Efficient Deep Spiking Multi-Layer Perceptrons with Multiplication-Free Inference

Boyan Li, Luziwei Leng, Ran Cheng, Shuaijie Shen, Kaixuan Zhang, Jianguo Zhang, and Jianxing Liao

***Abstract*—Advancements in adapting deep convolution architectures for Spiking Neural Networks (SNNs) have significantly enhanced image classification performance and reduced computational burdens. However, the inability of Multiplication-Free Inference (MFI) to harmonize with attention and transformer mechanisms, which are critical to superior performance on high-resolution vision tasks, imposes limitations on these gains. To address this, our research explores a new pathway, drawing inspiration from the progress made in Multi-Layer Perceptrons (MLPs). We propose an innovative spiking MLP architecture that uses batch normalization to retain MFI compatibility and introduces a spiking patch encoding layer to reinforce local feature extraction capabilities. As a result, we establish an efficient multi-stage spiking MLP network that effectively blends global receptive fields with local feature extraction for comprehensive spike-based computation. Without relying on pre-training or sophisticated SNN training techniques, our network secures a top-1 accuracy of 66.39% on the ImageNet-1K dataset, surpassing the directly trained spiking ResNet-34 by 2.67%. Furthermore, we curtail computational costs, model capacity, and simulation steps. An expanded version of our network challenges the performance of the spiking VGG-16 network with a 71.64% top-1 accuracy, all while operating with a model capacity 2.1 times smaller. Our findings accentuate the potential of our deep SNN architecture in seamlessly integrating global and local learning abilities. Interestingly, the trained receptive field in our network mirrors the activity patterns of cortical cells.**

***Index Terms*—Spiking Neural Network, Multi-layer Perceptron, Image Classification.**

I. Introduction

OVER the years, Convolutional Neural Networks (CNNs) have garnered considerable success within the field of computer vision. However, the advent and subsequent accomplishments of Transformer [1] in natural language processing have led to the emergence of the Vision Transformer (ViT) [2] as a promising contender. The ViT model replaces the convolution operation in CNNs with the self-attention operation from the Transformer, enabling it to model visual relationships across different spatial locations of an image. ViT and its subsequent developments [3], [4] have showcased performance on par with, or even superior to, CNNs. Unlike CNNs, which necessitate intricate convolution kernel designs, ViTs utilize several standard Transformer blocks, thus minimizing handcrafted manipulations and reducing inductive biases.

Boyan Li, Ran Cheng, Shuaijie Shen, Kaiwei Zhang and Jianguo Zhang are with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. (e-mail: ranchengcn@gmail.com) *(Boyan Li and Luziwei Leng contributed equally to this work.) (Corresponding author: Ran Cheng.)*

Luziwei Leng and Jianxing Liao are with the Advanced Computing and Storage Lab, Huawei Technologies Co., Ltd., Shenzhen 518055, China. (e-mail: {lengluziwei, liaojianxing}@huawei.com)

More recently, the MLP-Mixer [5] research proposed a simpler approach, leveraging Multilayer Perceptrons (MLP) to further mitigate induction bias. The fundamental block of MLP-Mixer consists of two components: the channel-mixing MLP and the token-mixing MLP. The channel-mixing module performs MLP calculations on the channel dimension of the feature map, facilitating information exchange across different channels. Simultaneously, the token-mixing module conducts MLP calculations across varying positions of the feature map, enabling communication over the spatial dimension. By harmonizing these two modules, the MLP-Mixer can efficiently extract and integrate features from both dimensions, enhancing the model's overall performance. This methodology presents a more straightforward route to reduce induction bias in comparison with previous models.

From a communication perspective, Spiking Neural Networks (SNNs) exhibit a finely-tuned mechanism for managing the exchange of information. Rooted in computational neuroscience, SNNs have been extensively utilized for the modeling of brain dynamics and functions [6], [7], [8], [9], [10]. Recently, the successes and challenges experienced by deep Artificial Neural Networks (ANNs) in tackling machine learning problems have fostered a growing curiosity towards SNNs. Researchers are exploring them as potential alternatives and are keen to leverage their bio-inspired properties to address similar issues.

The competence of SNNs has been convincingly demonstrated, with their performance proving to be on par with their artificial counterparts [11], [12], [13], [14], [15], [16], [17], [18], [19]. This has been achieved through the astute application of adaptive learning algorithms [20], [21], [22], [23] and the integration of efficient architectures borrowed from ANNs. These architectures span from the Boltzmann machines [24], prominent during the nascent stages of deep learning, to the currently dominant CNNs [25]. Furthermore, to identify task-specific cells or network structures for SNNs, recent research has employed Neural Architecture Search (NAS) [26], [27]. This approach not only strengthens the performance of SNNs but also broadens their potential application scope within the realm of machine learning.

While these networks have achieved significant reductions in spike occurrence and established new benchmarks in image classification, their underlying architectures still closely resemble deep CNNs. Recently, ANNs equipped with visual attention and transformer mechanisms [2], [28] have outperformed pure CNNs by learning global image dependencies. However, these mechanisms typically rely on matrix multiplication and softmax functions, presenting a contradiction to the

Multiplication-Free Inference (MFI) principle that characterizes SNNs [29], [30]. Recent research, such as the SNN-MLP study [31], offers a hybrid model that blends MLP and SNN neurons, effectively enhancing the MLP model's performance. Nonetheless, this model does not conform to a fully spiking architecture, suggesting that it does not constitute a strict SNN in the traditional sense.

In this study, we propose a novel approach to implementing MLPs to be inherently compatible with the MFI principle in a spike-based format. This partially comes on the heels of recent findings illustrating the equal efficiency of MLPs and transformers [5]. However, challenges persist as the original MLP-Mixer architecture for ANNs involves real-valued matrix multiplication, thus violating the MFI principle. To overcome these challenges, our contribution is twofold. First, we design a spiking MLP-Mixer architecture by utilizing MFI-friendly batch normalization (BN) combined with lightweight axial sampling in the token block. This architecture enables us to create a multi-stage spiking-MLP network, facilitating full spike-based computation. Second, we propose a spiking patch encoding module, anchored on a directed acyclic graph structure. This module replaces the original patch partition for downsampling, thereby enhancing local feature extraction capabilities of the MLP network. Moreover, our study underscores the critical role of skip connection configuration in achieving an optimal spiking MLP-Mixer design. Our efforts yield the following primary outcomes:

- We successfully engineer an efficient spiking MLP-Mixer using MFI-friendly BN and lightweight axial sampling in the token block, further emphasizing the importance of optimal skip connection configuration.
- By proposing a spiking patch encoding module, we enrich local feature extraction and enable downsampling, allowing for a multi-stage spiking-MLP network fully operating on spike-based computation.
- Our network delivers a 66.39% top-1 accuracy on the ImageNet-1K classification, marking a 2.67% improvement over the current state-of-the-art deep spiking ResNet-34 network. Moreover, the network operates with a similar model capacity, but with 2/3 of its simulation steps and significantly reduced computation cost. A larger variant of our network achieves a 71.64% top-1 accuracy, representing a 7.92% improvement and rivalling the spiking VGG-16 network. All of this is achieved without resorting to advanced training techniques.
- When fine-tuned on CIFAR10, CIFAR100, and CIFAR10-DVS datasets, our pre-trained networks on ImageNet set new benchmarks for SNNs, registering accuracies of 96.08%, 80.57%, and 81.12% respectively. This showcases the broad applicability of our architectural design as pre-trained models.

