spike generators, as discussed in Section III.E. What's more, another difficulty is to store the last spike

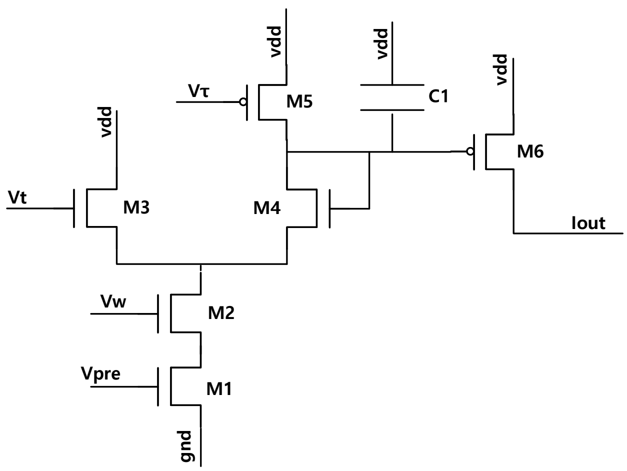


Fig. 5. Schematic of the spiking synapse circuit.

until the next spike so that a three-spike train can be used for the weight adjustment. We used two small capacitors as well as two spike-triggered switches to realize this function, as demonstrated in Section III.E. Another issue we faced in the designing process was how to utilize the output differences from time window generators and convert them to weight voltage changes. An operational transconductance amplifier with 4 differential pairs is designed for that purpose. With one differential pair, the charge on the weight capacitor will be accordingly increased or leak away, as illustrated in the last paragraph.

F. Circuit design of the spiking synapse

Since the synapses need to trigger spiking neurons in this neural network, spiking synapses that can supply firing current according to weight voltages should be specifically designed. The synapse should output a current spike signal in response to a firing pre-neuron spike. The larger the weight voltage, the larger the output current should be. This mechanism will increase the charges stored on the post-neuron membrane capacitor. As several input currents from different synapses accumulate, the membrane voltage will reach the post-neuron's threshold voltage. In that way, the post-neuron will be firing a spike, which will be most relevant to the pre-neuron.

As shown in Fig. 5, the synapse works like a current mirror. When pre-neuron spikes, the tail current starts to flow, and with higher weight voltage comes the larger tail current. Vt is used to tune the current on this side. Large Vt will lead to a large current flow in M3 and will cause the current through M4 to decrease. Conversely, $V\tau$ is a parameter that adjusts the time constant of the current mirror. A larger $V\tau$ voltage results in a larger parasitic resistance of M5 and an increased time constant. As a result, the voltage across the capacitor and the output current will increase, with the current slowly decreasing back to zero.

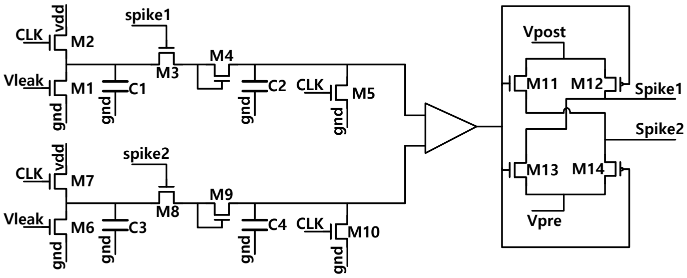


Fig. 6. Schematic of the judgmental and reward-based circuit.

G. Circuit design of the judgmental circuit

The judgmental circuit is utilized to judge whether the classification result of the output neuron is correct and give reward or punishment feedback according to it. Firstly, the network needs to define the classification result. In this case, the circuit must recognize which output neuron fires the first spike in one sampling window. To realize that, we can have a voltage that resets every sampling window and decreases exponentially. Then, we can capture the voltage at different spike timings. Thus, an earlier spike results in a larger captured voltage. Subsequently, the captured voltages are compared after each clock cycle, with the highest voltage indicating the winner. The winner will then be compared with the desired result. If it matches, reward feedback is sent to all synapses connected to this neuron, enhancing their relevance. If not, punishment feedback will be sent to the synapses.

The clock signal will be connected to M1 to provide initial voltage to C1, as depicted in Fig. 6. With the leak transistor M2 and C1, the voltage will decrease exponentially at a rate controlled by Vleak. When the spike fires, M3 will be open and push the C1 voltage to the peak detector formed by the diode-connected transistor M4 and capacitor C2. This peak detector will capture and hold the voltage until the next sampling window is reset by M5. Afterward, a comparator is implemented to compare the captured voltages from neurons. The desired result can change with different classes. It can be realized by simply switching the inputs of the comparator. When the comparator outputs a high digital voltage, the pre- and post-spike pairs for all the STDP training circuits remain the same to enhance the training weight. If the comparator outputs a low voltage, the pairs of pre- and post-neuron spikes are internally switched to reduce their relevance.

IV. Result Analysis of the On-chip Learning Spiking Neural Network

A. Result analysis of the triplet reconfigurable STDP circuit

As mentioned in Section II, the proposed STDP circuit can take into account spike trains and adjust weights according to their temporal relationship. Moreover, the reconfigurable feature enables the circuit to switch between different ways of adjusting weights. For instance, when a pre-spike fires before a post-spike, the weight will be increased in the conventional DP STDP learning rule, while the weight can also be decreased in the PD STDP rule. Different rules have their specific suitable tasks, as discussed in Section II.C.

