Inherent Redundancy in Spiking Neural Networks

Man Yao[1,2,3*], Jiakui Hu[4,2*], Guangshe Zhao[1†], Yaoyuan Wang[5], Ziyang Zhang[5], Bo Xu[2], Guoqi Li[2†]
[1]School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, China
[2]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]Peng Cheng Laboratory, Shenzhen, China
[4]Peking University Health Science Center, Peking University, Beijing, China
[5]Advanced Computing and Storage Lab, Huawei Technologies Co. Ltd.
manyao@stu.xjtu.edu.cn, jkhu29@stu.pku.edu.cn, zhaogs@mail.xjtu.edu.cn, guoqi.li@ia.ac.cn

Abstract

Spiking Neural Networks (SNNs) are well known as a promising energy-efficient alternative to conventional artificial neural networks. Subject to the preconceived impression that SNNs are sparse firing, the analysis and optimization of inherent redundancy in SNNs have been largely overlooked, thus the potential advantages of spike-based neuromorphic computing in accuracy and energy efficiency are interfered. In this work, we pose and focus on three key questions regarding the inherent redundancy in SNNs. We argue that the redundancy is induced by the spatio-temporal invariance of SNNs, which enhances the efficiency of parameter utilization but also invites lots of noise spikes. Further, we analyze the effect of spatio-temporal invariance on the spatio-temporal dynamics and spike firing of SNNs. Then, motivated by these analyses, we propose an Advance Spatial Attention (ASA) module to harness SNNs' redundancy, which can adaptively optimize their membrane potential distribution by a pair of individual spatial attention sub-modules. In this way, noise spike features are accurately regulated. Experimental results demonstrate that the proposed method can significantly drop the spike firing with better performance than state-of-the-art SNN baselines. Our code is available in `https://github.com/BICLab/ASA-SNN`.

1. Introduction

By mimicking the spatio-temporal dynamics behaviors of biological neurons, Spiking Neural Networks (SNNs) pose a paradigm shift in information encoding and transmitting [36, 37, 25]. Spiking neurons only fire when the membrane potential is greater than the threshold (Figure 1**a**), in theory, these complex internal dynamics make the representation ability to spiking neurons more powerful than existing artificial neurons [31]. Moreover, spike-based binary communication (0/1 spike) enables SNNs to be *event-driven* when deployed on neuromorphic chips [3, 34, 49], i.e., performing cheap synaptic Accumulation (AC) and bypassing computations on zero inputs or activations [6, 4].

For a long time, when referring to spike-based neuromorphic computing, people naturally believe that its computation is sparse due to the event-driven feature. Subject to this preconceived impression, although it is generally agreed that sparse spike firing is the key to achieving high energy efficiency in neuromorphic computing, there is a lack of systematic and in-depth analysis of redundancy in SNNs. Existing explorations are limited to specific methods of dropping spike counts. For instance, several algorithms have been proposed to exploit spike-aware sparsity regularization and compression by adding a penalty function[5, 53, 55, 52, 33, 24], designing network structures with fewer spikes using neural architecture search techniques [32, 23], or developing data-dependent models to regulate spike firing based on the input data [48, 51]. Generally, employing these methods to reduce spikes incurs a loss of accuracy or significant additional computation.

In this work, we provide a novel perspective to understand the redundancy of SNNs by analyzing the relationship between *spike firing* and *spatio-temporal dynamics* of spiking neurons. This analysis could be extended by asking three key questions. (i) **Which** spikes are redundant? (ii) **Why** is there redundancy in SNNs? (iii) **How** to efficiently drop the redundant spikes?

To perfectly demonstrate redundancy in SNNs, we select event-based vision tasks to observe spike responses. Event-based cameras, such as the Dynamic Vision Sensor (DVS) [27], are a novel class of bio-inspired vision sensors that only encode the vision scene's brightness change information into a stream of events (spike with address information)

*These authors contribute equally to this work
†Corresponding author

for each pixel. As shown in Figure 1**b**, the red and green dots represent pixels that increase and decrease in brightness, respectively, and the gray areas without events indicate no change in brightness. However, although the information given in the input is human gait without background, some spike features extracted by the vanilla SNN focus on background information. As depicted in Figure 1**c**, the spiking neurons in the noise feature map fire a large number of spikes in the background region, which are redundant.

Unfortunately, noise features exist widely in both temporal and spatial dimensions, but exhibit some interesting regularities. We argue that the underlying reason for this phenomenon is due to a fundamental assumption of SNNs, known as *spatio-temporal invariance*[21], which enables sharing weights for every location across all timesteps. This assumption improves the parameter utilization efficiency while boosting the redundancy of SNNs. Specifically, by controlling the input time window of event streams, we can clearly observe the temporal and spatial changes of the spike features extracted by the SNN (see Figure 2). In the spatial dimension, there are many similar noise features, which can be referred to as ghost features [14, 15]. In the temporal dimension, although the information extracted by SNN changes at different timesteps, the spatial position of the noise spike feature is almost the same.

Recently, several works [12, 13] have investigated the information loss caused by SNNs when quantizing continuous membrane potential values into discrete spikes. Inspired by these works, we transformed our problem "the relationship between spike firing and spatio-temporal dynamics of spiking neurons" to investigate the relationship between membrane potential distribution and redundant spikes. Motivated by the observations that redundancy is highly correlated with spike feature patterns and neuron location, we present the Advanced Spatial Attention (ASA) module for SNNs, which can convert noise features into normal or null (without spike firing) features by shifting the membrane potential distribution. We conduct extensive experiments using a variety of network structures to verify the superiority of our method on five event-based datasets. Experimental results show that the ASA module can help SNN reduce spikes and improve task performance concurrently. For instance, on the DVS128 Gait-day dataset[41], at the cost of negligible additional parameters and computations, the proposed ASA module decreases the baseline model's spike counts by 78.9% and increases accuracy by +5.0%. We summarize our contributions as follows:

1) We provide the first systematic and in-depth analysis of the inherent redundancy in SNNs by asking and answering three key questions, which are crucial to the high energy efficiency of spike-based neuromorphic computing but have long been neglected.

