Masked Spiking Transformer

Ziqing Wang[1,2*] Yuetong Fang[1*] Jiahang Cao[1] Qiang Zhang[1] Zhongrui Wang[3,4†] Renjing Xu[1†]

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]North Carolina State University
[3]The University of Hong Kong [4]ACCESS - AI Chip Center for Emerging Smart Systems

zwang247@ncsu.edu, {yfang870, jcao248, qzhang749}@connect.hkust-gz.edu.cn
zrwang@eee.hku.hk, renjingxu@ust.hk

Abstract

The combination of Spiking Neural Networks (SNNs) and Transformers has attracted significant attention due to their potential for high energy efficiency and high-performance nature. However, existing works on this topic typically rely on direct training, which can lead to suboptimal performance. To address this issue, we propose to leverage the benefits of the ANN-to-SNN conversion method to combine SNNs and Transformers, resulting in significantly improved performance over existing state-of-the-art SNN models. Furthermore, inspired by the quantal synaptic failures observed in the nervous system, which reduce the number of spikes transmitted across synapses, we introduce a novel Masked Spiking Transformer (MST) framework. This incorporates a Random Spike Masking (RSM) method to prune redundant spikes and reduce energy consumption without sacrificing performance. Our experimental results demonstrate that the proposed MST model achieves a significant reduction of 26.8% in power consumption when the masking ratio is 75% while maintaining the same level of performance as the unmasked model. The code is available at: `https://github.com/bic-L/Masked-Spiking-Transformer`.

1. Introduction

Spiking neural networks (SNNs), considered as the next generation neural networks [30], are brain-inspired neural networks based on the dynamic characteristics of biological neurons [31, 17]. SNNs have attracted significant attention due to their unique properties in handling sparse data, which can yield great energy efficiency benefits on neuromorphic hardware. Due to their specialties, they have been widely utilized in various fields, such as classification [32, 18], object detection [3] and tracking [51], etc. Nevertheless, SNNs currently can hardly realize a comparable performance to that of artificial neural networks (ANNs), especially for complex tasks such as ImageNet [40].

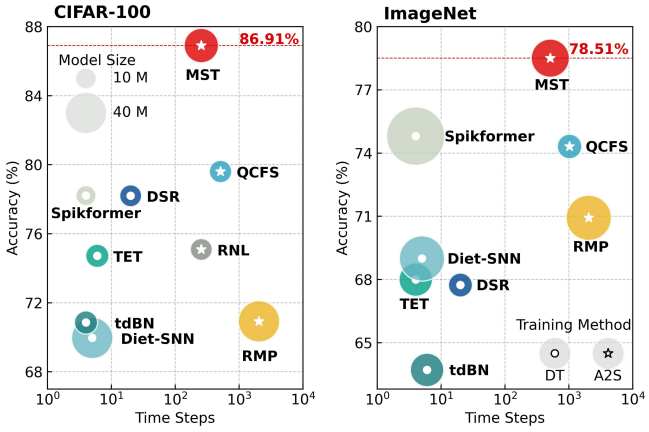

Figure 1. Performance of MST and other SOTA SNN models regarding top-1 accuracy and time steps. The markers, represented by circles and star shapes, denote the direct training (DT) and the ANN-to-SNN (A2S) conversion method, respectively, where the marker size corresponds to the model size. Results show that the proposed MST model achieves higher accuracy than the other SNN models.

In order to improve the performance of SNNs, various training methods have been proposed, broadly categorized as the direct training method and the ANN-to-SNN conversion method. Direct training methods leverage a continuous relaxation of the non-smooth spiking mechanism to enable backpropagation with a surrogate gradient function for handling non-differentiability [36], but this can lead to unstable gradient propagation and relatively low accuracy

*Equal contribution.
†Corresponding author

compared to leading ANNs [38]. Alternatively, ANN-to-SNN conversion methods convert pre-trained ANNs into SNNs for better performance while requiring more time steps, with increased power consumption to reduce conversion errors [46, 25, 2, 7]. Our focus is on implementing the ANN-to-SNN conversion method to narrow the performance gap between leading ANNs and SNNs, but the required long time steps pose challenges in reducing energy consumption. Therefore, identifying strategies to decrease power consumption while maintaining excellent performance is crucial.

The biological nervous system offers valuable insights for addressing the challenges of implementing high-performance Spiking Transformers using the ANN-to-SNN conversion method. The quantal synaptic failure theory suggests that missing information during neuronal signal transmission may not impact the computational information transmitted to a postsynaptic neuron under certain conditions, but can reduce energy consumption and heat production [23]. Likewise, in the ANN-to-SNN conversion process, missing spikes can possibly be compensated for by leveraging the correlations between signals in the space and time domains during the information propagation over multiple time steps. In addition, neural network models possess lots of redundant connections: prior works reveal that the redundancy in the self-attention module of Transformers can be pruned without significantly impacting performance [34, 49]. Therefore, eliminating redundant information during the transmission of neuronal signals can possibly reduce overall energy consumption in the Spiking Transformer model while preserving high performance.

In our work, we propose a Masked Spiking Transformer (MST), which incorporates a Random Spike Masking (RSM) method designed specifically for SNNs. The RSM method randomly selects only a subset of input spikes, significantly reducing the number of spikes involved in the computation process. We evaluate the MST model on both static and neuromorphic datasets, demonstrating its superiority over existing SNN models. Our experiments show that the RSM method can reduce energy consumption on the self-attention module and the MLP module in Transformer, enabling the SNNs to take advantage of energy efficiency and high performance. Furthermore, the proposed RSM method is not limited to Transformer, but can be extended to other backbones such as ResNet and VGG, highlighting its potential as a general technique to improve SNN efficiency. Our results demonstrate the potential of this approach to provide a new direction for developing high-performance and energy-efficient SNN models.