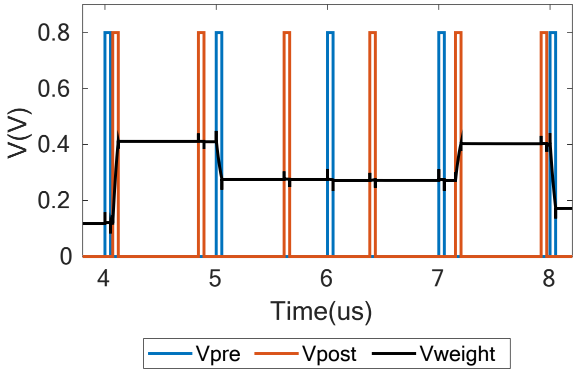


Fig. 7. Post-layout simulation result of the triplet reconfigurable STDP circuit when Vctr = 011001.

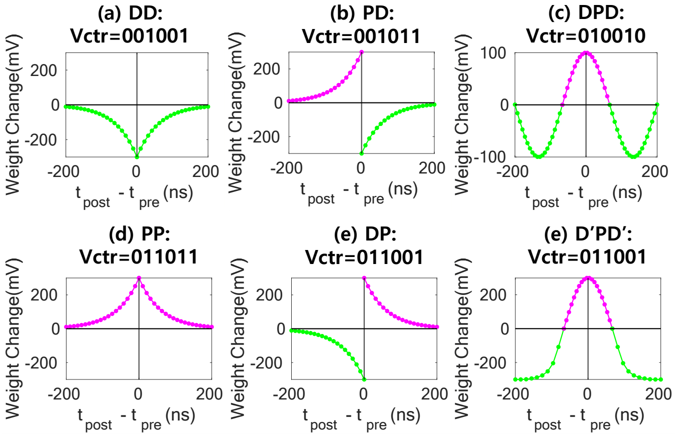


Fig. 8. Simulation Results of different triplet STDP algorithms controlled by various Vctrs.

The post-layout simulation result of the proposed STDP circuit when the control signal is equal to 011001 is demonstrated in Fig. 7. It is noticeable that the weight voltage always tends to increase when post-spike fires and decrease when pre-spike fires. The closer the pre- and post-spikes are to each other, the larger the change in weight voltage will be, whether it is an increase or a decrease. From that, the relationship between the spike train's time intervals and the weight voltage almost matches the relationship shown in Fig. 1. However, the high-order relationship of the triplet spike train may not be that obvious in this figure. This phenomenon occurs because the amplitude constant, A_2, in the high-order relationship term of formula 2, is much smaller than the constant A_1. This indicates that the effect of the high-order term is much smaller than the first-order term's effect.

To verify the reconfigurable feature of the circuit, we switched the control signal and tested the relationship between the time difference and the weight voltage. Fig. 8 depicts 6 of the different shapes. Although not all the shapes that can be realized in the reconfigurable STDP circuit, it still demonstrates that the circuit can adjust weight voltage in different ways controlled by digital signals.

B. Result analysis of the synapse and judgmental circuits

The task of the spike synapse circuit is to accept a pre-neuron spike and weight voltage and then provide current signals to prompt the post-neuron to fire spikes. With a larger weight voltage, the synapse should be capable of outputting a correspondingly larger current signal. Fig. 9 illustrates the post-layout simulation results of the weight voltage and current value relationship with different Vts and $V\tau = 0.7V$ as well as $Vt = 0.6V$ and different $V\tau$s. With the same $V\tau$ or Vt, relations between the output current and Vt or $V\tau$ are revealed. With larger Vt, the output current becomes smaller. On the contrary, larger $V\tau$ leads to larger current output. It is noticeable that although the current value follows the weight voltage, it has a threshold voltage of around 0.3V and also reaches saturation status after 0.6V. This situation is because of the property of the NMOS transistor M2.

As for the judgmental circuit, it converts the spikes back into voltages and compares these voltages. Spikes closer to the CLK signals should result in larger voltages. As depicted in Fig. 10, Spike1 is transferred to Output1, and Spike2 is transferred to Output2. Since Spike1 fires earlier in the sampling windows, Output1 is larger than Output2 in such situations. This simulation result demonstrates that the proposed judgmental circuit can compare the outputs of neurons and determine the winner.

C. Result analysis of the spike training network

Consisting of all the aforementioned blocks, the spike on-chip learning network is designed and implemented in the GlobalFoundries 22FDX process. It occupies a silicon area of $37 \times 62\mu m^2$, as demonstrated in Fig. 11. Starting from the Input Layer, it also has the Synapse and STDP training Circuit, the Output Layer, and the Judgmental Circuit. In the post-layout simulation, the network consumes $20.92\mu W$ of power. The input signals go into the input neurons, and the spike will be transmitted to the synapses and STDP circuit. With the current inputs from the synapses, the output neuron fires spikes, and the spikes will be compared to the desired outputs and then utilized to train the weights.