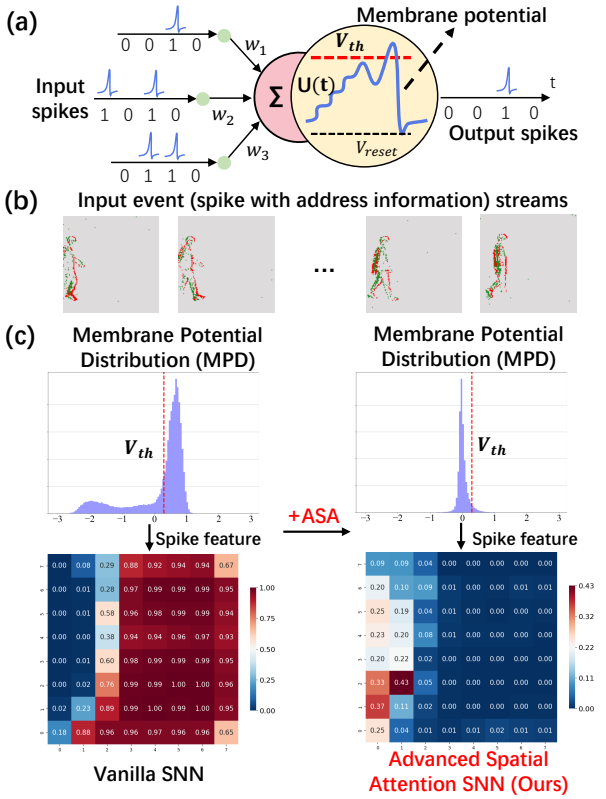


Figure 1: (a) Spatio-temporal dynamics of spiking neurons with binary spike input and output, synaptic weight w, membrane potential $U(t)$, threshold V_{th} and hard reset membrane potential V_{reset}. (b) An example of an event stream. (c) Examples of changes in the spike responses of vanilla SNN and ASA-SNN, in terms of Membrane Potential Distribution (MPD) and spike feature. Each pixel value on the spike feature represents the firing rate of a neuron. Noise spike feature fires lots of spikes while concentrating on insignificant background information (large area red). The ASA module can shift the spike pattern of SNNs to drop spike counts by optimizing the MPD.

2) For the first time, we relate the redundancy of SNNs to the distribution of membrane potential, and design a simple yet efficient advanced spatial attention to help SNN optimize the membrane potential distribution and thus reduce redundancy.

3) Extensive experimental results show that the proposed ASA module can improve SNNs' performance and significantly drop noise spikes concurrently. This inspires us that two of the most important nature of spike-based neuromorphic computing, bio-inspired spatio-temporal dynamics and event-driven sparse computing, can be naturally incorporated to achieve better performance with lower energy consumption.

2. Related work

Event-based vision and spike-based neuromorphic computing. Due to the unique advantages of high temporal resolution, high dynamic range, etc., DVS has broad application prospects in special visual scenarios, such as high-speed object tracking [56], low-latency interaction [1], etc. Event-based vision is one of the typical advantage application scenarios of SNNs, which can process event streams event-by-event to achieve minimum latency [10], and can be smoothly deployed on neuromorphic chips to realize ultra-low energy cost by spike-based event-driven sparse computing [37, 35, 36, 6]. As an example, a recent edge computing device called Speck[1] integrates an SNN-enabled asynchronous neuromorphic chip with a DVS camera [10]. Its peak power is mW level, and latency is ms level. In this work, we investigate the SNNs' inherent redundancy by using a variety of event-based datasets to further explore their enticing potential for accuracy and energy efficiency.

Attention in SNNs. Attention methods were included in deep learning with tremendous success and were motivated by the fact that humans can focus on salient vision information in complicated scenes easily and efficiently. A popular research direction is to present attention as an auxiliary module to boost the representation capacity of ANNs [19, 44, 47, 26, 11]. In line with this idea, Yao *et al*. [48] first suggested using an extra plug-and-play temporal-wise attention module for SNNs to bypass a few unnecessary input timesteps. Subsequently, a number of works were given to utilize multi-dimensional attention modules for SNNs, including temporal-wise, spatial-wise , or channel-wise simultaneously [30, 59, 54, 51, 50], where Yao *et al*. [51] highlighted that attention could aid SNNs in reducing spike firing while enhancing accuracy. However, to produce attention scores and refine membrane potentials, multi-dimensional attention inevitably adds a lot of extra computational burden to SNNs. In this work, we exclusively employ spatial attention, which is inspired by the investigation of the redundancy of SNNs.

Membrane Potential Distribution in SNNs. Rectifying MPD is crucial for SNN training because SNNs are more vulnerable to gradient vanishing or explosion since spikes are discontinuous and non-differentiable. Around this point, researchers have made many advances in SNN training, such as normalization techniques [57, 46], shortcut design [20, 7], extension with more learnable parameter [8, 38], distribution loss design [13, 12], etc. We here, in contrast to prior publications, concentrate on the connection between MPD and redundancy, a topic that is typically disregarded in the SNN community.

[1] https://www.synsense-neuromorphic.com/products/speck/

3. SNN Redundancy Analysis

3.1. SNN Fundamentals

The basic computational unit of a SNN is the spiking neuron, which is the abstract modeling of the dynamics mechanism of biological neuron [17]. The Leaky Integrate-and-Fire (LIF) model [31] is one of the most commonly used spiking neuron models since it establishes a good balance between the simplified mathematical form and the complex dynamics of biological neurons. We describe the LIF-SNN layer in its iterative representation version form [45]. First, the LIF layer will perform the following integration operations,

$$\boldsymbol{U}^{t,n} = \boldsymbol{H}^{t-1,n} + \boldsymbol{X}^{t,n}, \tag{1}$$

where $n \in \{1, \cdots, N\}$ and $t \in \{1, \cdots, T\}$ denote the layer and timestep, $\boldsymbol{U}^{t,n}$ means the membrane potential which is produced by coupling the spatial feature $\boldsymbol{X}^{t,n}$ and the temporal information $\boldsymbol{H}^{t-1,n}$, and $\boldsymbol{X}^{t,n}$ can be done by convolution operations,

$$\boldsymbol{X}^{t,n} = \mathrm{BN}\left(\mathrm{Conv}\left(\boldsymbol{W}^{n}, \boldsymbol{S}^{t,n-1}\right)\right), \tag{2}$$

