Towards Efficient Spiking Transformer: a Token Sparsification Framework for Training and Inference Acceleration

Zhengyang Zhuge [1 2] **Peisong Wang** [1 2 3] **Xingting Yao** [1 4] **Jian Cheng** [1 2 3]

Abstract

Nowadays Spiking Transformers have exhibited remarkable performance close to Artificial Neural Networks (ANNs), while enjoying the inherent energy-efficiency of Spiking Neural Networks (SNNs). However, training Spiking Transformers on GPUs is considerably more time-consuming compared to the ANN counterparts, despite the energy-efficient inference through neuromorphic computation. In this paper, we investigate the token sparsification technique for efficient training of Spiking Transformer and find conventional methods suffer from noticeable performance degradation. We analyze the issue and propose our Sparsification with Timestep-wise Anchor Token and dual Alignments (STATA). Timestep-wise Anchor Token enables precise identification of important tokens across timesteps based on standardized criteria. Additionally, dual Alignments incorporate both Intra and Inter Alignment of the attention maps, fostering the learning of inferior attention. Extensive experiments show the effectiveness of STATA thoroughly, which demonstrates up to $\sim$1.53$\times$ training speedup and $\sim$48% energy reduction with comparable performance on various datasets and architectures.

1. Introduction

Spiking Neural Networks (SNNs), considered the third generation of Neural Networks (Maass, 1997), hold immense promise due to their low power consumption, event-driven nature, and alignment with biological principles (Roy et al., 2019). The neurons in SNNs, inspired by biological counterparts, produce sparse and discrete events through the emission of binary spikes, facilitating communication with post-synaptic neurons (Krestinskaya et al., 2019). Due to the event-driven nature, SNNs offer a distinct advantage over Artificial Neural Networks (ANNs) in terms of energy efficiency (Furber et al., 2014; Merolla et al., 2014; Pei et al., 2019), which becomes particularly beneficial to edge computing scenarios where resources are constrained.

Recently, the pioneer Spiking Transformer (Zhou et al., 2023b) successfully introduced the Transformer-style architecture design to SNNs. Following this, there has been a growing emergence of Spiking Transformers (Yao et al., 2023a; Zhou et al., 2023a), gradually narrowing the performance gap between Spiking Transformers and the ANN counterparts (Dosovitskiy et al., 2020) on various datasets. However, despite the energy-efficient inference achieved through neuromorphic computation, the training of Spiking Transformer is significantly more time-consuming compared to its corresponding ANN-Transformer counterpart due to the incorporation of an additional temporal dimension. For instance, training a Spiking Transformer on ImageNet requires even thousands of GPU hours, demanding substantial computational resources and time. In addition to the inefficient training, these Spiking Transformers always rely on larger and more complex models to attain superior task performance, resulting in increased energy consumption during inference compared to traditional SNNs.

To enhance the energy and computational efficiency in SNNs, researchers have made some efforts through various methods. One of the most effective approaches to enhance computational efficiency is sparsification, commonly known as pruning (Chen et al., 2022; Yin et al., 2023). Nevertheless, existing pruning methods for SNNs primarily focus on enhancing inference efficiency, while paying less attention to the efficiency of the training process. Some pruning works (Chen et al., 2021c; Kim et al., 2022) typically require a considerable amount of additional training time and iterations to obtain the pruned network, which can even result in increased training overhead. Other sparsification works utilize unstructured pruning to achieve greater energy savings for the inference phase. However, unstructured sparsification

[1]Institute of Automation, Chinese Academy of Sciences [2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]AiRiA [4]School of Future Technology, University of Chinese Academy of Sciences. Correspondence to: Peisong Wang <peisong.wang@nlpr.ia.ac.cn>, Jian Cheng <jcheng@nlpr.ia.ac.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

relies on specific hardware to achieve efficient computation, thereby limiting its ability to accelerate the training process of SNNs on most GPUs. This raises a crucial question: *Can sparsification techniques accelerate the training of a high-performance Spiking Transformers?*