The rest of this paper is organized as follows: Section II delves into the related works MLPs and SNNs, two distinct deep learning approaches; Section III first provides an overview of our network architecture, followed by a detailed discussion on the vital Spiking MLP Block module, including the LIF neurons utilized therein; Section IV describes the tests conducted on three distinct datasets (ImageNet-1K, CIFAR10/100, and CIFAR10-DVS) for classification tasks, and presents our findings, including results from ablation, network sparsity, and neuronal weight visualization experiments; finally, Section V encapsulates the results and explores the implications of our work.

II. Related Work

A. Multi-Layer Perceptrons in Deep Learning

The recent advancements in attention and transformer mechanisms, particularly in speech [1] and vision tasks [2], have stimulated further investigations into similar mechanisms but in forms more biologically plausible. Studies outlined in [32], [33] demonstrated that the attention mechanism of transformers is equivalent to the update rule of modern Hopfield networks with continuous states. The high storage capacity of these networks in large-scale multiple instance learning was also highlighted. However, akin to the original transformer, these networks involve matrix multiplication and softmax function, both of which are incompatible with spike-based computation.

In the quest for an alternative architecture, the groundbreaking work of the MLP-Mixer [5] offered a model relying solely on Multi-Layer Perceptrons (MLPs), eschewing both convolution and self-attention layers. The MLP-Mixer is comprised of two unique parts - the channel-mixing MLP and the token-mixing MLP. The channel mixing module primarily focuses on conducting MLP computations within the channel dimension of the feature map, leading to an exchange of information among various channels and thereby fostering the extraction of effective features. Conversely, the token mixing module carries out MLP computations among different feature map positions. This enhances communication across spatial dimensions, fostering integration of diverse features from various map locations. Together, these modules equip the MLP-Mixer model with a robust proficiency for feature extraction and integration from both dimensions.

The MLP-Mixer model offers a simple yet powerful method to mitigate induction bias, surpassing previous models. This straightforward approach has proven highly successful in boosting model performance and has drawn considerable attention from the machine learning community.

Freed from the inductive biases of local connectivity and self-attention, the token-mixing MLP features enhanced flexibility and superior fitting capability [34], [35]. However, it also displays a higher susceptibility to overfitting and typically relies on pre-trained models with large-scale datasets to achieve competitive performances. These performances are on par with image classification benchmarks such as Vision Transformer (ViT) and Convolutional Neural Networks (CNNs). To overcome the issues of over-parametrization and overfitting associated with MLP-Mixer, a recent work proposed the SparseMLP [36]. This approach adopts a multi-stage pyramid network structure and applies axial sampling, as opposed to full sampling, in the token mixing MLP. This concept has also been echoed in other MLP variants [37], [38], [39].

B. Spiking Neural Networks in Deep Learning

Spiking Neural Networks (SNNs) have established their prowess in effective information processing, primarily due to their bio-inspired mechanisms. Deriving their roots from computational neuroscience, SNNs are frequently employed to model brain dynamics and functions [6], [7], [8], [9], [10]. As machine learning evolves through its successes and challenges, SNNs have emerged as promising alternatives with researchers exploring their biological properties for functional advantages in similar applications.

Although SNNs were not originally designed for gradient-based supervised learning, their bio-inspired architecture and capability to process spiking neural activity have led to a surge in popularity. Traditional training methods for SNNs, such as Spike Timing-Dependent Plasticity (STDP) [40], while effective, pose limitations in incorporating global information. This shortfall can slow convergence speed and impede applicability to large models. Further, STDP can result in unpredictable neural behavior, complicating the understanding and optimization of network performance.

Consequently, there has been a burgeoning interest in alternative SNN training techniques such as Backpropagation Through Time (BPTT) and surrogate gradients, which promise more efficient and effective learning for large-scale neural networks. Wu *et al.* advocated explicitly iterative LIF neuron training to enhance speed and accuracy [13], while Zheng *et al.* proposed a threshold-dependent batch normalization to streamline the training process [41]. Despite the advantages of gradient-based training, such as the production of efficient SNNs requiring minimal time steps, SNNs lag behind CNNs trained similarly, in terms of accuracy. This discrepancy stems from the complex spiking dynamics of SNNs, creating unique challenges in accurately modeling the underlying computations and optimizing network performance.

A different approach to creating SNN models involves transforming pre-trained ANN/CNN models into spiking neural networks. This strategy retains the original models' accuracy by first training non-spiking ANNs/CNNs using standard methods, followed by a conversion into spiking neural networks [42]. Despite the ease of this conversion process, it poses significant challenges, primarily requiring larger time steps to compensate for the reduced accuracy resulting from the transition from full-precision to binary output. Consequently, recent research has focused on integrating the conversion and training processes, exemplified by novel methodologies such as progressive conversion [43] and conversion during initialization [44].

Recently, the exploration of Neural Architecture Search (NAS) has led to the discovery of task-specific cells or network structures applicable to SNNs [26], [27]. These networks have achieved spike reduction and set new benchmarks in image classification. While these SNNs resemble deep CNNs, significant differences have emerged, such as alterations to traditional convolutional layers. In contrast, artificial neural networks that use visual attention and transformer mechanisms have recently outperformed pure CNNs in capturing global image dependencies [2], [28]. However, these models typically require heavy matrix multiplication and normalization operations, which contravene the Multiplication-Free Inference (MFI) principle integral to SNNs [29], [30]. Notably, these operations could lead to increased power consumption on neuromorphic hardware, which often possesses limited computational resources [45], [46], [47], [48], [49].

C. Discussion

The structural simplicity and efficacy of MLPs hint at a promising network paradigm. However, previous implementations of MLPs with non-spiking full-precision neurons fail to fulfill the MFI principle and eschew spike-based communication, rendering the network non-identical to typical SNNs [31]. Therefore, a pressing challenge lies in designing an MLP-based architecture compatible with spike-based computation and aligned with the MFI principle.