where $\mathrm{BN}(\cdot)$ and $\mathrm{Conv}(\cdot)$ mean the batch normalization[22] and convolution operation respectively, $\boldsymbol{W}^{n}$ is the weight matrix, $\boldsymbol{S}^{t,n-1}(n \neq 1)$ is a spike tensor from the last layer that only contain 0 and 1, and $\boldsymbol{X}^{t,n} \in \mathbb{R}^{c_n \times h_n \times w_n}$. Then, the fire and leaky mechanism inside the spiking neurons are respectively executed as

$$\boldsymbol{S}^{t,n} = \mathrm{Hea}\left(\boldsymbol{U}^{t,n} - V_{th}\right), \tag{3}$$

and

$$\boldsymbol{H}^{t,n} = V_{reset}\boldsymbol{S}^{t,n} + \left(\beta \boldsymbol{U}^{t,n}\right) \otimes \left(\mathbf{1} - \boldsymbol{S}^{t,n}\right), \tag{4}$$

where V_{th} is the threshold to determine whether the output spike tensor $\boldsymbol{S}^{t,n}$ should be spike or stay as zero, $\mathrm{Hea}(\cdot)$ is a Heaviside step function that satisfies $\mathrm{Hea}\,(x) = 1$ when $x \geq 0$, otherwise $\mathrm{Hea}\,(x) = 0$, V_{reset} denotes the reset potential which is set after activating the output spike, and $\beta = e^{-\frac{dt}{\tau}} < 1$ reflects the decay factor, τ is the membrane time constant, and $\otimes$ denotes the element-wise multiplication. When the entries in $\boldsymbol{U}^{t,n}$ are greater than the threshold V_{th}, the spatial output of spike sequence $\boldsymbol{S}^{t,n}$ will be activated (Eq. 3). Meanwhile, the entries in $\boldsymbol{U}^{t,n}$ will be reset to V_{reset}, the temporal output $\boldsymbol{H}^{t,n}$ should be decided by $\boldsymbol{X}^{t,n}$ since $\mathbf{1} - \boldsymbol{S}^{t,n}$ must be 0. Otherwise, the decay of $\boldsymbol{U}^{t,n}$ will be used to transmit the $\boldsymbol{H}^{t,n}$, since the $\boldsymbol{S}^{t,n}$ is 0, which means there is no activated spike output (Eq. 4).

3.2. Redundancy Analysis

We first define various terms to appropriately represent redundancy in SNNs, as below.

Definition 1. *Spike Firing Rate (SFR)*: We input all the samples on the test set into the network and count the spike distribution. We define a Neuron's SFR (N-SFR) at the t-th timestep as the ratio of the number of samples generating spikes on this neuron to the number of all tested samples. Similarly, at the t-th timestep, we define the SFR of a Channel (C-SFR) or this Timestep (T-SFR) as the average of the SFR values of all neurons in this channel or the entire network at this timestep. We define the Network Average SFR (NASFR) as the average of T-SFR over all timesteps T.

Definition 2. *Spike features.* We input all the samples on the test set into the network and define the average output of a channel at the t-th timestep as a spike feature, with each pixel's value being N-SFR.

Definition 3. *Ghost features.* There are numerous feature map pairs that resemble one another like ghosts [14, 15]. We call these feature maps ghost features.

Definition 4. *Spike patterns.* Spike features display a variety of patterns, and various patterns extract different types of information. We empirically refer to the features that focus on background information as the noise pattern since there is no background information in the input, and collectively refer to other features as the normal pattern.

Based on these definitions, we investigate the redundancy of SNNs in four granularities: spatial, temporal, channel, and neuron.

Observation 1. *In the spatial granularity, there are lots of ghost features in the spike response.*

Redundancy is inevitable in over-parameterized neural networks. For instance, from the perspective of feature maps, there are many ghost features in Conv-based ANNs (CNNs) [14, 58]. The same is true for SNNs, as demonstrated in Figure 2**a**. Plotting all spike features at the same timestep, we see that certain features concentrate on background information with a huge region of red, while others concentrate on gait information with a large area of blue, and ghost features can be seen in both patterns.

Observation 2. *In the temporal granularity, T-SFR at different timesteps does not change much.*

Given that each timestep shares the weight of 2D convolution for spatial modeling, the level of redundancy has increased significantly for SNNs that do temporal modeling. To show this, we give spike features of the same channel at various timesteps in Figure 2**b**. We see that for a fixed channel, the features derived at various timesteps differ, i.e., the human gait shifts progressively to the right as the timestep increases. Interestingly, the same channel's spike features—almost all of which is background information or all of which is information on human gait—are essentially the same at different timesteps. This demonstrates that spatio-temporal invariance will result in similar spike features at different timesteps. It also implies that redundancy in SNNs is linearly connected to timesteps.

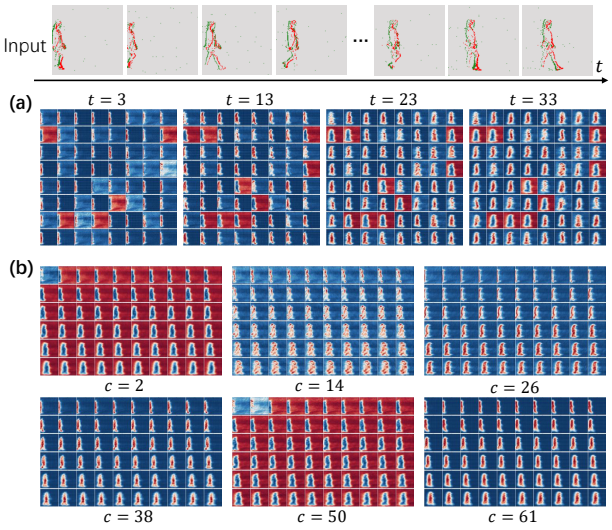


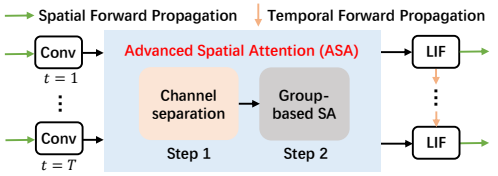
Figure 2: Inherent redundancy in SNNs exists in both spatial and temporal granularities, originating from network *over-parameterization* and *spatio-temporal invariance*, respectively. (a) spike features (averaging the spike tensors $\boldsymbol{S}^{t,n}$ over all samples) of different channels at the same timestep. Each pixel indicates the firing rate of a spiking neuron. The bluer the pixel, the closer the firing rate is to 0; the redder the pixel, the closer the firing rate is to 1. In the noise features, the background region is big and nearly totally red, indicating that many spikes are produced. (b) spike features of the same channel at different timesteps.