The main contributions of this paper can be summarized as follows:

- We propose a Masked Spiking Transformer (MST) using the ANN-to-SNN conversion method. To the best of our knowledge, it is the first exploration of applying the self-attention mechanism fully in SNNs utilizing the ANN-to-SNN conversion method.
- The MST model is evaluated on both static and neuromorphic datasets, and the results show that it outperforms state-of-the-art (SOTA) SNNs on all datasets. In specific, the top-1 accuracy of the MST model is 1.21%, 7.3%, and 3.7% higher than the current SOTA SNN model on the CIFAR-10, CIFAR-100, and ImageNet datasets, respectively.
- We design a Random Spike Masking (RSM) method for SNNs trained with the ANN-to-SNN conversion method to prune the redundant spikes during inference and save energy consumption.
- Extensive experiments show that our proposed RSM is a versatile and general method that can be utilized in other spike-based deep networks, such as ResNet and VGG SNN model variants.

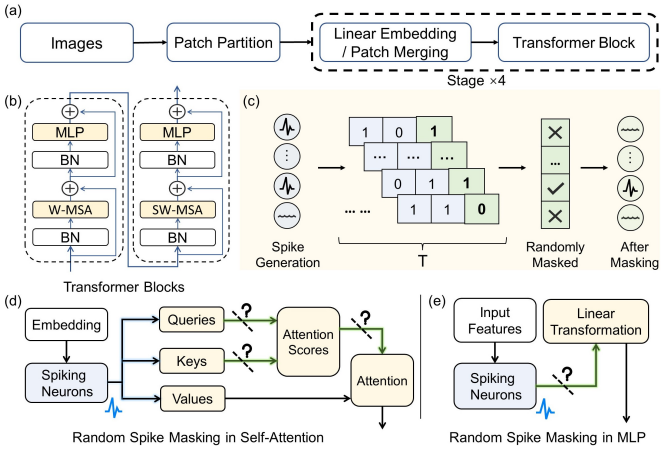

Figure 2. Overview of our MST. (a) Schematic of the model architecture of the Swin Transformer, which is the backbone of our model. (b) Schematic of the proposed Transformer blocks, where BN layers replace the original LN layers. (c) Conceptual illustration of the RSM method, which involves randomly masking the input spike. (d-e) The RSM method in self-attention and MLP module.

2. Related Work

Spiking Neural Networks SNNs have gained popularity in the field of brain-inspired intelligence due to their compatibility with neuromorphic hardware and biological properties. With the increasing interest in larger-scale and higher-performance SNNs, recent research has focused on developing novel training algorithms and architectures.

Zheng et al. proposed a threshold-dependent batch normalization (tdBN) method based on spatiotemporal backpropagation to train a large-scale SNN model with 50 layers [56]. Besides, Fang et al. proposed the SEW ResNet architecture for residual learning in deep SNNs to overcome the gradient vanishing problem [10]. Later, they introduced a training algorithm that learns the threshold of each spiking neuron to improve the performance of SNNs [11]. However, these methods mainly discuss the SNN models that are dominated by convolutional layers, such as the SNN variants of VGG [42] and ResNet [15]. Despite their improvements, the performance of these methods still struggles to match their ANN counterparts, limiting the application of SNNs. In this context, our proposed work focuses on implementing the self-attention mechanism in SNNs to design a Spiking Transformer that improves the performance of SNNs.

Transformer Transformer [47] was first introduced in Natural Language Processing (NLP) and quickly gained popularity for its remarkable capabilities in capturing long-range dependencies. Its success in NLP has inspired researchers to explore its potential in computer vision. Vision Transformer (ViT) [8] was the first attempt to apply the Transformer to image classification. ViT has achieved impressive results on various computer vision benchmarks, demonstrating the effectiveness of the self-attention mechanism in image understanding. Following the success of ViT, a series of works [29, 14] proposed improvements to the original ViT architecture. Motivated by the success of Transformers and its variations, this paper proposes a new architecture for SNNs that leverages the capacities of the Transformer and the energy efficiency of SNNs.

Spiking Transformer The combination of the Transformer and SNNs can achieve better performance, which has been discussed in prior studies, including STNet [54] and Spike-T [55]. These models utilized separate branches of SNNs and Transformers for feature extraction, leading to the inability to run independently on neuromorphic hardware and failing to exploit the energy efficiency benefits of SNNs fully. In addition, Mueller et al. [35] proposed a Spiking Transformer using the ANN-to-SNN conversion method, but they did not implement the self-attention module in SNNs. The recently proposed Spikformer [57] directly trained the Transformer in SNNs, but still struggles to achieve comparable performance to leading ANNs. To address these limitations, we apply the self-attention mechanism fully in SNNs by utilizing the ANN-to-SNN conversion method and propose the RSM method to improve both the performance and energy efficiency of the Spiking Transformer. Our model offers a new direction for developing high-performance SNNs using the ANN-to-SNN conversion method.