Moreover, testbenches for spiking neural networks using both the pair-based and triplet-based STDP learning rules are implemented in PyTorch. Compared with symbol detection tasks, the MNIST [45] and CIFAR-10 [46] datasets are also classification applications. Thus, performance verification based on these two datasets can indirectly show the functionality of the proposed neural network for symbol detection. What's more, as commonly used testbenches for machine learning classification tasks, MNIST and CIFAR-10 can help with widening the introduced training neural network's potential applications. The two neural networks that use PSTDP and TSTDP learning rules are both trained for MNIST and CIFAR-10 datasets. Fig. 12 depicts the error rates of the two networks for both the MNIST and CIFAR-10 datasets. The figure has shown that the triplet STDP learning rule can achieve lower error rates than the pair-based STDP training algorithm. The actual value of the error rate differences after 20 epochs for MNIST and CIFAR-10 are 3.28% and 3.63%, respectively.

TABLE I
COMPARISON OF THE TRIPLET RECONFIGURABLE STDP CIRCUIT IN THIS WORK WITH OTHER STATE-OF-THE-ART WORKS

	[47]	[48]	[16]	This work
# of STDP Shapes	1	2	8	8
Technology Node	10nm	180nm	65nm	22nm
Vdd (V)	0525	1.8	1.2	0.8
Area (mm^2)	0.003	0.006	N/A	0.0009
Energy/SOP (fJ)	3.8×10^3	1.2×10^7	400	178
Static Power (μW)	9420.8	N/A	N/A	19.2

TABLE II
COMPARISON OF SNN ENERGY EFFICIENCY RELATIVE TO ANN ($= E_{ANN}/E_{SNN}$) WITH DIFFERENT NEURAL NETWORK TOPOLOGIES AND NUMBERS OF TIMESTEPS

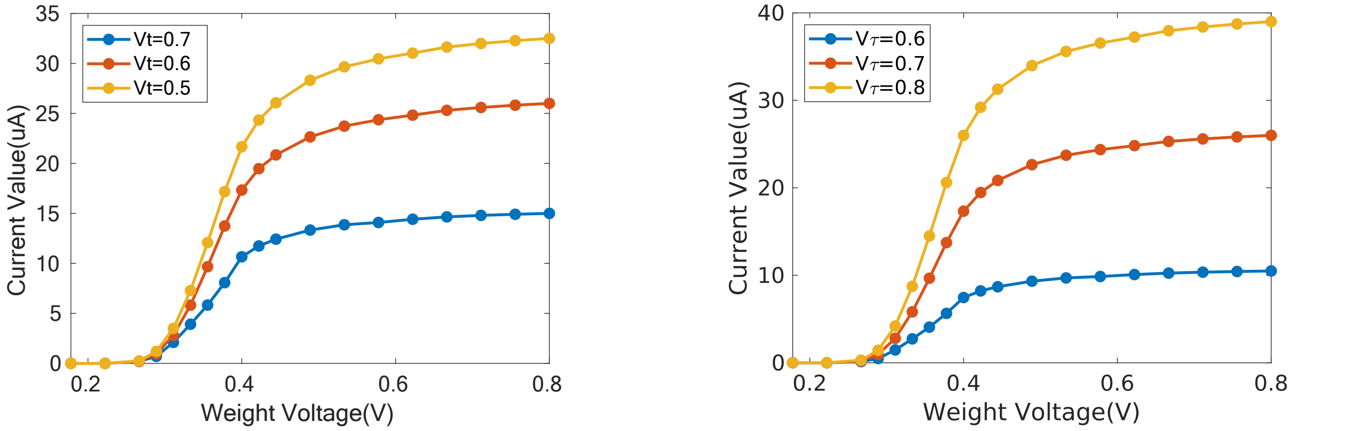
# of Timesteps	AlexNet	VGG16	MobileNet
400	2.0	1.7	1.4
600	1.7	1.4	1.2
800	1.5	1.2	1.0
1000	1.4	1.1	0.9

What's more, it is noticeable that the error rates have almost reached saturation after 20 epochs. Although the error rates still improve with epochs, the improving rates are not as high as those before 20 epochs. This figure has also proved that the triplet STDP rule achieves higher accuracy than the pair-based STDP rule.

We also compared the triplet reconfigurable STDP circuit from the proposed work with those from other state-of-the-art works. We have compared the number of STDP shapes the works can switch between, their silicon area, and their dynamic energy consumption based on their technology node, as illustrated in Table I. The dynamic energy consumption is represented by energy per spike operation, which means the energy consumed with one spike operation by the works. With the help of the advanced technology node and carefully selected component size, this work's dynamic energy consumption is significantly lower than others. Moreover, the static power of our work is significantly smaller than the first work. The results of this comparison demonstrate that the STDP circuit in our work achieves superior reconfigurability, a significantly smaller silicon area, and higher energy efficiency.

Moreover, in Table. II, we have also summarized the energy efficiency ratio between SNN and ANN. According to [49], the power consumptions of SNN and ANN are compared for different neural network topologies, including AlexNet, VGG16 and MobileNet. The SNN energy efficiency is also compared across various numbers of timesteps. From the result, we can notice that although the SNN energy efficiency relative to ANN decreases with the increase of timesteps across all the neural network topologies, the energy consumption ratios are larger than 1.0 in most of the scenarios. It shows that SNN is more power efficient for almost all applications and situations.