Observation 3. *In the channel granularity, the C-SFR is closely related to the spike patterns learned by this channel. In the neuron granularity, the N-SFR is tightly linked to the location of neurons.*

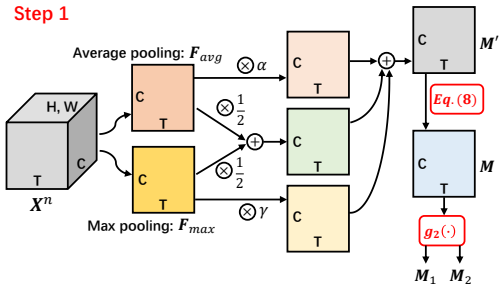
We zoom in to highlight two typical spike features that pertain to various patterns in Figure 1**c**. We see that the two features have substantially different C-SFRs. The spike feature of the noise pattern fires many spikes while focusing on trivial background information. By contrast, the spike feature of the normal pattern with lower C-SFR focuses on salient gait information in a condensed region. Furthermore, the N-SFRs of neurons in the background region of normal features are almost zero, but the N-SFRs of neurons in the same region in noise features are very high.

Definition 5. *Membrane Potential Distribution (MPD).* We input all the samples on the test set into the network. In the c-th channel of the n-th layer, we count the membrane potential values of all neurons at the t-th timestep. We can represent the membrane potential distribution of the channel by a 2D histogram, where the horizontal axis is the value of the membrane potential, and the vertical axis is the number of neurons located in a certain window.

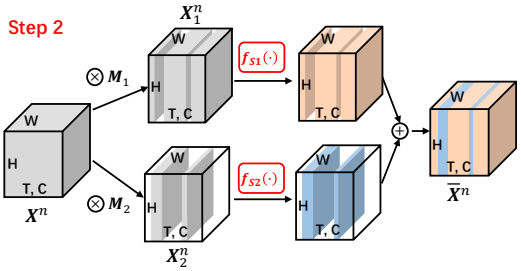
Observation 4. *Membrane potential distributions are*



(a) Overview of ASA-SNN.



(b) Channel separation technique (Eq. 5).



(c) Group-based SA (Eq. 6).

Figure 3: Details of ASA-SNN. The ASA module is divided into two steps: (b) Channel separation and (c) Group-based SA, and consists of four functions, $g_1(\cdot)$, $g_2(\cdot)$, $f_{S1}(\cdot)$, and $f_{S2}(\cdot)$.

highly correlated with spike patterns.

Note, to facilitate the analysis of the effect of the proposed method on the MPD and the spike feature, we discuss them in detail later in Section 5.4.

4. Methodology

Motivation. We can infer the following three empirical conclusions from : (1) Features can basically be separated into noise, and normal patterns; (2) The quality of the feature is determined by the MPD of the channel; (3) The firing of spiking neurons is related to their location. Based on these observations, we concentrate our optimization on the MPD of each channel, i.e., performing spatial attention. Meanwhile, considering that all features have two patterns, we exploit independent spatial attention to optimize them separately, called advanced spatial attention.

Method. As shown in Figure 3**a**, we implement our ASA module in two steps. We first exploit a channel separation technique to separate all features into two complementary groups based on their importance. Then individual SA sub-modules are performed on the two groups of features. Suppose $\boldsymbol{X}^n = \left[\cdots, \boldsymbol{X}^{t,n}, \cdots\right] \in \mathbb{R}^{T\times c_n\times h_n\times w_n}$ is an intermediate feature map as input tensor, this two-step process can be summarized by the following equations:

$$\boldsymbol{M}_1, \boldsymbol{M}_2 = g_2(g_1(\boldsymbol{X}^n)), \tag{5}$$

$$\overline{\boldsymbol{X}}^n = f_{S1}(\boldsymbol{X}^n \otimes \boldsymbol{M}_1) \oplus f_{S2}(\boldsymbol{X}^n \otimes \boldsymbol{M}_2), \tag{6}$$

where $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{T\times c_n\times 1\times 1}$ are the complementary mask (separation) maps that contain only 0 and 1 elements, $g_1(\cdot)$ is a function that assesses the channel's importance, $g_2(\cdot)$ is the separation policy function used to generate mask maps for feature grouping, $f_{S1}(\cdot)$ and $f_{S2}(\cdot)$ are SA functions with the same expression, $\overline{\boldsymbol{X}}^n$ is the output feature tensor which has the same size as $\boldsymbol{X}^n$. During multiplication, the mask score are broadcast (copied) along the temporal and channel dimensions accordingly. Finally, compared with $\boldsymbol{U}^{t,n}$ of vanilla SNN in Eq. 1, the *new membrane potential* behaviors of ASA-SNN layer follow

$$\boldsymbol{U}^{t,n} = \boldsymbol{H}^{t-1,n} + \overline{\boldsymbol{X}}^{t,n}. \tag{7}$$

Empirically, the design of $g_1(\cdot)$ is critical to the task accuracy, as well as the number of additional parameters and computations. The classic channel attention models in CNNs [19, 44, 39, 47, 26] generally judge the importance of the channel by fusing the global degree information (max pooling) and local significance information (average pooling) of the features. Inspired by these works, here we design two schemes for $g_1(\cdot)$, one that is learnable (*ASA-1*) and the other that directly judges importance based on pooled information (*ASA-2*).

As shown in Figure 3**b**, temporal-channel features are aggregated by using both average-pooling and max-pooling operations, which infer two different tensors $\boldsymbol{F}_{avg}, \boldsymbol{F}_{max} \in \mathbb{R}^{T\times c_n\times 1\times 1}$.

In ASA-1, we get the importance map $\boldsymbol{M}$ by

$$\boldsymbol{M}' = \frac{1}{2}\otimes(\boldsymbol{F}_{avg}+\boldsymbol{F}_{max}) + \alpha\otimes\boldsymbol{F}_{avg} + \gamma\otimes\boldsymbol{F}_{max}, \tag{8}$$

$$\boldsymbol{M} = \sigma\left(\boldsymbol{W}_2^n(\mathrm{ReLU}(\boldsymbol{W}_1^n(\boldsymbol{M}')))\right), \tag{9}$$

where α and γ are trainable parameters which are initialised with 0.5, σ means the sigmoid function, $\boldsymbol{W}_1^n \in \mathbb{R}^{\frac{T}{r}\times T}$ and $\boldsymbol{W}_2^n \in \mathbb{R}^{T\times\frac{T}{r}}$ are trainable parameters independent at each layer, and r represents the dimension reduction factor. Note, $\boldsymbol{M}', \boldsymbol{M} \in \mathbb{R}^{T\times c_n\times 1\times 1}$, we share $\boldsymbol{W}_1^n$ and $\boldsymbol{W}_2^n$ on the channel dimension.