3. Methods

3.1. Spiking Neuron Model

For ANNs, the input $\boldsymbol{a}^{l-1}$ to layer l is mapped to the output $\boldsymbol{a}^l$ by a linear transformation matrix $\boldsymbol{W}^l$ and a nonlinear activation function $f(\cdot)$, that is $(l = 1, 2, 3, \cdots, L)$:

$$\boldsymbol{a}^l = f\left(\boldsymbol{W}^l \boldsymbol{a}^{l-1}\right) \tag{1}$$

where $f(\cdot)$ is often set as the $ReLU$ activation function.

In SNNs, the Integrate-and-Fire (IF) spiking neuron model is commonly used in ANN-to-SNN conversion [25, 2, 7]. The dynamics of the IF model are described by:

$$\boldsymbol{v}^l(t) = \boldsymbol{v}^l(t-1) + \boldsymbol{W}^l \theta^{l-1} \boldsymbol{s}^{l-1}(t) - \theta^l \boldsymbol{s}^l(t) \tag{2}$$

where $\boldsymbol{v}^l(t)$ denotes the membrane potential of neurons in layer l at time step t, which corresponds to the linear transformation matrix $\boldsymbol{W}^l$, the threshold θ^l, and the binary output spikes of neurons in the previous layer $l-1$, denoted as $\boldsymbol{s}^{l-1}(t)$. The $s^l(t)$ is defined as follows:

$$\boldsymbol{s}^l(t) = H\left(\boldsymbol{u}^l(t) - \theta^l\right) \tag{3}$$

where $\boldsymbol{u}^l(t) = \boldsymbol{v}^l(t-1) + \boldsymbol{W}^l \theta^{l-1} \boldsymbol{s}^{l-1}(t)$ denotes the membrane potential of neurons before the trigger of a spike at time step t, $H(\cdot)$ denotes the Heaviside step function. The neurons generate output spikes whenever the membrane potential $\boldsymbol{u}^l(t)$ exceeds the threshold value θ^l, and the membrane potential is reset by subtracting the threshold value to reduce information loss [41].

3.2. ANN-to-SNN conversion

To achieve the ANN-SNN conversion, a relationship is established between the rectified linear unit (ReLU) activation of analog neurons in ANNs and the firing rate or postsynaptic potential of spiking neurons in SNNs. This is obtained by summing Eq. 2 from time step 1 to T dividing T on both sides, resulting in the following equation:

$$\frac{\boldsymbol{v}^l(T) - \boldsymbol{v}^l(0)}{T} = \frac{\sum_{t=1}^{T} \boldsymbol{W}^l \theta^{l-1} \boldsymbol{s}^{l-1}(t)}{T} - \frac{\sum_{t=1}^{T} \theta^l \boldsymbol{s}^l(t)}{T} \tag{4}$$

The linear relationship between $\phi^l(T)$ and $\phi^{l-1}(T)$ is established by defining $\phi^l(T) = \frac{\sum_{t=1}^{T} \theta^l s^l(t)}{T}$ as the average postsynaptic potential:

$$\boldsymbol{\phi}^l(T) = \boldsymbol{W}^l \boldsymbol{\phi}^{l-1}(T) - \frac{\boldsymbol{v}^l(T) - \boldsymbol{v}^l(0)}{T} \tag{5}$$

The equivalence between Eq. 1 and 5 holds only as T goes to infinity, resulting in a conversion error. To address this issue, we replace the ReLU activation function with the quantization clip-floor-shift (QCFS) [2] function in the ANNs.

3.3. Model Architecture

An overview of the MST is depicted in Fig. 2, where the Swin Transformer [29] is adopted as the backbone network. To convert the original network into a fully-spiking manner, we incorporate the QCFS activation functions after each linear or regularization layer during the training phase, which are replaced with Integrate-and-Fire (IF) neurons in the inference process, resulting in more efficient computation.

The entire computation process in the spiking self-attention module can be formulated as:

$$\begin{aligned} Q_{spk}[t] &= \mathrm{IF}(X[t] * W_q) \\ K_{spk}[t] &= \mathrm{IF}(X[t] * W_k) \end{aligned} \tag{6}$$

where Q_{spk}, K_{spk} denote the spike matrices of the query and key at t time step, $\mathrm{IF}(\cdot)$ is the IF neuron function, W_q, W_k denote the corresponding weight matrices. Attention score is defined as:

$$A_{spk}[t] = \mathrm{IF}(\frac{Q_{spk}[t] * K^T_{spk}[t]}{\sqrt{d}}) \tag{7}$$

where A_{spk} represents the spike matrix of the attention score calculated by the dot product of the query spike matrix and the key spike matrix, and d is a scaling factor equal to the feature dimension of a given attention head.

The calculation of LN and BN can be expressed by:

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta \tag{8}$$

where x denotes the input tensor to be normalized, $\mathrm{E}[x]$ and $\mathrm{Var}[x]$ represent the mean and variance of x, ϵ is a small constant added to the variance for numerical stability, γ and β are trainable scaling and bias parameters respectively, and y is the normalized output of the layer.

In LN, the mean and variance are calculated across the features of each sample in a batch. Therefore, each sample in the batch has its normalization parameters. On the other hand, BN calculates the mean and variance across all samples in a batch for each feature, which means the normalization parameters are shared across all samples in a batch.