To verify the impact of Process-Voltage-Temperature (PVT) variations on the proposed circuit, three PVT simulations are carried out. The first is for the process variations and five corner situations are set up for this simulation. They are FF, FS, NN, SF and SS, in which the first character means NMOS operation speed and the second means PMOS operation speed. F means fast, S means slow and N means normal. The second simulation is for the supply voltage variation and three corners are set up for it. They are 0.78V, 0.8V and 0.82V. Last but not least, for the temperature variation simulation, 0°C, 27°C and 80°C are set as the corners. After the PVT simulation, the result shows that the triplet reconfigurable STDP training functionality is not affected by the PVT variations. The weight adjusting is stable among all the corners and have the same results as the normal situation.



(a) Simulation results of current output and weight voltage relationship of the synapse with Vτ = 0.7V and different Vt.

(b) Simulation results of current output and weight voltage relationship of the synapse with different Vτ and Vt = 0.6V.

Fig. 9. Simulation results of current output and weight voltage relationship of the synapse.

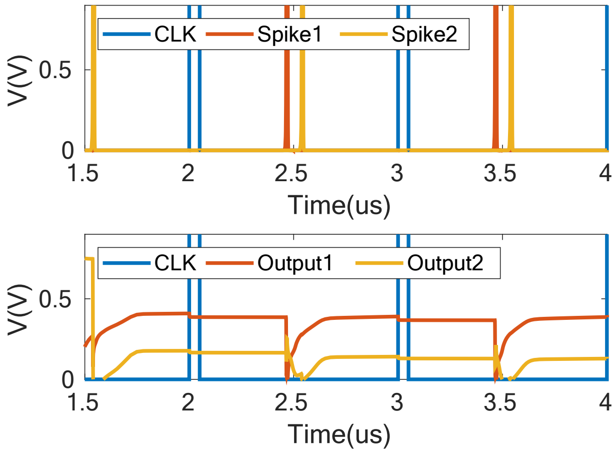


Fig. 10. Simulation result of the judgmental circuit when comparing two neuron's output spikes.

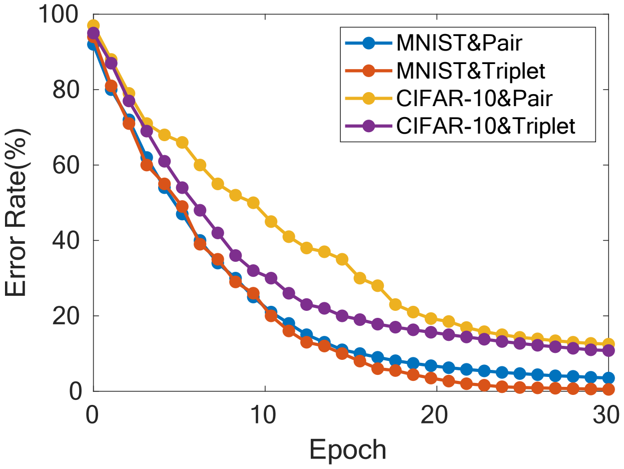


Fig. 12. Error rates of the pair-based STDP and triplet STDP learning rule for the MNIST and CIFAR-10 datasets.

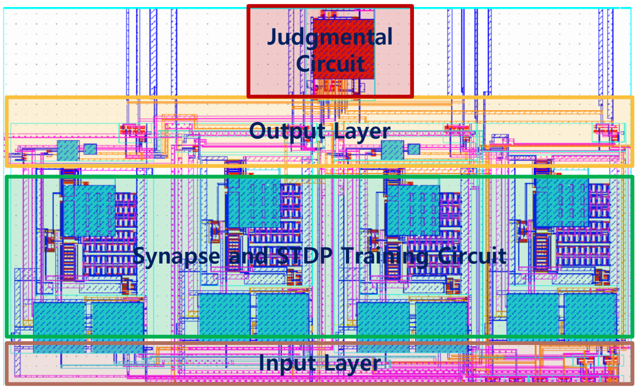


Fig. 11. Layout of the spike on-chip training network.

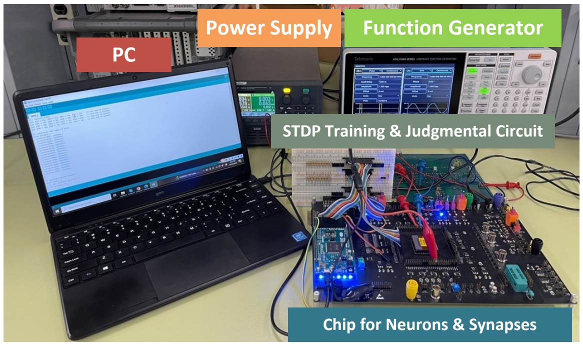

Power Supply
Function Generator
PC
STDP Training & Judgmental Circuit
Chip for Neurons & Synapses

Fig. 13. Hardware/software hybrid experiment setup for the StructNet MIMO-OFDM symbol detection task.