In ASA-2, we set $\boldsymbol{M} = \boldsymbol{M}'$ directly. Then, we get $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, denoting the important and sub-important channel indexes respectively, by combining the y-th largest values of two dimensions in $\boldsymbol{M}$. Specifically, the pseudo-code of $g_2(\cdot)$ is represented by

```
# X: input feature [N, T, C, H, W]
# k: 0.5 * C
def select_max(X, dim="C", k=k):
    mask = zeros_like(X)
    mask[topk(X, dim=dim, k=k)] = 1
    return mask

def mask(X, k):
    mask_c = select_max(X, dim="C", k=k)
    mask_t = select_max(X, dim="T", k=k)
    mask = (mask_c + mask_t) / 2
    return where(mask == 0.5, 1, 0)
```

After obtaining $\boldsymbol{X}_1^n = \boldsymbol{X}^n \otimes \boldsymbol{M}_1$ and $\boldsymbol{X}_2^n = \boldsymbol{X}^n \otimes \boldsymbol{M}_2$, we perform individual SA module to optimize them. As shown in Fig. 4, the SA is follow [44]:

$$f_S(\cdot) = \sigma(f^{3\times3}([MaxPool(\cdot); AvgPool(\cdot)])), \quad (10)$$

where $\mathrm{AvgPool}(\cdot), \mathrm{MaxPool}(\cdot) \in \mathbb{R}^{1\times1\times h_n \times w_n}$, $f^{3\times3}$ represents a convolution operation with the filter size of 3×3, $f_S(\cdot) \in \mathbb{R}^{1\times1\times h_n \times w_n}$ is the 2-D spatial attention scores, and we set $f_{S1}(\cdot) = f_{S2}(\cdot) = f_S(\cdot)$.

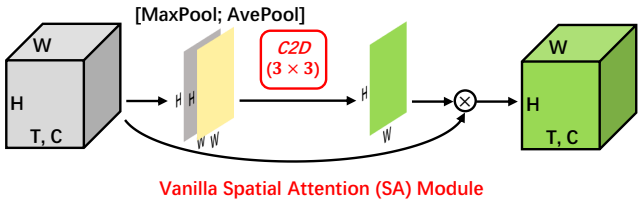


Figure 4: Diagram of spatial attention module. As illustrated, the spatial attention exploits two outputs that are pooled along the temporal-channel axis and forward them to a 3×3 convolution layer.

5. Experiments

For an event stream, we exploit the frame-based representation [48, 8] as the preprocessing method to convert it into an event frame sequence. Suppose the interval between two frames (i.e., temporal resolution) is dt and there are T frames (i.e., timesteps), the length of the input event stream is $t_{lat} = dt \times T$ millisecond. After processing these divided frames through SNN, a prediction can then be retrieved.

5.1. Experimental Setup

We evaluate our method on five datasets, all generated by recording actions in real scenes. DVS128 Gesture [1], DVS128 Gait-Day [41], and DVS128 Gait-Night [42] were captured by a 128x128 pixel DVS128 camera. As their names imply, Gesture comprises hand gestures, while Gait-day and Gait-night include human gaits in daylight and at night, respectively. DailyAction-DVS [29] and HAR-DVS [40] are acquired by a DAVIS346 camera with a spatial resolution of 346x260, of which HAR-DVS has 300 classes and 107,646 samples and is currently the *largest* event-based human activity recognition (HAR) dataset. The raw HAR-DVS exceeds 4TB. The authors convert each event stream into frames and randomly sample 8 frames to form a new HAR-DVS for ease of processing.

Dataset	Model	Acc.(%)	NASFR
Gesture	LIF-SNN [48]	91.3	0.176
	+ ASA (Ours)	**95.2(+3.9)**	0.038(**-78.4%**)
	LIF-SNN [8]	95.5	0.023
	+ ASA (Ours)	**97.7(+2.2)**	0.018(**-21.7%**)
Gait-day	LIF-SNN [48]	88.6	0.214
	+ ASA (Ours)	**93.6(+5.0)**	0.045(**-78.9%**)
Gait-night	LIF-SNN [48]	96.4	0.197
	+ ASA (Ours)	**98.6(+2.2)**	0.126(**-36.0%**)
DailyAction -DVS	LIF-SNN [8]	92.5	0.017
	+ ASA (Ours)	**94.6(+2.1)**	0.013(**-23.5%**)
HAR-DVS	Res-SNN-18 [7]	45.5	0.206
	+ ASA (Ours)	**47.1(+1.6)**	0.183(**-11.2%**)

Table 1: Main results of vanilla vs. ASA- SNNs (ASA-1). Except for HAR-DVS, reported accuracies are average of five replicates.

We execute the baseline for each group of ablation trials, then plug the proposed ASA to run the model again (Table 1). Each group of trials for vanilla and ASA- SNNs employed the same hyper-parameters, training methods, and other training conditions[2]. In all experiments, we exploit a total of three baselines with different structures. We carefully selected baselines for various datasets to examine the relationship between the spike firing, the dataset, and the network structure. One is the shallow three-layer Conv-based LIF-SNN presented in [48, 51]. The other is a deeper five-layer Conv-based LIF-SNN, following [8]. Finally, the Res-SNN-18 [7] in the SpikingJelly framework[3] is used to verify the large datasets.

5.2. Ablation Study for ASA Module

In terms of accuracy and NASFR, We present the main results in Table 1. ASA-SNN achieves higher task accuracy with lower spike firing in all ablation studies. The performance and energy gains are more noticeable, particularly when the network structure is small. For example, in

[2]Details of datasets and training are given in the Supplementary.
[3]https://github.com/fangwei123456/spikingjelly

the Gait-day, plugging the ASA module into a three-layer SNN [48] can reduce the spike counts by 78.9% and improve the performance by +5.0 percent. This is crucial for the deployment of SNN algorithms on neuromorphic chips, which usually have strict memory limitations [3, 34, 36]. The ASA module also performs well on the deep Res-SNN. For instance, on HAR-DVS, the ASA-SNN outperforms the original SNN +1.7 percent while firing fewer spikes. In addition, we provide more ablation studies on the ASA module in the Supplementary.