One promising direction is to combine the advantages of MLPs and SNNs, a strategy that could bridge the performance gap between spike-based models and state-of-the-art deep learning models. It would also potentially unlock the advantages of spiking computation, such as low energy usage and robustness to noise. However, to realize this promise, researchers must overcome several hurdles, including addressing the challenge of matrix multiplication and softmax operations, as these operations are incompatible with the spike-based computation and the MFI principle inherent to SNNs.

Overall, the emergence and development of SNNs have shown promising potential for efficient and biologically inspired computation. As the research progresses, SNNs could pave the way for a new generation of neural networks that bridge the gap between artificial and biological systems, ultimately advancing our understanding of both machine learning and neurobiology.

III. Proposed Approach

This section introduces our novel approach to addressing the challenges associated with the implementation of the MLP-Mixer in spiking neural networks. We propose a new network architecture, an optimized version of the MLP-Mixer that leverages the strengths of SNNs, and a unique implementation of the Leaky Integrate-and-Fire (LIF) neuron model. Key elements include a multi-stage pyramid network structure, the Spiking Patch Encoding (SPE) module, and the Spiking MLP-Mixer. We then elaborate on the structure of the spiking token block and the spiking channel block respectively, and discuss the implementation of the LIF spiking neuron model within the network.

A. Network Architecture

The architecture used by MLP-Mixer [5] is characterized by an *isotropic* design. This design holds the input and output resolutions constant across different layers. This type of structure tends to create a profuse number of parameters, leading to overfitting when training on medium-scale datasets, such as ImageNet-1K. To overcome this issue, we utilize a multi-stage pyramid network architecture as suggested by

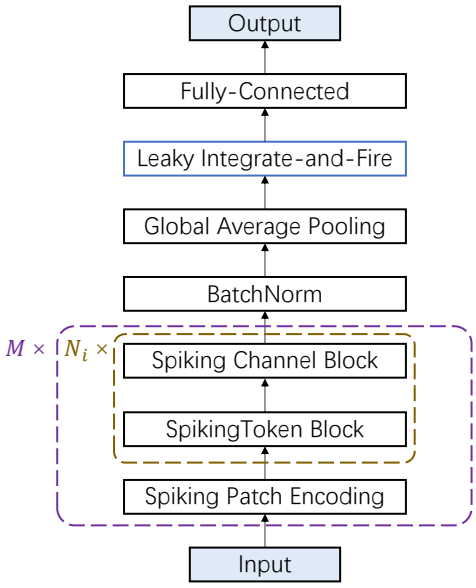


Fig. 1: The overall network architecture. The multi-stage network is downsampled with a spiking patch encoding (SPE) module at each stage. Within each stage, the SPE is followed by a sequence of spiking MLP-Mixers with identical architecture, each containing a spiking token block with axial sampling and a spiking channel block with full sampling.

[28], [50], [36]. Fig. 1 provides a visual representation of our network architecture.

Specifically, our approach starts with an RGB input image of size 3×H×W. The original MLP-Mixer breaks this image down into non-overlapping patches of size $p \times p$, subsequently projecting the new channel dimension of $3 \times p^2$ to a hidden dimension of C_1. This process parallels a 2D convolution operation with a kernel and stride size of p, and an output dimension of C_1. Although the MLP-Mixer benefits from a global receptive field, it lacks the necessary inductive bias for learning local features. To address this deficiency, we develop a spiking patch encoding (SPE) module to replace the original patch partitioning approach. We construct the SPE with a spike-based directed acyclic graph, following the design outlined in a recent work [18], as illustrated in Figure 3. Since this structure draws inspiration from DARTS [51], we term it as a spiking cell, which comprises three nodes. Each node receives the same input from the previous stage, which is processed through a convolution operation followed by Batch Normalization (BN). Node pairs (1,2) and (2,3) are interconnected through a convolution operation followed by BN, whereas node pair (1,3) is connected via an identity connection. Within each node, multiple operations are added after BN, with the subsequent spiking activation serving as the output of the node. The cell's output is the concatenated spiking states of all nodes.

The output of the spiking cell is then fed into a sequential of spiking MLP-Mixers which finally output a spiking feature map of the same shape. In the following stages, this process is repeated with the input feature consecutively downsampled in spatial dimension and expanded in channel dimension, both with a ratio of 2. The output of the final stage is fed into a spiking classification head with a spiking MLP block consisting of a fully connected (FC) layer followed by BN and a spiking activation, with the FC layer performing full sampling on the flattened spatial dimension of the feature map. Finally, a linear classifier projects the output of spiking MLP block to a label layer. The loss function is a cross entropy function with the network output averaged over simulation time steps.

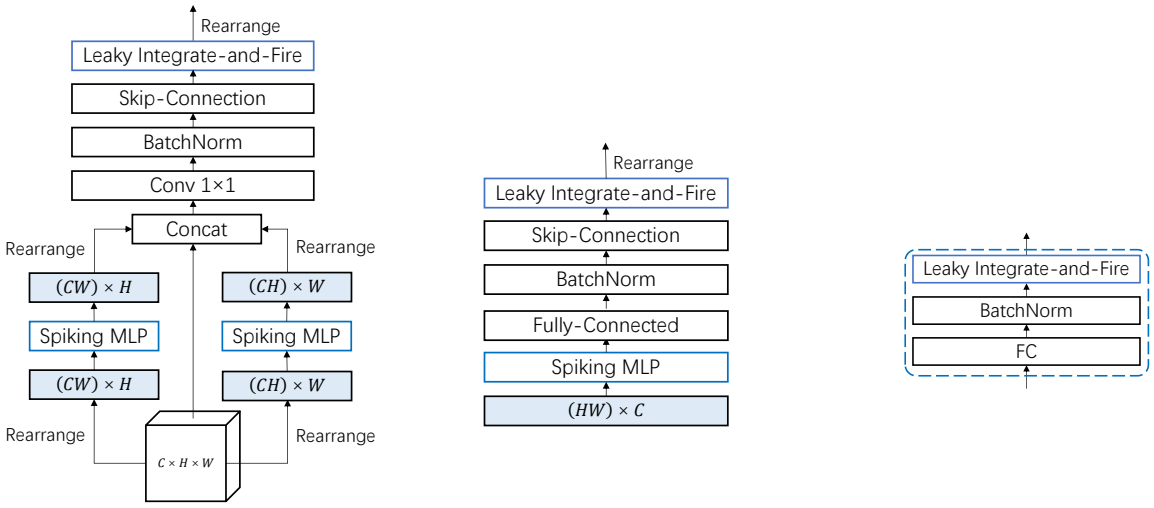
B. Spiking MLP-Mixer

A straightforward adoption of existing MLP block designs from artificial neural networks (ANNs) by substituting the real-valued activation function with a spiking activation function (SAF) results in a violation of the MFI principle of spike-based computation. In both the token and channel blocks of the original MLP-Mixer, the fully connected (FC) operation is carried out on the layer normalization (LN) state of the feature map, leading to real-valued matrix multiplication. To rectify this, we move the normalization process to after the FC operation and replace LN with batch normalization (BN). This adjustment enables the parameters of the latter to be integrated with the linear projection weights during inference, aligning with the MFI principle [52], [42].