V. Experimental Setup and Result Analysis for the MIMO-OFDM Symbol Detection

To demonstrate the application of our designed neural network training strategies in the MIMO-OFDM symbol detection, we have implemented a hardware/software hybrid testbench for this task. In the following sections, the detailed setup of the testbench, the mechanism of this setup, and the test result of our introduced method will be discussed and will be compared with purely software-based methods.

A. Testbench setup for the application of MIMO symbol detection

As discussed in Section II.E, StructNet is composed of two main parts: the linear layer for the channel estimation and an MLP network for the binary classification. To train StructNet for the symbol detection task in the testbench, the designed testbench also has two main parts: the software part for the linear layer training and symbol error rate (SER) calculation and the hardware part for the MLP neural network with spiking signal transmission which works as a binary classifier.

As shown in Fig. 13, the testbench consists of a laptop, a chip over print circuit board (PCB) for the neurons and synapses, demonstrated in Fig. 14, another PCB board with the reconfigurable triplet STDP training circuits and judgmental circuits, pictured in Fig. 15, a power supply, and a signal generator for the clock signal. Firstly, the symbol detection dataset is put through the pre-processing, which includes the linear layer. Afterward, the transformed data is input into the microcontroller on the board and will be transferred to the voltage signal and then current signals into the neurons. With the leaky integrate and fire (LIF) neurons and spiking synapses, the current signals will be transmitted in the neural network as spiking signals. On the other hand, the triplet reconfigurable STDP will take into account the output spikes of both the pre-neurons and post-neurons and adjust the weight according to their time correlations.

The weights stored in the form of voltages with the training circuit will be transferred back to the synapses on the chip for the output current tuning. The judgmental circuit on the board will compare the spikes from the output layer and decide whether the result is correct. Accordingly, a reward or punishment will be given to the training circuit. When the classification result is correct, a reward will be delivered to the circuit so that the training circuit will adjust the weight as usual. On the other hand, when the training result does not match the label, a punishment will be delivered to the STDP circuit so that the pre-neuron and post-neuron spikes

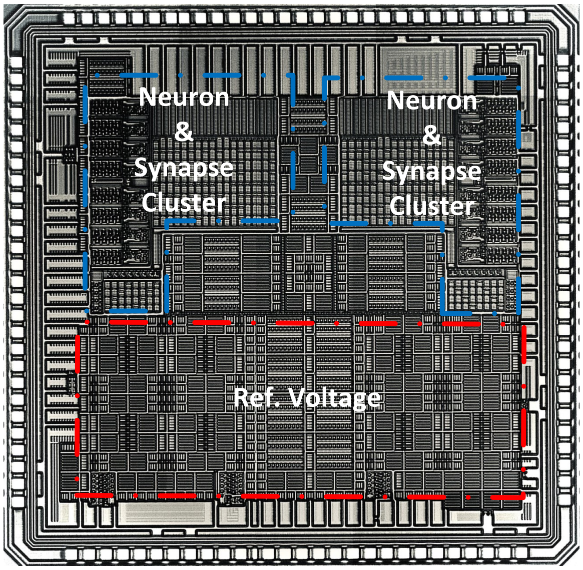


Fig. 14. Circuit chip for neurons and synapses.

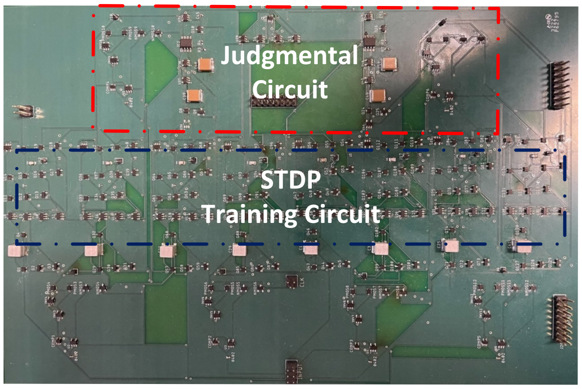


Fig. 15. PCB for STDP training and judgmental circuit.

will be switched. The weight voltage adjustment will be correspondingly opposite to the regular changes. What's more, the classification results are also transmitted back to the PC through the microcontroller. The linear layer in the software parts will be trained with the result. The SER is calculated simultaneously with the training process.