Although it is beyond the scope of this work, by observing results in Table 1, we raise another complex and important question: "*What factors affect the redundancy of SNN?*" Intuitively, we could exploit NASFR as a redundancy indicator for SNNs. We argue that the NASFR of SNNs depends on various factors, the core of which includes dataset size, network size, spiking neuron types, etc. For instance, on Gesture, the NASFRs in three-layer [48] and five-layer vanilla SNN [8] are 0.176 and 0.023, respectively. Empirically, vanilla SNN's NASFR also affects the function of the ASA module, where SNNs with more redundancy may be easier to reduce spikes. We hope that these observations will inspire more theoretical and optimization work on redundancy.

5.3. Comparison with the State-of-the-Art

In Table 2, we make a comprehensive comparison with prior works in terms of input temporal window and accuracy. Since some datasets were created recently, there is a lack of benchmarks in the field of SNNs. In this paper, we benchmark these datasets using models from the open-source framework SpikingJelly and fill in the corresponding accuracies in Table 1 and Table 2. We can see that on four small datasets, ASA-SNN can produce SOTA or comparable performance. Compared to GCN methods [41, 42] with full input, we observe that SNNs can always achieve higher performance with less input (i.e., smaller $dt \times T$). Moreover, on the largest HAR-DVS dataset, our Top-1 accuracy is 47.1% based on Res-SNN-18, which is comparable to the ANN-based benchmark results from 46.9% to 51.2%. This is a reasonable result since SNNs employ binary spikes, generally gaining higher energy efficiency at the expense of accuracy.

5.4. Comparison with Other Attention SNNs

In this work, based on redundancy analysis, we design the ASA module, which only performs spatial attention. As mentioned, the current practice of attention mechanisms in SNNs [51, 30, 59, 50] is dominated by multi-dimensional composition. An easily overlooked fact is that adding attention modules inevitably introduces additional computation. These extra computations are trivial in CNNs, but require special care in SNNs, as otherwise the energy advantage of attention SNNs is lost. Specifically, the energy shift between vanilla and attention SNNs can be computed as

$$\Delta_E = E_{MAC} \cdot \Delta_{MAC} - E_{AC} \cdot \Delta_{AC}, \qquad (11)$$

where $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$ represent the energy cost of Multiply-and-Accumulate (MAC) and AC operation [18], Δ_{MAC} and Δ_{AC} represent the additional MAC operation and the reduced AC number caused by the attention modules, respectively (detailed energy evaluation is in the Supplementary). We need to try our best to make the benefit ($E_{AC} \cdot \Delta_{AC}$) outweigh the cost ($E_{MAC} \cdot \Delta_{MAC}$).

Dataset	Methods	$dt \times T$	Acc. (%)
Gesture	12 layers CNN [1]	1×120	92.6
	PLIF-SNN [8]	300×20	97.6
	Res-SNN-18 [48]	375×16	97.9
	MA-SNN [51]	300×20	**98.2**
	This Work	300×20	97.7
Gait-day	EV-Gait GCN [41]	4400×1	89.9
	TA-SNN [48]	15×60	88.6
	3D GCN [42]	1500×1	86.0
	MA-SNN [51]	15×60	92.3
	This Work	15×60	**93.6**
Gait-night	TA-SNN [48]	15×60	96.4
	3D GCN [42]	5500×1	96.0
	This Work	15×60	**98.6**
DailyAction -DVS	HMAX-SNN [28]	-	76.9
	Motion-SNN [29]	-	90.3
	PLIF-SNN [8]	120×36	92.5
	This Work	120×36	**94.6**
HAR-DVS	Res-CNN-18 [16]	$T = 8$	49.2
	ACTION-Net [43]	$T = 8$	46.9
	TimeSformer [2]	$T = 8$	50.8
	SlowFast [9]	$T = 8$	46.5
	ES-Transformer [40]	$T = 8$	**51.2**
	Res-SNN-18 [7]	$T = 8$	45.5
	This Work	$T = 8$	47.1

Table 2: The comparison between the proposed methods and existing SOTA techniques on five event-based vision datasets. Note, all the results of the ANN models in HAR-DVS in this table are taken from [40]. (Bold: the best)

Model	Acc. (%)	Params (↑)	NASFR	Δ_{MAC} (↑)
Vanilla SNN [48]	88.6	2,323,531	0.214	-
+ SA [51]	89.5(+0.9)	+294	0.091(-57.4%)	+14.3M
+ TCSA [51]	92.3(+3.7)	+24,126	**0.045(-78.9%)**	+27.3M
+ **ASA-1 (Ours)**	**93.6(+5.0)**	+10,914	**0.045(-78.9%)**	**+2.6M**
+ **ASA-2 (Ours)**	89.6(+1.0)	**+114**	0.088(-58.9%)	**+2.6M**
Vanilla SNN [48]	91.3	2,323,531	0.176	-
+ SA [51]	92.6(+1.3)	+294	0.073(-58.5%)	+14.3M
+ TCSA [51]	96.5(**+5.2**)	+24,126	**0.029(-83.5%)**	+27.3M
+ **ASA-1 (Ours)**	95.2(+3.9)	+10,914	0.038(-78.4%)	**+2.6M**
+ **ASA-2 (Ours)**	94.4(**+3.1**)	**+114**	0.050(-71.6%)	**+2.6M**

Table 3: Effect of Different attention modules in three-layer SNN[48] on Gait-day (the above table) and Gesture (the below table) with $dt = 15, T = 60$.