The spiking MLP-Mixer comprises a token block and a channel block, each of which includes multiple FC layers with spiking activation, as depicted in Fig. 1(b) and Fig. 1(c).

1) Spiking Token Block: Given input patched image $\boldsymbol{I}$ of shape $C \times H \times W$, the token block of the original MLP-Mixer flattens its spatial dimension to form a 2D tensor. The process performs full sampling on the token dimension with weights shared across the channel dimension, thus creating a global receptive field that can be heavily parameterized for small patch sizes. Drawing inspiration from previous works [38], [37], [36], [39], we design a two-branch structure for the spiking token block. This structure separately encodes the feature representation along the horizontal and vertical spatial dimensions. It learns long-range dependencies along one direction while preserving positional information along the other.

Given input patched image $\boldsymbol{I}$ of shape $C \times H \times W$, the token block of the original MLP-Mixer flattens its spatial dimension to form a 2D tensor. The process performs full sampling on the token dimension with weights shared across the channel dimension, thus creating a global receptive field that can be heavily parameterized for small patch sizes. Drawing inspiration from previous works [38], [37], [36], [39], we design a two-branch structure for the spiking token block. This structure separately encodes the feature representation along the horizontal and vertical spatial dimensions. It learns long-range dependencies along one direction while preserving positional information along the other. As illustrated in Figure 1(c), each branch comprises an FC layer followed by a BN and an SAF. The two binary feature maps are then concatenated along the channel dimension, along with an identity connection from the input feature map, and projected back to C through an FC layer with BN and an SAF. Furthermore, we add a skip connection from the BN states of the SPE.



(a) Spiking Token Block **(b)** Spiking Channel Block **(c)** Spiking MLP Block

Fig. 2: Key blocks in the proposed network architecture.

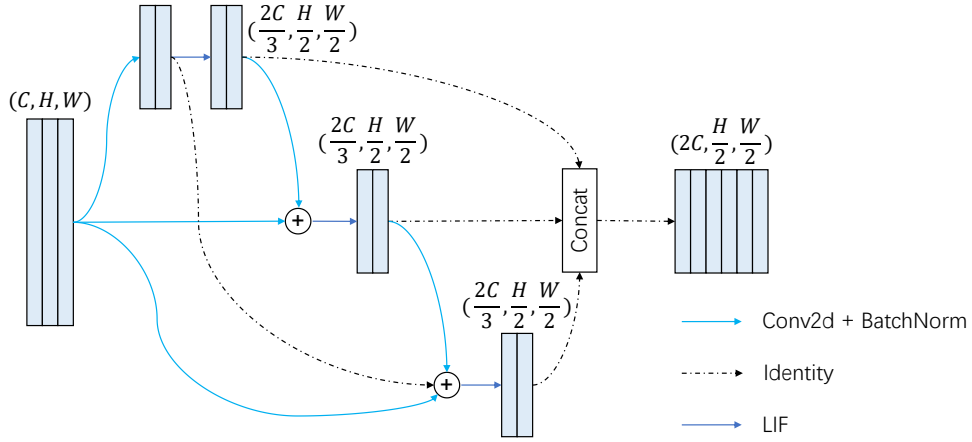


Fig. 3: Spiking patch encoding with a directed acyclic graph structure. The structure adheres to the MFI principle, with additions performed on BN states and multiplications performed between convolution weights and binary spikes.

In a single stage of the Spiking MLP-Mixer, assuming the input $\boldsymbol{X}$ of shape $C\times(HW)$ and the output $\boldsymbol{Y}$, we can express the Spiking Token Block as follows:

$$\boldsymbol{I} = f(\boldsymbol{X}), \tag{1}$$
$$\boldsymbol{U}_h = \mathrm{BN}(\boldsymbol{W}_h \boldsymbol{I}^h), \tag{2}$$
$$\boldsymbol{U}_w = \mathrm{BN}(\boldsymbol{W}_w \boldsymbol{I}^w), \tag{3}$$
$$\boldsymbol{Y} = \mathrm{BN}(\boldsymbol{W}_f \mathrm{Concat}[f(\tilde{\boldsymbol{U}}_h), f(\tilde{\boldsymbol{U}}_w), \boldsymbol{I}]) + \boldsymbol{X}, \tag{4}$$

where BN signifies batch normalization, f represents an SAF, $\boldsymbol{W}_h$ and $\boldsymbol{W}_w$ are the token FC weights acting on the height and width dimensions with shapes $H\times H$ and $W\times W$ respectively. $\boldsymbol{W}_f$ are the branch fusion weights with a shape of $C\times 3C$. Superscripts h and w denote the matrix reshape operation that retains the height, width, and channel dimensions, and ˜ denotes the reshape operation that flattens the spatial dimension with $\tilde{\boldsymbol{U}}_h$ and $\tilde{\boldsymbol{U}}_w$ of shape $C\times(HW)$.

2) Spiking Channel Block: The channel block, comprised of two spiking FC layers, takes the transposed output from the token block as its input. The initial layer expands the channel dimension by a predetermined ratio, α, and the subsequent layer restores this dimension to its original state. A skip connection is integrated between the BN state of the second layer and the aggregated BN state from the token block.

In a single stage of the Spiking MLP-Mixer, given the input $\boldsymbol{X}$ with shape $C\times(HW)$ and the output $\boldsymbol{Y}$, we can formulate the Spiking Channel Block as follows:

$$\boldsymbol{I} = f(\boldsymbol{X}), \tag{5}$$
$$\boldsymbol{V} = \boldsymbol{X}^c + \mathrm{BN}(\boldsymbol{W}_{c2} f(\mathrm{BN}(\boldsymbol{W}_{c1}\boldsymbol{I}^c))), \tag{6}$$
$$\boldsymbol{Y} = f(\tilde{\boldsymbol{V}}), \tag{7}$$

where $\boldsymbol{W}_{c1}$ and $\boldsymbol{W}_{c2}$ are the channel FC weights with shapes $\alpha C\times C$ and $C\times\alpha C$ respectively, and α is the channel expansion ratio.

To counteract the gradient vanishing issue commonly observed in deep SNNs, we incorporate skip connections between the final BN states of both the token and channel blocks, as well as the BN states from the SPE. Our ablation studies highlight the significance of this design in achieving optimal network efficiency.

It is noteworthy that all matrix multiplications involve a real-valued matrix and a binary matrix, which implies that the

spiking MLP-Mixer essentially operates without requiring any multiplications. In subsequent spiking MLP-Mixers, the input tensor is the output generated from the channel block of the preceding Mixer.