B. Result analysis of training neural network for the MIMO symbol detection

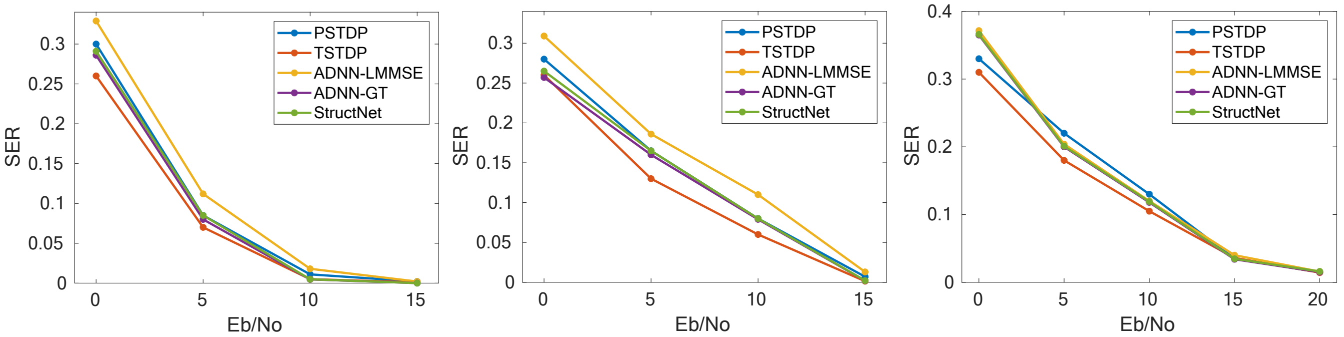
After each time step, the classification result will be sent back to the PC, and the SER will be calculated, as mentioned in the last section. The experimental setting follows the work in [3]. Specifically, we consider the MIMO systems with 2 transmit antennas and 2 receive antennas. We first compare the performance of different approaches in the Gaussian channel under both 4-pulse amplitude modulation (4-PAM) and 8-PAM modulation. Then we evaluate different approaches in a more practical setting, where the WINNER II channel model [50] and 16 quadrature amplitude modulation (16-QAM) are considered. The transmitter and receiver employ uniform linear arrays with half-wavelength antenna spacing. Additionally, the channel scenario is set as urban macrocell non-line-of-sight (NLOS) outdoor to indoor environments. More simulation results for higher modulation orders, 3GPP-3D channel model, and larger antenna numbers have been provided in [3]. The dataset consists of 4992 samples. Among them, 1992 samples were used for the training process, while 3000 samples were adopted for testing purposes [3]. For each sample, the input dimension equals to the number of receive antennas. Following the design of StructNet in [3], the neural network is trained for each transmit antenna. The output dimension is 1, which is the class of each transmitted symbol.

We have compared the performance of training neural networks with the pair-based STDP algorithm and the triplet-based STDP algorithm. Moreover, these two training strategies are compared with purely software-based neural networks, including ADNN-GT, ADNN-LMMSE, and StructNet. It is worth noting that, unlike the method we introduced in this paper, StructNet, in this comparison, is purely software-based.

As depicted in Fig. 16, the SER is plotted as a function of the bit energy to noise ratio (E_b/N_o). We have compared those works at different E_b/N_o regimes, including 0 dB, 5 dB, 10 dB, and 15 dB. The 20 dB of E_b/N_o regime is also included in the last experiment. As demonstrated in Fig. 16(a), in the Gaussian channel model with 4-PAM modulation, compared with the PSTDP, the TSTDP training algorithm achieves lower SER in different E_b/N_o regimes. Moreover, it also has a lower SER than the ADNN-LMMSE model. Python-based StructNet and the TSTDP training network-based StructNet have very similar SER in almost every E_b/N_o regime. However, when the E_b/N_o is 5dB, TSTDP performs slightly better than the other two models. In Fig. 16(b), the TSTDP performs better than all the other methods in all E_b/N_o regimes except 0 dB for the Gaussian channel model under 8-PAM modulation. In this system, the software-based StructNet performs similarly to ADNN-GT and the PSTDP learning rule has higher SERs than ADNN-GT and the software-based StructNet but lower SERs than ADNN-LMMSE. Moreover, as illustrated in Fig. 16(c), the SERs of different methods are very close to each other in the WINNER II channel model with 16-QAM modulation. However, it is still noticeable TSTDP training method achieves the lowest SERs in 0 dB, 5dB, and 10 dB. Besides that, the purely software-based StructNet performs similarly to ADNN methods. Compared with PSTDP, the ADNN methods have lower SER except at 0 dB of E_b/N_o regime.

The processing time of the introduced training neural network implementation and other Python-based symbol detection methods are compared in Table III. The running time of the introduced implementation includes the CPU running time for pre-and post-processing and hardware running time for the on-chip learning, while the training time for other methods only has CPU running time. As a result, we have noticed that ADNN methods require a shorter processing time. However, ADNN-GT has assumed perfect channel knowledge, while the ADNN-LMMSE has a higher SER than other methods in almost every E_b/N_o regime. On the other hand, the SNNOT with PSTDP algorithm has achieved a faster training speed than RC-Struct and RC-AttStructNet-DF, while the SNNOT with TSTDP has a training speed between RC-Struct and RC-

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2024.3393854



(a) Experiment Results for 4-PAM and Gaussian channel model. **(b) Experiment Results for 8-PAM and Gaussian channel model.** **(c) Experiment Results for 16-QAM and WINNER II channel model.**

Fig. 16. Experiment results for the MIMO symbol detection.

TABLE III
RUNNING TIME COMPARISON OF THE SNNOT AND OTHER PYTHON-BASED SYMBOL DETECTION METHODS

Method	Running time(S)
ADNN-GT	1.68
ADNN-LMMSE	3.58
RC-Struct [33]	18.27
RC-AttStructNet-DF [3]	29.57
SNNOT with PSTDP	16.35
SNNOT with TSTDP	20.26

AttStructNet-DF. The main reason the SNNOT with PSTDP runs faster than the SNNOT with TSTDP is that PSTDP only considers two spikes when adjusting the weight. On the other hand, the TSTDP learning rule takes into account three spikes when it changes the weight. As a result, the circuit implementation of TSTDP will be more complex than PSTDP and thus will take more time to process the input signals. However, this tradeoff between the complexity and test performance is worthwhile after comparing the results.