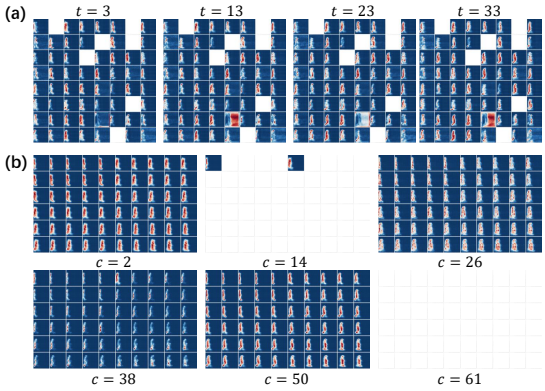
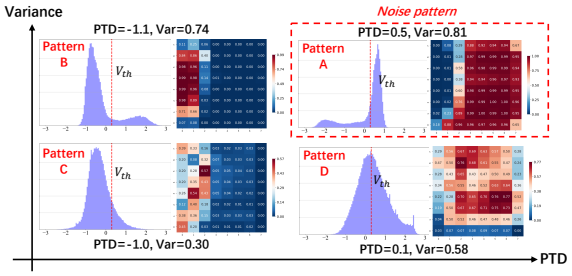


Figure 5: Spike features in ASA-SNN. (a) Spike features of different channels at the same timestep. (b) Spike features of the same channel at different timesteps.

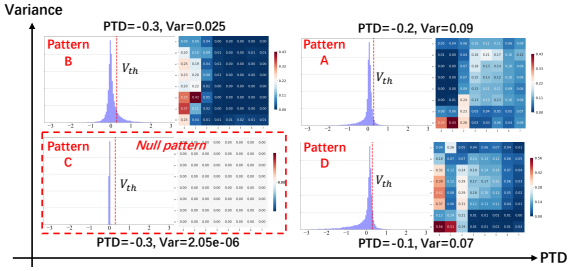
In Table 3, we compare the number of extra parameters and computations needed for various attention modules. We see that ASA module is a cost-effective solution, less Δ_{MAC} (just 2.4M), better or comparable performance. For instance, a 78.9% decrease in spike firing results in a 65% reduction in energy consumption in the TCSA-SNN [51] but a 76% reduction in our ASA-SNN. The ASA-2 design, which only adds 114 parameters to obtain a nice performance improvement on Gesture, is highlighted lastly (albeit it is not stable).

5.5. Result Analysis

Spike patterns in ASA-SNNs. We re-examine the spike response in ASA-SNNs as we did in Section 3.2. In the spatial granularity, the spike patterns in ASA-SNNs are altered. As shown in Figure 5**a**, there are almost no noise features,



(a) Spike patterns in vanilla SNNs.



(b) Spike patterns in ASA-SNNs.

Figure 6: When the ASA is plugged, the spike pattern shifts in membrane potential distribution and spike feature.

but some null features without spikes appear. In the temporal granularity, spatial-temporal invariance still holds. As depicted in Figure 5**b**, spike features of the same channel at different timesteps are similar.

Membrane Potential Distribution (MPD) and spike pattern. We already know that the redundancy in SNNs depends directly on the learned spike patterns. Therefore, we are interested in the question of "how the spike feature changes", which can help us understand the dynamics inside the network and inspire future work. Here we analyze the relationship between the MPD and the spike feature (pattern). We define the following indicator.

Definition 6. *Peak-to-threshold distance (PTD).* We picked out the highest three pillars in membrane potential distribution and obtained the peak interval by averaging these pillars' membrane potential intervals. We then define peak-to-threshold distance as the difference between the center point of the peak interval and the threshold.

Observation 5. *The PTD and Variance of membrane potential distribution of a channel can be exploited to measure the quality of the spike feature extracted by this channel to a certain extent. When the value of PTD is near 0 or greater than 0, it indicates that the membrane potential of most spiking neurons on a map is to the right of the threshold. Consequently, most neurons have a relatively high neuron spike firing rate, and intuitively, the pattern learned by the channel is background noise since the key information is usually located within a small area. The variance measures the degree of focus. In the same or similar normal pattern*

of the same model, the larger the variance, the clearer the edge information of the learned feature.

Accordingly, as shown in Figure 6, we show how the spike feature follows the MPD. Specifically, in vanilla SNNs (Figure 6**a**), spike features and MPDs in Patterns A and B appear to be in a complementary relationship, corresponding to perfect focus on the background and object regions, respectively. If the PTD is maintained constant, as the variance gradually decreases, the information in the edge regions of the background and object begins to blur, as shown in Patterns C and D.

Then we compare the shifts in spike features of vanilla and ASA-SNNs. Obviously, peak regions of the MPDs across all channels in ASA-SNNs are located to the left of the threshold (Figure 6**b**), i.e., $PTD < 0$. This indicates that one channel does not fire a lot of spikes after the ASA module has optimized the MPD. That is, the MPD is highly compact, which implies that the edge information of the spike feature is clearer.

By combining the two indicators PTD and variance, we can quickly determine what a "good" spike feature's MPD should be. For instance, as shown in Pattern B of ASA-SNNs, both the PTD and variance values should fall within an appropriate range, neither too high nor too low.

Model	Gesture	Gait-day	HAR-DVS
Vanilla SNN	0.158	0.362	0.584
ASA-SNN	0.024	0.031	0.339

Table 4: Comparison of QEs on vanilla and ASA-SNNs.

Information loss. As discussed in [13, 12], a good MPD can reduce information loss, which arises from the quantization error (QE) introduced by converting the analog membrane potential into binary spikes. Similar to [13, 12], we define the QE as the square of the difference between the membrane potential and its corresponding quantization spike value. The proposed ASA module concurrently optimizes the PTD and variance of MPDs in vanilla SNNs, which significantly reduces the information loss caused by the spike quantization (see Table 4). It is evident from a comparison of Figure 6**a** and **b** that each channel's MPD grows thinner (the variance becomes larger). In this work, the increased variance implies that the edge information in the spike feature is clearer from the perspective of feature visualization. By contrast, from the perspective of QE, it implies that the information loss becomes less.

6. Conclusions

In this work, three key questions are exploited to analyze the redundancy of SNNs, which are usually ignored in other prior works. To answer these questions, we present a new perspective on the relationship between the spatio-temporal dynamics and the spike firing. These findings inspired us to develop a simple yet efficient advanced spatial attention module for SNNs, which harnesses the inherent redundancy in SNNs by optimizing the membrane potential distribution. Experimental results and analysis show that the proposed method can greatly reduce the spike firing and further improve performance. The new insight onto SNN redundancy not only reveals the unique advantages of spike-based neuromorphic computing in terms of bio-plausibility, but may also bring some interesting enlightenment to the following-up research on efficient SNNs.