Normalization is important for ensuring the feasibility of ANN-to-SNN conversion. As illustrated in Fig. 3, different normalization approach results in vastly different membrane potential in the inference process. Normalizing along the channel dimension (LN) cause a distribution mismatch between ANN and SNN, which leads to performance degradation, while normalizing along the batch dimension (BN) preserves a similar result. Replace all LN layers with BN layers is a straightforward approach to normalize the post-activation distribution of ANNs during the conversion process, but for large datasets like ImageNet, there are convergence issues. We resolve this problem by simply adding a BN layer after each linear layer in the MLP module, inspired by [53]. Mathematically, the first layer of the modified MLP module is formulated as follows:

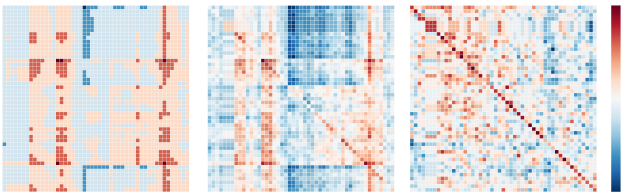

Figure 3. Illustration of distributions of (a) post-activation distribution in ANN, and (b-c) cumulative membrane potential distributions of SNN model with BN and LN, respectively. The heatmap shows a similar distribution between ANN and SNN(BN) models, while the widely varying distribution between ANN and SNN(LN) models leads to performance degradation.

$$\mathrm{MLP}(x) = \mathrm{IF}(\mathrm{Linear}(\mathrm{BN}(x))) \tag{9}$$

where x represents the input tensor, Linear denotes the linear layer, BN represents the batch normalization layer. Notably, the MLP module consists of two linear layers, each followed by a BN layer and an IF neuron function.

3.4. Random Spike Masking Method

Implementing the Transformer in SNNs using the ANN-to-SNN conversion method presents a challenge due to the high power consumption demand. To address this issue, we propose the Random Spike Masking (RSM) method for reducing redundant spikes during inference.

Traditional ANN pruning methods remove weights with low magnitudes, which are believed to have little impact on the final output of the model [58]. In contrast, our RSM method prunes input spikes randomly according to the masking ratio, where each spike ($s = 1$) has the probability of turning into a failure state ($s = 0$) that can be implemented using binary mask matrices.

It is important to note that our proposed RSM method for spike pruning in SNNs differs from Dropout [44] used in ANNs. Dropout is a regularization technique that randomly masks some neurons during training, but all neurons are used for inference, and the output needs to be scaled by the retained probability used during training. This means that Dropout cannot directly reduce the number of spikes in the network. Unlike Dropout, the RSM method reduces the computational cost of SNNs by randomly pruning input spikes in both training and inference, explicitly removing redundancy in input spikes while obtaining excellent performance.

The RSM method, illustrated in Fig. 2(c-e), randomly generates a binary mask with the same shape as the input spike matrix based on the masking ratio. This determines

Dataset	Model	Method	Achitecture	# Param (M)	Time Steps	Accuracy (%)
CIFAR-10	ANN [29]	Direct Training	Swin-T (BN)	27.6	/	98.14
	Diet-SNN [39]	Direct Training	VGG-16	39.9	10	93.44
	tdBN [56]	Direct Training	ResNet-19	12.6	4	92.92
	TET [6]	Direct Training	ResNet-19	12.6	6	94.50
	DSR [32]	Direct Training	PreActResNet-18	11.2	20	95.40
	Spikformer [57]	Direct Training	Spikformer-4-384	9.3	4	95.51
	RMP [13]	ANN-to-SNN	VGG-16	39.9	2048	93.63
	RNL [7]	ANN-to-SNN	PreActResNet-18	11.2	256	93.45
	QCFS [2]	ANN-to-SNN	RestNet-18	11.7	512	96.06
	MST (ours)	ANN-to-SNN	Swin-T (BN)	27.6	64	**96.32**
					128	**97.06**
					256	**97.27**
CIFAR-100	ANN [29]	Direct Training	Swin-T (BN)	27.6	/	88.72
	Diet-SNN [39]	Direct Training	VGG-16	39.9	5	69.67
	tdBN [56]	Direct Training	ResNet-19	12.6	4	70.86
	TET [6]	Direct Training	ResNet-19	12.6	6	74.72
	DSR [32]	Direct Training	PreActResNet-18	11.2	20	78.50
	Spikformer	Direct Training	Spikformer-4-384	9.3	4	78.21
	RMP [13]	ANN-to-SNN	VGG-16	39.9	2048	70.93
	RNL [7]	ANN-to-SNN	PreActResNet-18	11.2	256	75.10
	QCFS [2]	ANN-to-SNN	RestNet-18	11.7	512	79.61
	MST (ours)	ANN-to-SNN	Swin-T (BN)	27.6	64	**85.40**
					128	**86.73**
					256	**86.91**
ImageNet	ANN [29]	Direct Training	Swin-T (BN)	28.5	/	80.51
	Diet-SNN [39]	Direct Training	VGG-16	39.9	5	69.00
	tdBN [56]	Direct Training	SEW-ResNet-34	21.8	4	67.04
	TET [6]	Direct Training	SEW-ResNet-34	21.8	4	68.00
	DSR [32]	Direct Training	PreActResNet-18	11.2	50	67.74
	Spikformer [57]	Direct Training	Spikformer-8-768	66.3	4	74.81
	RMP [13]	ANN-to-SNN	VGG-16	39.9	2048	73.09
	QCFS [2]	ANN-to-SNN	RestNet-18	11.7	1024	74.32
	MST (ours)	ANN-to-SNN	Swin-T (BN)	28.5	128	**77.88**
					256	**78.37**
					512	**78.51**

Table 1. Performance comparison between the proposed model and the SOTA models on different static datasets, where Swin-T (BN) refers to the self-implemented ANN baseline replacing LN with BN [29].

which spikes are transmitted for calculation in subsequent neurons, enabling high matrix sparsity. With a masking ratio of 50%, for example, each spike has a 50% chance of being masked, reducing the number of spikes and power consumption. Hence, combining both the SNN-based self-attention mechanism with the RSM approach promises a balance between performance and energy efficiency.