These results have indicated that the ASIC implementation of the MIMO-OFDM symbol detection networks can achieve similar or even better performance than von Neumann structure-based methods, which provides the potential of utilizing AI chips for the edge computing tasks of wireless communication. Moreover, due to the use of the spiking signal transmission, the power efficiency of this work is much better than that of Python-based symbol detection methods.

VI. Conclusion

This paper has introduced an SNN with on-chip STDP learning capabilities, designed using the GlobalFoundries 22FDX technology node. The STDP learning rule can take into account three spikes to capture higher-order spiking patterns and accurately reflect various biologically observed phenomena. Moreover, the STDP circuit in the training network can switch between different shapes of STDP learning algorithms so that the training network can adapt to various applications. We also designed a spiking synapse capable of adjusting current output in response to weight voltages, as well as a judgmental circuit that compares spikes, defines the winner, and provides either reward or punishment feedback to the STDP training circuit. Testbenches implemented in PyTorch for both the pair-based and triplet-based STDP learning networks have demonstrated that the triplet STDP learning rule can achieve error rates 3.28% and 3.63% lower for the MNIST and CIFAR-10 datasets, respectively. Moreover, the post-layout simulation shows that the training network takes $37 \times 62 \mu m^2$ of silicon area and consumes $20.92 \mu W$ of power. A comparison with other state-of-the-art works demonstrates that our approach offers superior reconfigurability, significantly smaller silicon area, and dynamic energy consumption. Moreover, this introduced work has shown comparable or better performance with lower power consumption than other Python-based methods for symbol detection applications in various experiment settings, revealing a promising potential of the training network as well as the reconfigurable triplet STDP training algorithm for wireless communication applications.

References

[1] A. Rawat, R. Kaushik, and A. Tiwari, “An overview of mimo ofdm system for wireless communication,” *International Journal of Technical Research & Science*, vol. 6, no. X, pp. 1–4, 2021.

[2] H. Ye, G. Y. Li, and B.-H. Juang, “Power of deep learning for channel estimation and signal detection in ofdm systems,” *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[3] J. Xu, L. Li, L. Zheng, and L. Liu, “Detect to learn: Structure learning with attention and decision feedback for mimo-ofdm receive processing,” *IEEE Transactions on Communications*, 2023.

[4] R. Brette, “Philosophy of the spike: rate-based vs. spike-based theories of the brain,” *Frontiers in systems neuroscience*, vol. 9, p. 151, 2015.

[5] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[6] M. Davies *et al.*, “Taking neuromorphic computing to the next level with loihi 2,” *Intel Newsroom Technology brief*, 2021.

[7] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, “The spinnaker project,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[8] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam *et al.*, “Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip,” *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[9] C. Frenkel, J.-D. Legat, and D. Bol, “Morphic: A 65-nm 738k-synapse/mm˜2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning,” *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 5, pp. 999–1010, 2019.

[10] C. Frenkel, M. Lefebvre, J.-D. Legat, and D. Bol, "A 0.086-mmˆ212.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 1, pp. 145–158, 2018.

[11] N. Caporale and Y. Dan, "Spike timing–dependent plasticity: a hebbian learning rule," *Annu. Rev. Neurosci.*, vol. 31, pp. 25–46, 2008.

[12] R. Kempter, W. Gerstner, and J. L. Van Hemmen, "Hebbian learning and spiking neurons," *Physical Review E*, vol. 59, no. 4, p. 4498, 1999.

[13] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.

[14] N. Anwani and B. Rajendran, "Normad-normalized approximate descent based supervised learning rule for spiking neurons," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[15] J.-P. Pfister and W. Gerstner, "Triplets of spikes in a model of spike timing-dependent plasticity," *Journal of Neuroscience*, vol. 26, no. 38, pp. 9673–9682, 2006.

[16] C. Huan, Y. Wang, X. Cui, and X. Zhang, "A reconfigurable mixed signal cmos design for multiple stdp learning rules," in *2020 IEEE 3rd International Conference on Electronics Technology (ICET)*. IEEE, 2020, pp. 639–643.

[17] T. Iakymchuk, A. Rosado-Muñoz, J. F. Guerrero-Martínez, M. Bataller-Mompeán, and J. V. Francés-Víllora, "Simplified spiking neural network architecture and stdp learning algorithm applied to image classification," *EURASIP Journal on Image and Video Processing*, vol. 2015, pp. 1–11, 2015.

[18] S. Glatz, J. Martel, R. Kreiser, N. Qiao, and Y. Sandamirskaya, "Adaptive motor control and learning in a spiking neural network realised on a mixed-signal neuromorphic processor," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9631–9637.

[19] J. Wu, Y. Chua, and H. Li, "A biologically plausible speech recognition framework based on spiking neural networks," in *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[20] K. Bai, L. Liu, and Y. Yi, "Spatial-temporal hybrid neural network with computing-in-memory architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2850–2862, 2021.