C. LIF Spiking Neuron

In our study, we employ the Leaky Integrate-and-Fire (LIF) neuron model, which is characterized by a hard threshold and decay input, as described by the following equation:

$$\boldsymbol{u}^{t,\mathrm{pre}} = \boldsymbol{u}^{t-1} + \frac{\boldsymbol{z}^t - \boldsymbol{u}^{t-1}}{\tau}, \tag{8}$$

$$\boldsymbol{y}^t = g(\boldsymbol{u}^{t,\mathrm{pre}}), \tag{9}$$

$$\boldsymbol{u}^t = (1-\boldsymbol{y}^t)\boldsymbol{u}^{t,\mathrm{pre}}, \tag{10}$$

where$t$ signifies the time step, τ represents the membrane time constant, and $\boldsymbol{u}$ is the membrane potential, denoted by a bold, italicized letter to represent a vector. $\boldsymbol{y}$ stands for the spike output and g is a threshold function. $\boldsymbol{z}^t = \boldsymbol{W}\boldsymbol{y}^{t,\mathrm{pre}}$ refers to the synaptic input, where $\boldsymbol{W}$ is the weight matrix, and $\boldsymbol{y}^{t,\mathrm{pre}}$ signifies the afferent spikes originating from pre-synaptic neurons.

When the membrane potential of a neuron surpasses a certain threshold V_{th}, the neuron fires a spike and conveys it to the post-synaptic neuron. This action is followed by a hard reset of the membrane potential. If the membrane potential does not exceed the threshold, the neuron does not transmit any signals, as delineated by:

$$y_i^t = \begin{cases} 1 & \text{if } u_i^{t,\mathrm{pre}} \geq V_{th} \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

Previous works have shown that training thresholds or membrane dynamics could potentially improve network performances [30], [53], [54]. Since this work focuses more on architectures, we set $\tau = 2$ and $V_{th} = 1$ during experiments. Given a loss L and using the chain rule, the weight update of a Spiking Neural Network (SNN) can be expressed as follows:

$$\frac{\partial L}{\partial \boldsymbol{W}} = \sum_t \frac{\partial L}{\partial \boldsymbol{y}^t}\frac{\partial \boldsymbol{y}^t}{\partial \boldsymbol{u}^{t,\mathrm{pre}}}\frac{\partial \boldsymbol{u}^{t,\mathrm{pre}}}{\partial \boldsymbol{z}^t}\frac{\partial \boldsymbol{z}^t}{\partial \boldsymbol{W}}, \tag{12}$$

where $\frac{\partial \boldsymbol{y}^t}{\partial \boldsymbol{u}^{t,\mathrm{pre}}}$ is the gradient of the firing function, which equals zero at all points except at the threshold. The surrogate gradient method leverages continuous functions to estimate the gradients. Various continuous functions have been used in previous studies, such as rectangular [41], triangular [55], and exponential curves [56], [57], among others. In our experiments, we chose to use the sigmoid function.

IV. Experiments

In order to thoroughly evaluate the performance and capabilities of our model, a variety of image classification benchmarks are utilized. These include the ImageNet-1K [58], CIFAR10/100 [59], and CIFAR10-DVS [60] datasets. The SpikingJelly [61] toolkit, complemented by its cupy backend, enables rapid simulation of Leaky Integrate-and-Fire (LIF) neurons. The Spiking Pooling Equilibrium (SPE) module incorporated in our architecture utilizes 3×3 convolution operations. The stride for input-to-node connections is set to 2, while node-to-node connections have a stride of 1. To avoid statistical bias, the batch mean and variance are synchronized across each device, following the approach suggested by [15].

The following sections will present results from the aforementioned datasets, discuss an ablation study, assess the network sparsity and computational cost, and finally visualize our findings.

A. Results on ImageNet-1K

The ImageNet-1K dataset is comprised of a training set, validation set, and a test set, with 1.28M, 50K, and 100K 224×224 images, respectively. Our training settings are predominantly derived from [36], encompassing data augmentation. We adopt a cosine decay learning rate strategy, initializing the learning rate at 0.1 and gradually lowering it to zero over 100 epochs. Stochastic Gradient Descent (SGD) is utilized as the optimizer with a momentum of 0.9.

In the first stage of the spiking MLP network, we use the original patch partitioning approach with a patch size of 4, while employing the SPE module for downsampling in subsequent stages. Three variants of the spiking MLP network are developed, specifically spiking MLP-SPE-T/S/B. The respective architectures are: spiking MLP-SPE-T: $C_1 = 78$, number of layers in each stage = $\{2; 8; 14; 2\}$; spiking MLP-SPE-S: $C_1 = 96$, number of layers in each stage = $\{2; 8; 14; 2\}$; spiking MLP-SPE-B: $C_1 = 108$, number of layers in each stage = $\{2; 10; 24; 2\}$.

The expansion ratio of the channel FC layer is set to $\alpha = 3$. To study the influence of the SPE module, we also train an MLP-S model with the original patch partition approach while other parts of the network remaining the same. The simulation step of SNN is set to $T = 4$. To better compare with other SNNs with larger time steps, we use time inheritance training (TIT) [17] and obtain the $T = 6$ result for MLP-SPE-T, where we initialize the network with the pre-trained MLP-SPE-T ($T = 4$) model and fine-tune for 50 epochs with $T = 6$ using a cosine learning rate decaying from 0.1 to 0.

Our model is compared to other state-of-the-art SNNs, including those using direct training and Artificial Neural Network (ANN)-to-SNN conversion methods. The results are presented in Table I, where we report the top-1 accuracy along with the number of model parameters and simulation steps.

It can be observed that ANN-SNN conversion methods can indeed achieve high accuracy, but they require exceedingly lengthy simulation steps. Hybrid training methods manage to reduce the simulation steps but remain considerably longer than methods using direct training. Among directly trained methods, and considering similar network capacity, MLP-SPE-T significantly outperforms STBP-tdBN ResNet-34 by 2.67%, all the while requiring fewer simulation time steps. Post-TIT, the model achieves an accuracy of 69.09% with $T = 6$, exceeding ResNet-34-large and VGG-16 models by 2.04% and 0.09% respectively, while maintaining a model capacity more than 3 and 5.5 times smaller. These results underscore the superior efficiency of our architecture.

The enhancement observed in MLP-SPE-S over MLP-S confirms that the SPE module can substantially improve

TABLE I: Results on ImageNet-1K. T denotes simulation length.