4. Experiments

We conduct extensive experiments on both static datasets including CIFAR-10 [22], CIFAR-100 [21], and ImageNet datasets [5], and neuromorphic datasets including CIFAR10-DVS [24], N-Caltech101 [37], N-Cars [43], ActionRecognition [33], and ASL-DVS [1] datasets, to validate the effectiveness of the MST model. In addition, we evaluate the effect of the RSM approach on accuracy and energy efficiency using the SpikingJelly framework [9]. More details of the training can be found in the supplementary.

4.1. Performance on Static Datasets

Tab. 1 presents a comprehensive comparison of the MST model with the current SOTA SNN models on the CIFAR-10/100 and ImageNet datasets. The results show that the

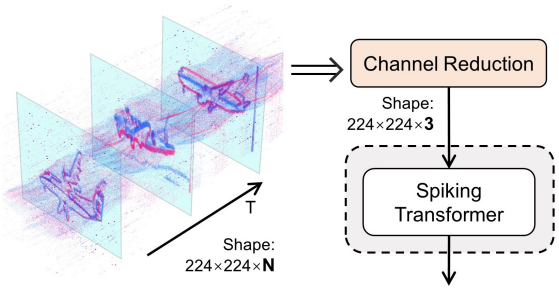

Figure 4. Model Architecture for Neuromorphic Datasets.

proposed MST model outperforms all other models in terms of top-1 accuracy on all three datasets.

Direct training models, such as Diet-SNN [39], tdBN [56], TET [6], and DSR [32], require lower time steps, but their top-1 accuracy is relatively low, compared to that of the MST model, in which tdBN [56] is 10% less accurate on the CIFAR-100 and ImageNet datasets.

Conversely, models employing the ANN-to-SNN conversion method typically necessitate a greater number of time steps to reach optimal performance, including RMP [13], RNL [7], and QCFS [2]. The MST model also employs this method but requires only 64 time steps to achieve superior performance, which is far fewer than RMP [13]. This suggests that the MST model can achieve the highest accuracy in a reasonable number of time steps.

Compared to transformer-based SNN models such as Spikformer [57], the MST model achieves higher top-1 accuracy on all three datasets. Specifically, the top-1 accuracy of the MST model is 1.8%, 8.7%, and 3.7% higher than Spikformer on the CIFAR-10, CIFAR-100, and ImageNet datasets, respectively. Moreover, the MST model has significantly fewer parameters compared to the 8-layer Spikformer [57] model.

4.2. Performance on Neuromorphic Datasets

We showcase the suitability of the MST model for processing event-based data by evaluating its performance on neuromorphic datasets. Fig. 4 shows the framework for the MST model to process the neuromorphic datasets. We utilize a frame-based representation for preprocessing, where the event streams are transformed into a sequence of high-rate frames. Each event in the stream consists of four dimensions, including two spatial coordinates (x, y), timestamp, and polarity. The frames are integrated into an input tensor of size (n×C, H, W), where n is the number of frames, C is the number of original channels (which equals two, representing polarity), and H and W represent the height and width of the input, respectively. To align the input dimension with the model, an additional reduction layer is added to the first layer of the entire model, which reduces the channel dimension to 3. This enables us

Dataset	Model	Time Steps	Accuracy (%)
CIFAR10-DVS	Swin-T (BN)	/	88.98
	TA-SNN [52]	10	72.00
	PLIF [11]	20	74.80
	Dspkie [26]	10	75.40
	DSR [32]	10	77.30
	TET [6]	10	83.17
	NDA [27]	10	81.70
	Spikformer [57]	10	80.90
	MST (ours)	128	**86.60**
		256	**87.20**
		512	**88.12**
N-CALTECH101	Swin-T (BN)	/	92.00
	SALT [19]	20	55.00
	NDA [27]	10	83.70
	MST (ours)	64	**84.71**
		128	**89.42**
		256	**91.38**
N-CARS	Swin-T (BN)	/	97.14
	CarSNN [48]	10	86.00
	NDA [27]	10	91.90
	MST (ours)	32	**94.67**
		64	**96.58**
		128	**97.28**
Action Recognition	Swin-T (BN)	/	90.14
	STCA [12]	10	71.20
	Mb-SNN [28]	10	78.10
	MST (ours)	64	**84.76**
		128	**86.92**
		256	**88.21**
ASL-DVS	Swin-T (BN)	/	99.90
	Meta-SNN [45]	100	96.04
	MST (ours)	64	**98.04**
		128	**98.51**
		256	**99.10**

Table 2. Performance comparison between the proposed model and the SOTA models on different neuromorphic datasets.

to leverage pre-training weights from the ImageNet dataset, accelerating the training convergence. Data augmentation for SNNs [27] is also applied to improve the accuracy. The ensuing experimental results demonstrate that the proposed fine-tuning approach for ANN-to-SNN conversion achieves remarkably high accuracy.

The experimental results presented in Tab. 2 demonstrate the effectiveness of the proposed MST model in processing neuromorphic datasets. We compare the MST model with several SOTA SNN models on five popular neuromorphic datasets, including CIFAR10-DVS, N-Caltech101, N-Cars, Action Recognition, and ASL-DVS dataset. The experimental results show that the MST model achieves the highest top-1 accuracy on all datasets.