[21] M. Shahsavari, D. Thomas, A. Brown, and W. Luk, "Neuromorphic design using reward-based stdp learning on event-based reconfigurable cluster architecture," in *International Conference on Neuromorphic Systems 2021*, 2021, pp. 1–8.

[22] Z. Yang, Z. Han, Y. Huang, and T. T. Ye, "55nm cmos analog circuit implementation of lif and stdp functions for low-power snns," in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2021, pp. 1–6.

[23] P. J. Srinidhi, T. R. Yashaswini, N. Uttunga, S. A. Ali, and M. R. Ahmed, "Implementation of stdp based learning rule in neuromorphic cmos circuits," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, pp. 1105–1110.

[24] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 717–737, 2014.

[25] S. J. S and B. J. Kailath, "Bistable-triplet stdp circuit without external memory for integrating with silicon neurons," in *2021 IEEE World AI IoT Congress (AIIoT)*, 2021, pp. 0297–0302.

[26] C. Lammie, T. J. Hamilton, A. van Schaik, and M. Rahimi Azghadi, "Efficient fpga implementations of pair and triplet-based stdp for neuromorphic architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1558–1570, 2019.

[27] S. Aghnout, G. Karimi, and M. R. Azghadi, "Modeling triplet spike-timing-dependent plasticity using memristive devices," *Journal of Computational Electronics*, vol. 16, pp. 401–410, 2017.

[28] H. Zheng and Y. Yi, "Enhancing snn training performance: A mixed-signal triplet reconfigurable stdp circuit with multiplexing encoding," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2023, pp. 1–5.

[29] Y. G. Li, J. H. Winters, and N. R. Sollenberger, "Mimo-ofdm for wireless communications: Signal detection with enhanced channel estimation," *IEEE Transactions on communications*, vol. 50, no. 9, pp. 1471–1477, 2002.

[30] J. Eilert, D. Wu, and D. Liu, "Implementation of a programmable linear mmse detector for mimo-ofdm," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 5396–5399.

[31] Y. Dai, S. Sun, and Z. Lei, "A comparative study of qrd-m detection and sphere decoding for mimo-ofdm systems," in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1. IEEE, 2005, pp. 186–190.

[32] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive mimo," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[33] J. Xu, Z. Zhou, L. Li, L. Zheng, and L. Liu, "RC-Struct: a structure-based neural network approach for MIMO-OFDM detection," 2022.

[34] Z. Zhou, L. Liu, and H.-H. Chang, "Learning for detection: Mimo-ofdm symbol detection through downlink pilots," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3712–3726, 2020.

[35] A. K. Nair and V. Menon, "Joint channel estimation and symbol detection in mimo-ofdm systems: A deep learning approach using bi-lstm," in *2022 14th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2022, pp. 406–411.

[36] S.-J. Choi, S.-J. Shim, Y.-H. You, J. Cha, and H.-K. Song, "Novel mimo detection with improved complexity for near-ml detection in mimo-ofdm systems," *ieee access*, vol. 7, pp. 60 389–60 398, 2019.

[37] Z. Zhou, S. Jere, L. Zheng, and L. Liu, "Learning for integer-constrained optimization through neural networks with limited training," in *NeurIPS Wkshps on Learn. Meets Combinatorial Algorithms*, Dec. 2020.

[38] M. R. Azghadi, S. Al-Sarawi, N. Iannella, and D. Abbott, "Efficient design of triplet based spike-timing dependent plasticity," in *The 2012 International joint conference on neural networks (IJCNN)*. IEEE, 2012, pp. 1–7.

[39] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," *Journal of Neuroscience*, vol. 2, no. 1, pp. 32–48, 1982.

[40] C.-C. Chang, P.-C. Chen, B. Hudec, P.-T. Liu, and T.-H. Hou, "Interchangeable hebbian and anti-hebbian stdp applied to supervised learning in spiking neural network," in *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2018, pp. 15–5.

[41] B. Guo, Y. Cai, Y. Pan, Z. Zhang, Y. Fang, and R. Huang, "Associative learning based on symmetric spike time dependent plasticity," in *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. IEEE, 2014, pp. 1–3.

[42] Y. Jin and P. Li, "Calcium-modulated supervised spike-timing-dependent plasticity for readout training and sparsification of the liquid state machine," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2007–2014.

[43] 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.

[44] M. Mozafari, S. R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, and M. Ganjtabesh, "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.

[45] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[46] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[47] G. K. Chen, R. Kumar, H. E. Sumbul, P. C. Knag, and R. K. Krishnamurthy, "A 4096-neuron 1m-synapse 3.8-pj/sop spiking neural network with on-chip stdp learning and sparse weights in 10-nm finfet cmos," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 992–1002, 2018.

[48] F. L. M. Huayaney, S. Nease, and E. Chicca, "Learning in silicon beyond stdp: a neuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2189–2199, 2016.

[49] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, "Are snns really more energy-efficient than anns? an in-depth hardware-aware study," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.

[50] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, *WINNER II Channel Models*. John Wiley Sons, Ltd, 2009, ch. 3, pp. 39–92. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470748077.ch3

[51] C. Lin, M. F. Azmine, and Y. Yi, "Accelerating next-g wireless communications with fpga-based ai accelerators," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–8.