Method	Architecture	Model Size	T	Accuracy[%]
ANN-SNN [62]	ResNet-34	22M	768	71.6
ANN-SNN [63]	VGG-16	138M	2500	69.96
Hybrid training [44]	ResNet-34	22M	250	61.48
Hybrid training [64]	VGG-16	138M	250	65.19
STBP-tdBN [41]	ResNet-34	22M	6	63.72
TET [17]	ResNet-34	22M	6	64.79
STBP-tdBN [41]	ResNet-34-large	86M	6	67.05
Diet-SNN [30]	VGG-16	138M	5	69.00
Spiking MLP (our model)	MLP-SPE-T	**25M**	**4**	**66.39**
Spiking MLP (our model)	MLP-SPE-T	**25M**	**6**	**69.09**
Spiking MLP (our model)	MLP-S	**34M**	**4**	**63.25**
Spiking MLP (our model)	MLP-SPE-S	**38M**	**4**	**68.84**
Spiking MLP (our model)	MLP-SPE-B	**66M**	**6**	**71.64**

network performance. This result attests to the efficacy of amalgamating global receptive fields with local feature extraction. However, it is important to note that our method is more accurately compared with the STBP-tdBN method since both use the same conventional temporal averaged loss function. The Temporal Error Transport (TET) method, which employs a temporal moment-wise loss function, has been demonstrated to outperform the temporal averaged method [17].

B. Results on CIFAR

Both the CIFAR10 and CIFAR100 datasets consist of 50K training images and 10K testing images, each of size 32×32 pixels. For data augmentation, we implement random resized cropping and random horizontal flipping, in line with other studies. We follow a pre-training and fine-tuning paradigm, akin to other transformer and MLP works [2], [5].

The pre-training phase employs the spiking MLP-SPE-T network trained on the ImageNet-1K dataset. We then reset the final classification output layer of the network and fine-tune it on the resized 224×224 CIFAR10/100 dataset. The fine-tuning phase uses a cosine decay learning rate strategy, starting from an initial value of 0.1 and gradually reducing to zero over 100 epochs. The optimizer we utilize is SGD with a momentum of 0.9.

Our results are benchmarked against other state-of-the-art SNNs, as depicted in Table II. Despite a larger network size, our model sets new records on both datasets, outperforming existing SNNs significantly. While comparing our results with directly trained SNNs might seem somewhat disproportionate, the outcomes convincingly illustrate the efficacy of the spiking MLP network as a pre-trained model applicable to other datasets.

C. Results on CIFAR10-DVS

Neuromorphic datasets generally possess higher noise levels in comparison to static datasets, which heightens the risk of overfitting in well-optimized SNNs. Among all mainstream neuromorphic datasets, CIFAR10-DVS is recognized as one of the most challenging, presenting approximately 0.9k training samples for each label. Recent literature suggests an inclination towards complex architectures to handle this dataset effectively, albeit this strategy carries an inherent risk of overfitting without correspondingly enhancing the model's accuracy. Therefore, a pressing need exists for novel methodologies that can abate the influence of noise and tackle the difficulties inherent in working with CIFAR10-DVS.

As with the CIFAR10/100 datasets, we employ the spiking MLP-SPE-T network, pre-trained on the ImageNet-1K dataset, as the pre-trained model for the CIFAR10-DVS dataset. We reset the final classification output layer of the network, and subsequently fine-tune it on the resized 224×224 CIFAR10-DVS dataset. Our fine-tuning strategy employs a cosine decay learning rate, starting with an initial value of 0.1 and progressively reducing to zero over 100 epochs. SGD, with a momentum of 0.9, is utilized as the optimizer.

We compare our results with those achieved by other state-of-the-art SNNs, as shown in Table III. Our experiments indicate that the spiking MLP-SPE-T outperforms the spiking ResNet-19 and even exceeds the performance of SNNs trained using Dspike.

D. Ablation Study

In this section we further study the influence of different configurations of skip connections on the performance of the network. As shown in Figure 4, we distinguish different skip connections, i.e patch block to token block (PT), patch block to channel block (PC), token block to channel block (TC) and channel block to the next token block (CT), with corresponding indices and colors. The skip connection from the patch block to the first token block is set as default. To shorten the simulation time, we set the initial channel number to 60 and directly train networks of different skip connection configurations from scratch on the CIFAR10 dataset for 100 epochs. The results are collected in Table IV. Networks without skip connections from the initial patch block to sequential token and channel blocks perform much worse than the others. This could be due to long range skip connections alleviate the gradient vanish problem of deep SNNs during training. Our spiking MLP-mixer adopts the optimal skip connection configuration of (PT, PC, TC).

TABLE II: Results on CIFAR. T denotes simulation length.

Dataset	Method	Architecture	T	Accuracy[%]
CIFAR10	Diet-SNN [30]	ResNet-20	10	92.54
	STBP-tdBN [41]	ResNet-19	6	93.16
	STBP-tdBN [41]	ResNet-19	4	92.92
	TET [17]	ResNet-19	4	94.44 ± 0.08
	Spiking MLP (our model)	MLP-SPE-T	**4**	**96.08**
CIFAR100	Diet-SNN [30]	ResNet-20	5	64.07
	STBP-tdBN [41]	ResNet-19	6	71.12 ± 0.57
	STBP-tdBN [41]	ResNet-19	4	70.86 ± 0.22
	TET [17]	ResNet-19	6	74.72 ± 0.28
	TET [17]	ResNet-19	4	74.47 ± 0.15
	Spiking MLP (our model)	MLP-SPE-T	**4**	**80.57**

TABLE III: Comparison on CIFAR10-DVS. T denotes simulation length.

Method	Architecture	T	Accuracy[%]
STBP-tdBN [41]	ResNet-19	10	67.80
TET [17]	VGG	10	83.17
Dspike [15]	ResNet-18	10	75.40
Spiking MLP (our model)	MLP-SPE-T	**10**	**81.12**

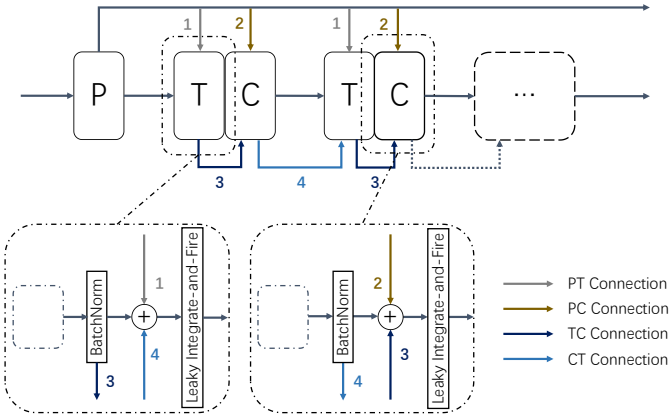


Fig. 4: Potential skip connections for the spiking MLP-Mixer.

TABLE IV: Network performance under different skip connection configurations on CIFAR10.