CIFAR10-DVS, N-Caltech101, and N-Cars datasets are

constructed by converting the static datasets into event data by using event-based cameras. Tab. 2 shows that the MST model outperforms other SOTA SNN models significantly, achieving improvements of 4.95%, 7.68%, and 5.38% on CIFAR10-DVS, N-Caltech101, and N-Cars, respectively. The Action Recognition dataset consists of a series of human actions captured by event-based cameras. By adopting the data preprocessing method of [20], our MST model achieves a top-1 accuracy of 88.21%, which is far higher than other models. Additionally, the ASL-DVS dataset is a large 24-class dataset of gestures and the experimental results demonstrate that our MST model outperforms other SOTA models by 3.06%.

These results highlight the effectiveness of our proposed MST model in processing various types of neuromorphic datasets.

4.3. Effectiveness of the Random Spike Masking Method

The ANN-to-SNN conversion method typically involves a large number of time steps, leading to high power consumption. To address this issue, we propose the RSM method for spike pruning, and we demonstrate its effectiveness in the following section.

Drawing inspiration from knowledge distillation [16], we use the model without masked as a teacher model, and the model after masked as a student model. By fine-tuning the student model, the masked model achieves high performance comparable to the original model without masking in terms of accuracy. This finding highlights the efficacy of the knowledge distillation technique in training masked models without compromising their performance. Consequently, we employ the fine-tuning approach as the training method for our subsequent comparative analysis.

We evaluate the RSM method in two critical modules in the Transformer: the self-attention (SA) module and the Multi-layer Perceptron (MLP) module. As shown in Fig. 2(d-e), we apply the RSM method to the Query, Key, and Attention matrices in the SA module, as well as the output spike matrix of the first fully connected layer within each block of the MLP module.

Fig. 5(a) and (b) depict the variation in accuracy for different masking ratios on the CIFAR-10 and CIFAR-100 datasets, respectively. The experimental results demonstrate that the accuracy decreases with the increasing masking ratio for both the SA and MLP modules, but their sensitivity to masking ratio changes differs. Specifically, the accuracy of the SA module remains stable over a certain range of masking ratios. In contrast, the accuracy of the MLP module declines more sharply and is more sensitive to masking ratio changes. Consistent with prior research [34, 49], the redundancy in the SA module enables the MST to withstand the losses caused by missing input spikes without significantly affecting the overall performance of the SNN.

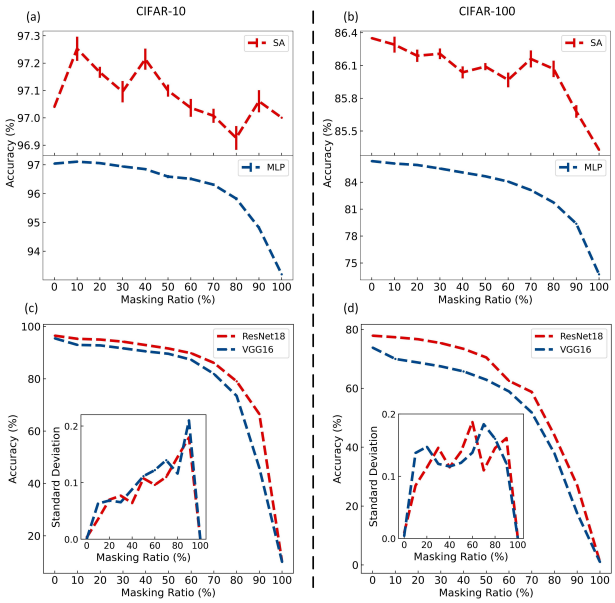

Figure 5. The effectiveness of the RSM method in (a) SA and (b) MLP modules of the MST model, as well as in Spiking (c) ResNet-18 and (d) VGG-16 models, with varying masking ratios. The inset photographs show the standard deviation of accuracy in 10 runs.

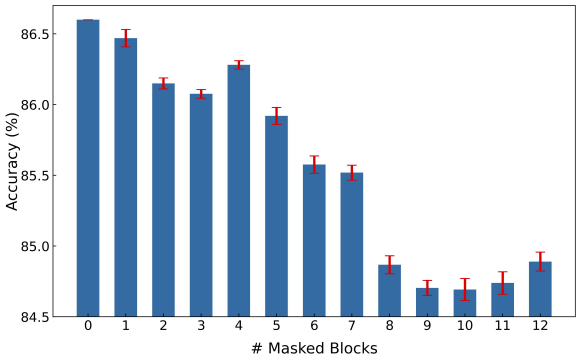

Figure 6. Comparison of the accuracy of masking different numbers of blocks on the CIFAR-100 dataset.

In addition to the variation in sensitivity to changes in the masking ratio between different modules, our experiments also demonstrate that the sensitivity can vary depending on the dataset used. The CIFAR-10 dataset is relatively less sensitive to changes in the masking ratio, with only a 0.1% decrease in accuracy when the masking ratio reaches 80%. In contrast, the CIFAR-100 dataset is more sensitive to changes in the masking ratio, experiencing a more substantial accuracy decline as the masking ratio increases.