Acknowledgement

This work was partially supported by National Science Foundation for Distinguished Young Scholars (62325603), and National Natural Science Foundation of China (62236009, U22A20103), and Beijing Natural Science Foundation for Distinguished Young Scholars (JQ21015).

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[3] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[4] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, and et al. Rethinking the performance comparison between snns and anns. *Neural Networks*, 121:294–307, 2020.

[5] Lei Deng, Yujie Wu, Yifan Hu, Ling Liang, Guoqi Li, Xing Hu, Yufei Ding, Peng Li, and Yuan Xie. Comprehensive snn compression using admm optimization and activity regularization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021.

[6] Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *arXiv preprint arXiv:2109.12894*, 2021.

[7] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS2021)*, 2021.

[8] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable

membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2661–2671, October 2021.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

[11] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, pages 1–38, 2022.

[12] Yufei Guo, Yuanpei Chen, Liwen Zhang, YingLei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[13] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2022.

[14] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020.

[15] Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chunjing Xu, Enhua Wu, and Qi Tian. Ghostnets on heterogeneous devices via cheap operations. *International Journal of Computer Vision*, 130(4):1050–1069, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Andreas VM Herz, Tim Gollisch, Christian K Machens, and Dieter Jaeger. Modeling single-neuron dynamics and computations: a balance of detail and abstraction. *science*, 314(5796):80–85, 2006.

[18] Mark 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)*, pages 10–14. IEEE, 2014.

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[20] Yifan Hu, Lei Deng, Yujie Wu, Man Yao, and Guoqi Li. Advancing spiking neural networks towards deep residual learning. *arXiv preprint arXiv:2112.08954*, 2021.

[21] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H Ang Jr. Tada! temporally-adaptive convolutions for video understanding. In *ICLR*, 2022.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[23] Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, and Priyadarshini Panda. Neural architecture search for spiking neural networks. *arXiv preprint arXiv:2201.10355*, 2022.

[24] Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2021.

[25] Guoqi Li, Lei Deng, Huajing Tang, Gang Pan, Yonghong Tian, Kaushik Roy, and Wolfgang Maass. Brain inspired computing: A systematic survey and future trends. 2023.

[26] Guoqiang Li, Qi Fang, Linlin Zha, Xin Gao, and Nenggan Zheng. Ham: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recognition*, 129:108785, 2022.

[27] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128x128 120 db 15 microsecond latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-state Circuits*, 43(2):566–576, 2008.

[28] Qianhui Liu, Haibo Ruan, Dong Xing, Huajin Tang, and Gang Pan. Effective aer object classification using segmented probability-maximization learning in spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1308–1315, 2020.

[29] Qianhui Liu, Dong Xing, Huajin Tang, De Ma, and Gang Pan. Event-based action recognition using motion information and spiking neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1743–1749, 2021.

[30] Xin Liu, Mingyu Yan, Lei Deng, Yujie Wu, De Han, Guoqi Li, Xiaochun Ye, and Dongrui Fan. General spiking neural network framework for the learning trajectory from a noisy mmwave radar. *Neuromorphic Computing and Engineering*, 2(3):034013, 2022.

[31] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[32] Byunggook Na, Jisoo Mok, Seongsik Park, Dongjin Lee, Hyeokjun Choe, and Sungroh Yoon. AutoSNN: Towards energy-efficient spiking neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16253–16269. PMLR, 17–23 Jul 2022.

[33] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Learning to be efficient: Algorithms for training low-latency, low-compute deep spiking neural networks. In *Proceedings of the 31st annual ACM symposium on applied computing*, pages 293–298, 2016.

[34] Jing Pei, Lei Deng, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

[35] Arjun Rao, Philipp Plank, Andreas Wild, and Wolfgang Maass. A long short-term memory for ai applications in spike-based neuromorphic hardware. *Nature Machine Intelligence*, 4(5):467–479, 2022.

[36] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[37] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Bill Kay, et al. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

[38] Ahmed Shaban, Sai Sukruth Bezugam, and Manan Suri. An adaptive threshold neuron for recurrent spiking neural networks with nanodevice hardware implementation. *Nature Communications*, 12(1):1–11, 2021.

[39] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Xiao Wang, Zongzhen Wu, Bo Jiang, Zhimin Bao, Lin Zhu, Guoqi Li, Yaowei Wang, and Yonghong Tian. Hardvs: Revisiting human activity recognition with dynamic vision sensors. *arXiv preprint arXiv:2211.09648*, 2022.

[41] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2019.

[42] Yanxiang Wang, Xian Zhang, Yiran Shen, Bowen Du, Guangrong Zhao, Lizhen Cui, and Hongkai Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3436–3449, 2022.

[43] Zhengwei Wang, Qi She, and Aljosa Smolic. Action-net: Multipath excitation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13214–13223, 2021.

[44] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[45] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

[46] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1311–1318, 2019.

[47] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International Conference on Machine Learning*, pages 11863–11874. PMLR, 2021.

[48] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10221–10230, October 2021.

[49] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *arXiv preprint arXiv:2307.01694*, 2023.

[50] Man Yao, Hengyu Zhang, Guangshe Zhao, Xiyu Zhang, Dingheng Wang, Gang Cao, and Guoqi Li. Sparser spiking activity can be better: Feature refine-and-mask spiking neural network for event-based visual recognition. *Neural Networks*, 166:410–423, 2023.

[51] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2023.

[52] Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

[53] Hang Yin, John Boaz Lee, Xiangnan Kong, Thomas Hartvigsen, and Sihong Xie. Energy-efficient models for high-dimensional spike train classification using sparse spiking neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2017–2025, 2021.

[54] Chengting Yu, Zheming Gu, Da Li, Gaoang Wang, Aili Wang, and Erping Li. Stsc-snn: Spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks. *Frontiers in Neuroscience*, 16, 2022.

[55] Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021.

[56] Rong Zhao, Zheyu Yang, Hao Zheng, Yujie Wu, Faqiang Liu, Zhenzhi Wu, Lukai Li, Feng Chen, Seng Song, Jun Zhu, et al. A framework for the general design and computation of hybrid neural networks. *Nature Communications*, 13(1):1–12, 2022.

[57] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:11062–11070, 2021.

[58] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[59] Rui-Jie Zhu, Qihang Zhao, Tianjing Zhang, Haoyu Deng, Yule Duan, Malu Zhang, and Liang-Jian Deng. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *arXiv preprint arXiv:2206.10177*, 2022.