PT(1)	PC(2)	TC(3)	CT(4)	Accuracy[%]
✓	✓			80.53
✓	✓	✓		81.35
✓	✓	✓	✓	79.46
		✓	✓	10.53
			✓	10.00

E. Network Sparsity and Computational Cost

SNNs are recognized for their superior energy efficiency compared to dense-computing ANNs, a quality that primarily stems from the inherent event-driven and sparse computing characteristics of SNNs. Additionally, the MFI principle facilitates an entirely addition-based operation for SNNs, thus further diminishing their potential energy usage in contrast to ANNs that necessitate real-valued matrix multiplications. In this section, we assess the sparsity and computation cost of the spiking MLP-SPE-T model, contrasting it with the spiking ResNet-34.

Figure 5 portrays the average sparsity of each spiking MLP-Mixer on the ImageNet-1K test set. Our model demonstrates diverse sparse activity across different layers, yielding an average network sparsity of 0.14. Following the methodologies established by [15], [30], we compute the number of addition operations in the SNN using the formula sTA, where s represents the average sparsity (the total count of spikes divided by the total feature map size), T indicates the simulation time step, and A signifies the number of additions.

The findings are presented in Table V. The least amount of multiplication arises from the input layer, which receives floating-value images. To estimate the network's energy consumption, we refer to the study by [45] on the 45nm CMOS technology, a method also employed by previous works [15], [30]. According to this estimate, a single addition operation in an SNN costs 0.9pJ, whereas a multiply-accumulate (MAC) operation in ANNs consumes 4.6pJ. Notwithstanding their comparable model capacities, our network's computation cost is nearly half of that of the spiking ResNet-34.

TABLE V: The computation and estimated energy cost.

Model	#Param.	Accuracy[%]	#Add.	#Mult.	Energy
Spiking ResNet-34	22M	63.72	1.85G	118M	2.21 mJ
Spiking MLP-SPE-T	25M	66.39	1.18G	12M	**1.12** mJ

F. Visualization

Lastly, we provide a visualization of weights from token blocks across various stages of the spiking MLP-SPE-T after training on ImageNet-1K. These weights are depicted in Figure 6. Interestingly, the weights in the early stages appear to concentrate on local areas, while those in the later stages progressively sample more global areas. Without an inductive bias, the SNN naturally learns to form hierarchically arranged receptive fields. Further, the receptive fields of spiking neurons in the initial stages resemble the 'off-center on-surround'

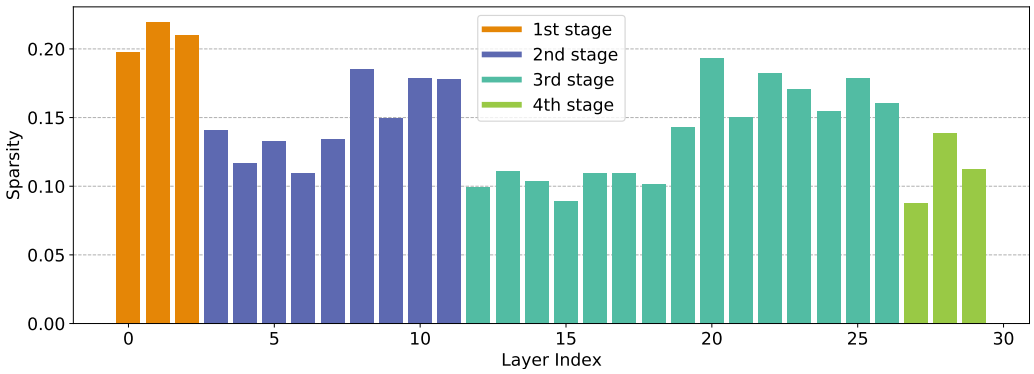


Fig. 5: Mean network sparsity of spiking MLP-SPE-T on the ImagNet-1K test set. We distinguish each stage with different colors.

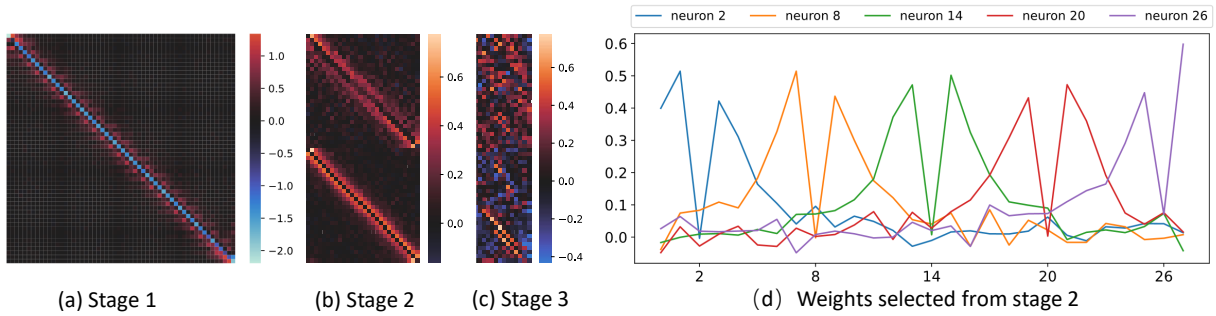


Fig. 6: Axial sampling weights from token blocks across different stages after training on ImageNet-1K. Figures (a-c) show randomly selected token weights plotted in a $H \times H$ or $W \times W$ 2D matrix. 1, 2, and 4 weight matrices from stage 1, 2, and 3 are displayed, respectively. The diagonal distribution of values in the early stages indicates local sampling. Figure (d) represents weights of several neurons from the same token block in stage 2. For the sake of simplicity, we only plot a few neurons here; however, it should be noted that this 'M' shaped weight kernel is a universal finding in the network's early stages.

receptive fields of cortex cells [65], characterized by a central inhibitory area encircled by an excitatory area. This 'M' shaped weight kernel is universally observed in the network's early stages. It would be intriguing to determine whether similar phenomena occur in ANNs.

V. Conclusion

This work presented the construction of MLP architectures utilizing spiking neurons, adhering to the MFI principle. The spiking MLP-Mixer, within our framework, employed batch normalization instead of layer normalization in both the token and channel block to ensure compatibility with MFI. Furthermore, the network's local feature learning capability was augmented with a spiking patch encoding layer, which notably enhanced the network's performance.

Leveraging these foundational building blocks, we investigated an optimal skip connection configuration and developed a proficient multi-stage spiking MLP network. This network combined global receptive field and local feature extraction. With comparable model capacity, our networks markedly surpassed state-of-the-art, mainstream deep spiking convolutional networks on the ImageNet-1K dataset. This is evident in terms of balancing accuracy, model capacity, and computation cost, thus demonstrating the efficacy of this alternative architecture for deep SNNs.

Our work underscores the importance of integrating global and local learning for optimal SNN architecture design. Interestingly, the trained receptive fields of our network bear a striking resemblance to those of cells in the cortex. Future comparative analysis with ANNs could potentially yield insightful results.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 558–567.