We also investigate the potential of the RSM method as a general approach for reducing power consumption in the context of ANN-to-SNN conversion. By applying the RSM

Model	Random Ratio	P (α Watts)	Accuracy (%)
MST	0%	3.9G ($\times$1)	97.27 (+0)
	50%	3.2G ($\times$0.82)	97.25 (-0.02)
	75%	2.9G ($\times$0.74)	97.29 (+0.02)
ResNet-18	0%	58.2M ($\times$1)	96.48 (+0)
	50%	40.7M ($\times$0.70)	92.88 (-3.60)
	75%	34.1M ($\times$0.58)	82.68 (-13.80)
VGG-16	0%	24.4M ($\times$1)	95.46 (+0)
	50%	18.9M ($\times$0.77)	89.56 (-5.90)
	75%	16.7M ($\times$0.68)	79.09 (-16.37)

Table 3. Comparison of power consumption and accuracy between models with a Masking ratio of 0%, 50%, and 75% on the CIFAR-10 dataset, respectively. $(\cdots)$ in the table denotes the power consumption/accuracy compared to the unmasked model (with 0% random ratio).

method to other SNN models like Spiking ResNet-18 and VGG-16 and introducing a masking ratio to each convolution layer, we aim to decrease the transmitted spikes. Our results, presented in Fig. 5, demonstrate that the redundancy of these models can also be leveraged to maintain performance while reducing power consumption within a specific range. This indicates the potential of the RSM method for widespread applications in various SNN models for improving energy efficiency.

Fig. 6 illustrates the impact of masking different numbers of blocks on the accuracy of the MST model. Our results indicate a non-linear, positive relationship between the number of masked blocks and performance loss. Notably, masking the first 2-5 blocks causes a slight performance loss, and the loss increases with more masked blocks until saturation at around 9-10 blocks. These observations provide valuable insights for designing partially masked blocks.

To better evaluate the effects of the RSM method on energy consumption reduction, we utilize theoretical estimates of energy consumption on neuromorphic chips based on previous studies [4, 7]. Assuming that a spike activity consumes α Joules and 1 time step takes 1 ms. Then the power model is defined as:

$$P = \frac{\text{total spikes}}{1 \times 10^{-3}} \times \alpha(\text{Watts}) \tag{10}$$

Tab. 3 presents the power consumption and accuracy of the MST, ResNet-18, and VGG-16 at 0%, 50%, and 75% masking ratios on the CIFAR-10 dataset. The RSM method is applied to the SA module in the MST and each block in ResNet-18 and VGG-16. The results demonstrate that the RSM method has a direct effect on the number of spikes transmitted, which in turn reduces power consumption. In specific, applying the RSM method to the MST

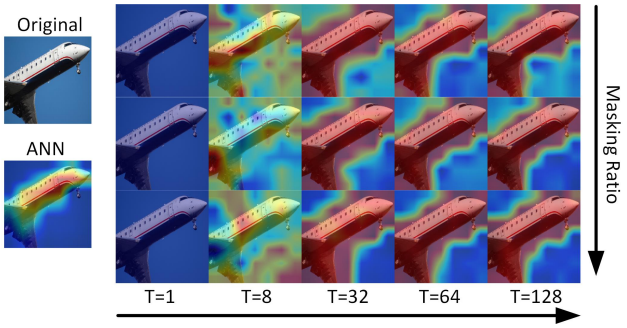

Figure 7. Comparison of attention maps between the MST model with 0%, 50%, and 75% masking ratio (from top to bottom) with 5 different time steps.

model can reduce power consumption by 26.8% without any loss in performance when the masking ratio reaches 75%. This finding suggests that there exist substantial redundant spikes in the SA module that can be pruned for better energy efficiency. As for ResNet-18 and VGG-16, the RSM method also yields significant power reduction. For instance, ResNet-18 can reduce power consumption by 30.1% with a moderate accuracy loss of 3.6%. However, excessive masking leads to a significant drop in accuracy.

In summary, our results demonstrate that the proposed RSM method is effective when applied to the SA module in the Transformer, as well as other SNN models. By eliminating redundant spikes, the RSM method reduces power consumption while preserving performance, making it a promising approach for energy-efficient ANN-to-SNN conversion.

To visualize the impact of different masking ratios on attention maps, we compared the models with 0%, 50%, and 75% masking ratios using the Spike Activation Maps (SAM) method. The results, shown in Fig. 7, demonstrate that the models with different masking ratios concentrate on similar areas of the object at the same time step, with the red parts outlining the object. Comparison with the ANN model using the ScoreCAM method [50] reveals that both models focus on similar key information. These results suggest that the proposed RSM method preserves the regions of interest within the model, contributing to accuracy preservation.

5. Discussion

In this paper, we proposed the Masked Spiking Transformer (MST) with the Random Spike Masking (RSM) method. By pruning input spikes, the proposed RSM method effectively reduces power consumption while maintaining the performance of the model within a certain range.

Though our experiments highlight the superiority of the MST model over state-of-the-art SNN models, our model still has limitations that need addressing. A key constraint is the relatively long time steps needed by the ANN-to-SNN

conversion method, limiting the suitability of the proposed model for real-time applications with strict timing demands.

Furthermore, even though the RSM method reduces the number of spikes and energy consumption, the experiment in this article merely investigates its applicability of ANN-to-SNN conversion, thus exhibiting relatively high energy consumption compared to direct-trained SNN models, which take fewer time steps. Consequently, future research could apply the RSM method to direct training methods to optimize energy consumption further and make SNNs more practical for real-world applications. Additionally, adopting different masking ratios across layers may achieve a better balance between performance and energy efficiency.