[4] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[5] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.

[6] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[7] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005.

[8] G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. Friston, "The dynamic brain: from spiking neurons to neural masses and cortical fields," *PLoS computational biology*, vol. 4, no. 8, p. e1000092, 2008.

[9] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[10] A. Korcsak-Gorzo, M. G. Müller, A. Baumbach, L. Leng, O. J. Breitwieser, S. J. van Albada, W. Senn, K. Meier, R. Legenstein, and M. A. Petrovici, "Cortical oscillations support sampling-based computations in spiking neural networks," *PLoS computational biology*, vol. 18, no. 3, p. e1009753, 2022.

[11] L. Leng, M. A. Petrovici, R. Martel, I. Bytschok, O. Breitwieser, J. Bill, J. Schemmel, and K. Meier, "Spiking neural networks as superior generative and discriminative models," *Cosyne Abstracts, Salt Lake City USA*, 2016.

[12] L. Leng, R. Martel, O. Breitwieser, I. Bytschok, W. Senn, J. Schemmel, K. Meier, and M. A. Petrovici, "Spiking neurons with short-term synaptic plasticity form superior generative networks," *Scientific reports*, vol. 8, no. 1, pp. 1–11, 2018.

[13] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1311–1318.

[14] L. Leng, "Solving machine learning problems with biological principles," Ph.D. dissertation, 2020.

[15] Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu, "Differentiable spike: Rethinking gradient-descent for training spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[16] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[17] S. Deng, Y. Li, S. Zhang, and S. Gu, "Temporal efficient training of spiking neural network via gradient re-weighting," *arXiv preprint arXiv:2202.11946*, 2022.

[18] K. Che, L. Leng, K. Zhang, J. Zhang, Q. Meng, J. Cheng, Q. Guo, and J. Liao, "Differentiable hierarchical and surrogate gradient search for spiking neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 975–24 990, 2022.

[19] R. Zhang, L. Leng, K. Che, H. Zhang, J. Cheng, Q. Guo, J. Liao, and R. Cheng, "Accurate and efficient event-based semantic segmentation using adaptive spiking encoder-decoder network," *arXiv preprint arXiv:2304.11857*, 2023.

[20] S. M. Bohte, J. N. Kok, and J. A. La Poutré, "Spikeprop: backpropagation for networks of spiking neurons." in *ESANN*, vol. 48. Bruges, 2000, pp. 419–424.

[21] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.

[22] 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.

[23] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature communications*, vol. 11, no. 1, pp. 1–15, 2020.

[24] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[26] B. Na, J. Mok, S. Park, D. Lee, H. Choe, and S. Yoon, "Autosnn: Towards energy-efficient spiking neural networks," *arXiv preprint arXiv:2201.12738*, 2022.

[27] Y. Kim, Y. Li, H. Park, Y. Venkatesha, and P. Panda, "Neural architecture search for spiking neural networks," *arXiv preprint arXiv:2201.10355*, 2022.

[28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[29] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[30] N. Rathi and K. Roy, "Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[31] W. Li, H. Chen, J. Guo, Z. Zhang, and Y. Wang, "Brain-inspired multilayer perceptron with spiking neurons," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 783–793.

[32] M. Widrich, B. Schäfl, M. Pavlović, H. Ramsauer, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter *et al.*, "Modern hopfield networks and attention for immune repertoire classification," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 832–18 845, 2020.

[33] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve *et al.*, "Hopfield networks is all you need," *arXiv preprint arXiv:2008.02217*, 2020.

[34] Y. Zhao, G. Wang, C. Tang, C. Luo, W. Zeng, and Z.-J. Zha, "A battle of network structures: An empirical study of cnn, transformer, and mlp," *arXiv preprint arXiv:2108.13002*, 2021.

[35] R. Liu, Y. Li, L. Tao, D. Liang, and H.-T. Zheng, "Are we ready for a new paradigm shift? a survey on visual deep mlp," *Patterns*, vol. 3, no. 7, p. 100520, 2022.

[36] C. Tang, Y. Zhao, G. Wang, C. Luo, W. Xie, and W. Zeng, "Sparse mlp for image recognition: is self-attention really necessary?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2344–2351.

[37] Q. Hou, Z. Jiang, L. Yuan, M.-M. Cheng, S. Yan, and J. Feng, "Vision permutator: A permutable mlp-like architecture for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[38] Y. Tatsunami and M. Taki, "Raftmlp: Do mlp-based models dream of winning over computer vision?" *arXiv e-prints*, pp. arXiv–2108, 2021.

[39] Z. Wang, W. Jiang, Y. M. Zhu, L. Yuan, Y. Song, and W. Liu, "Dynamixer: a vision mlp architecture with dynamic mixing," in *International Conference on Machine Learning*. PMLR, 2022, pp. 22 691–22 701.

[40] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS computational biology*, vol. 3, no. 2, p. e31, 2007.

[41] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 062–11 070.

[42] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.

[43] W. Severa, C. M. Vineyard, R. Dellana, S. J. Verzi, and J. B. Aimone, "Training deep neural networks for binary communication with the whetstone method," *Nature Machine Intelligence*, vol. 1, no. 2, pp. 86–94, 2019.

[44] N. Rathi, G. Srinivasan, P. Panda, and K. Roy, "Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation," *arXiv preprint arXiv:2005.01807*, 2020.

[45] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[46] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[47] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[48] 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.

[49] A. F. Kungl, S. Schmitt, J. Klähn, P. Müller, A. Baumbach, D. Dold, A. Kugele, E. Müller, C. Koke, M. Kleider *et al.*, "Accelerated physical emulation of bayesian inference in spiking neural networks," *Frontiers in neuroscience*, p. 1201, 2019.

[50] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9355–9366, 2021.

[51] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[53] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of

spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671.

[54] K. Zhang, K. Che, J. Zhang, J. Cheng, Z. Zhang, Q. Guo, and L. Leng, "Discrete time convolution for fast event-based stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8676–8686.

[55] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," *Advances in neural information processing systems*, vol. 31, 2018.

[56] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *arXiv preprint arXiv:1810.08646*, 2018.

[57] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[59] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[60] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "Cifar10-dvs: an event-stream dataset for object classification," *Frontiers in neuroscience*, vol. 11, p. 309, 2017.

[61] W. Fang, Y. Chen, J. Ding, D. Chen, Z. Yu, H. Zhou, Y. Tian, and other contributors, "Spikingjelly," https://github.com/fangwei123456/spikingjelly, 2020, accessed: YYYY-MM-DD.

[62] Y. Hu, H. Tang, and G. Pan, "Spiking deep residual networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[63] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[64] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in neuroscience*, p. 119, 2020.

[65] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.