6. Conclusion

In this work, we propose a Masked Spiking Transformer (MST) framework that combines the energy efficiency of SNNs with the high-performance self-attention mechanism of Transformers using the ANN-to-SNN method. Additionally, we introduce a Random Spike Masking (RSM) method to prune input spikes, thus reducing power consumption. The experimental results demonstrate that the MST model outperforms current SOTA SNN models on both static and neuromorphic datasets. Furthermore, the proposed RSM method shows significant power reduction while maintaining performance in different modules of the Transformer and other SNN models. Our work opens up new possibilities for developing high-performance SNN models, paving the way for future research in this area.

7. Acknowledgement

This work is supported by the Guangzhou-HKUST(GZ) Joint Funding Program (Grant No. 2023A03J0682), HK RGC (Grant Nos. 27206321, 17205922, 17212923), NSFC (Grant No. 62122004), National Key R&D Program of China (Grant No. SQ2022YFB3600159). This research is also partially supported by ACCESS – AI Chip Center for Emerging Smart Systems, sponsored by Innovation and Technology Fund (ITF), Hong Kong SAR.

References

[1] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019.

[2] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. In *International Conference on Learning Representations*, 2021.

[3] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

[4] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[6] Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

[7] Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*, 2021.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Timothée Masquelier, Yonghong Tian, and other contributors. Spikingjelly. `https://github.com/fangwei123456/spikingjelly`, 2020. Accessed: 2023-02-21.

[10] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

[11] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée 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*, pages 2661–2671, 2021.

[12] Pengjie Gu, Rong Xiao, Gang Pan, and Huajin Tang. STCA: Spatio-Temporal Credit Assignment with Delayed Feedback in Deep Spiking Neural Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 1366–1372, Macao, China, Aug. 2019. International Joint Conferences on Artificial Intelligence Organization.

[13] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13558–13567, 2020.

[14] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021.

[15] 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.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, Mar. 2015.

[17] Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[18] Youngeun Kim and Priyadarshini Panda. Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Networks*, 144:686–698, 2021.

[19] Youngeun Kim and Priyadarshini Panda. Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Networks*, 144:686–698, 2021.

[20] Karthik Sivarama Krishnan and Koushik Sivarama Krishnan. Benchmarking Conventional Vision Models on Neuromorphic Fall Detection and Action Recognition Dataset. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0518–0523. IEEE, 2022.

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov. 1998.

[23] William B. Levy and Robert A. Baxter. Energy-Efficient Neuronal Computation via Quantal Synaptic Failures. *Journal of Neuroscience*, 22(11):4746–4755, June 2002.

[24] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11, 2017.

[25] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning*, pages 6316–6325. PMLR, 2021.

[26] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34:23426–23439, 2021.

[27] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic Data Augmentation for Training Spiking Neural Networks. *arXiv preprint arXiv:2203.06145*, 2022.

[28] 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*, pages 1743–1749, Montreal, Canada, Aug. 2021. International Joint Conferences on Artificial Intelligence Organization.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

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

[31] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[32] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12444–12453, 2022.

[33] Shu Miao, Guang Chen, Xiangyu Ning, Yang Zi, Kejia Ren, Zhenshan Bing, and Alois Knoll. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Frontiers in neurorobotics*, 13:38, 2019.

[34] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[35] Etienne Mueller, Viktor Studenyak, Daniel Auge, and Alois Knoll. Spiking Transformer Networks: A Rate Coded Approach for Processing Sequential Data. In *2021 7th International Conference on Systems and Informatics (ICSAI)*, pages 1–5, Nov. 2021.

[36] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36(6):51–63, Nov. 2019.

[37] Garrick Orchard, Ajinkya Jayawant, Gregory K. Cohen, and Nitish Thakor. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. *Frontiers in Neuroscience*, 9, 2015.

[38] Wachirawit Ponghiran and Kaushik Roy. Spiking neural networks with improved inherent recurrence dynamics for sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8001–8008, 2022.

[39] Nitin Rathi and Kaushik Roy. DIET-SNN: Direct Input Encoding With Leakage and Threshold Optimization in Deep Spiking Neural Networks, Dec. 2020.

[40] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*, 2020.

[41] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

[42] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, Apr. 2015.

[43] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[45] Kenneth M. Stewart and Emre O. Neftci. Meta-learning spiking neural networks with surrogate gradient descent. *Neuromorphic Computing and Engineering*, 2(4):044002, 2022.

[46] Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, 2021.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[48] Alberto Viale, Alberto Marchisio, Maurizio Martina, Guido Masera, and Muhammad Shafique. Carsnn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2021.

[49] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[50] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[51] Zheyu Yang, Yujie Wu, Guanrui Wang, Yukuan Yang, Guoqi Li, Lei Deng, Jun Zhu, and Luping Shi. DashNet: A hybrid artificial and spiking neural network for high-speed object tracking. *arXiv preprint arXiv:1909.12942*, 2019.

[52] 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*, pages 10221–10230, 2021.

[53] Zhuliang Yao, Yue Cao, Yutong Lin, Ze Liu, Zheng Zhang, and Han Hu. Leveraging batch normalization for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 413–422, 2021.

[54] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking Transformers for Event-Based Single Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8801–8810, 2022.

[55] Jiyuan Zhang, Lulu Tang, Zhaofei Yu, Jiwen Lu, and Tiejun Huang. Spike Transformer: Monocular Depth Estimation for Spiking Camera. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 34–52. Springer, 2022.

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

[57] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When Spiking Neural Network Meets Transformer. *arXiv preprint arXiv:2209.15425*, 2022.

[58] Michael Zhu and Suyog Gupta. To prune, or not to prune: Exploring the efficacy of pruning for model compression, Nov. 2017.