In this paper, we initially evaluate several conventional metrics for token sparsification within the current Spiking Transformer framework. However, we find that these metrics inevitably result in performance degradation. We analyze that the performance degradation mainly stems from two factors: 1. Failure to capture semantic information. 2. Lack of standardized and consistent evaluation of importance. To tackle these challenges, we propose the approach called Sparsification with Timestep-wise Anchor Token and dual Alignments (STATA), which introduces the concept of the Anchor Token for objective identification of the informative tokens during token sparsification of Spiking Transformer. Then we utilize the attention information of Anchor Token across different timesteps to determine whether each token should be pruned or retained. Specifically, the computational cost of the additional Anchor Token can be negligible compared with that of numerous original tokens. Moreover, we directly obtain the importance rankings of tokens based on the inherent computations of the Self-Attention module, resulting in minimal costs. Then, to tackle the disparities in attention map quality and foster the improvement of inferior attention maps, we propose dual Alignments, which incorporate both Intra-Alignment and Inter-Alignment to enhance the poor attention across timesteps, heads, and layers. Finally, our STATA not only demonstrates efficiency in both training and inference stages of the Spiking Transformer, but also maintains a comparable level of performance.

The main contributions of this work are threefold:

- We explore various token sparsification metrics in Spiking Transformers, then introduce our Timestep-wise Anchor Token for accurate token sparsification based on attention, mitigating the sharp performance drop.
- To further enhance the pruning accuracy of inferior attention in terms of timestep, head, and layer, we propose dual Alignments, which include Intra-Alignment and Inter-Alignment of attention maps for training.
- Extensive experiments and ablation studies demonstrate the efficacy of our STATA in enhancing the training and inference efficiency of various Spiking Transformers while maintaining competitive performance.

2. Related Work

2.1. Spiking Neural Networks

The key distinction between Spiking Neural Networks (SNNs) and traditional Artificial Neural Networks (ANNs) lies in their utilization of discrete spikes instead of continuous decimal values for information processing and transmission. Spikes are typically generated by spike neurons, such as Izhikevich neuron (Izhikevich, 2003) and Leaky Integrate-and-Fire (LIF) neuron (Wu et al., 2018). Due to the non-differentiability of spikes, the performance of SNNs is influenced to some extent. To enhance performance, a lot of works lift the performance of SNNs by incorporating advanced architectures from ANNs, such as ResNet-like SNNs (Hu et al., 2021a; Fang et al., 2021a; Zheng et al., 2021; Hu et al., 2021b; Yao et al., 2023b), Spiking RNNs (Lotfi Rezaabad & Vishwanath, 2020) and Spiking GNNs (Zhu et al., 2022b).

2.2. Spiking Transformers

Recently, there has been a growing interest in Transformer-like SNN, i.e. Spiking Transformer. Spikeformer(Li et al., 2022) proposed to combine the architecture of Transformer to SNNs, but this model is not a pure SNN due to the presence of numerous floating-point multiplication, division, and exponential operations, which are not suitable for neuromorphic computation. Spikformer (Zhou et al., 2023b) first introduces the innovative Spiking Self Attention (SSA), which achieves energy-efficient self-attention in SNNs and shows good performance with low energy consumption. Subsequently, Spike-driven Transformer (Yao et al., 2023a) further enhances the efficiency of Spiking Self Attention and rearranges residual connections to ensure spikes are binary. Additionally, Spikingformer (Zhou et al., 2023a) improves upon the full binary spike architecture, enhancing gradient backpropagation and achieving higher performance.

2.3. Sparsification of SNNs

Numerous approaches have been developed to enhance the efficiency of neural networks, including sparsification(He et al., 2018; Zhao et al., 2022), quantization(Wang et al., 2019; Chen et al., 2021b; Yue et al., 2022; Xiao et al., 2023), and distillation(Chen et al., 2023a; Guo et al., 2023), etc. Among these, sparsification provides a prospect of mitigating the computing and storage overhead in neural networks. It can be applied to various components, such as weights(Xu et al., 2022), data(Sorscher et al., 2022; Zhuge et al., 2024), and gradients(Perez-Nieves & Goodman, 2021). Recently, many studies have investigated the sparsification of SNNs to enhance their efficiency. Current research mainly focuses on the sparsity of SNN weights. Notable approaches include using the magnitude-based method (Yin et al., 2021; Chen et al., 2021c), integrating classic optimization tools with the SNN training method (Deng et al., 2021a), removing weak weights based on threshold (Chen et al., 2023b), and exploring the lottery ticket hypothesis in SNNs(Kim et al., 2022), etc. However, current sparsification methods of SNNs primarily concentrate on enhancing inference energy efficiency,

while paying less attention to the time-consuming training process on GPU caused by the introduction of temporal dimension in SNN. Some pruning methods (Chen et al., 2021c; Kim et al., 2022) even require extended training time and iterations to attain pruned networks, resulting in significant additional training costs.

3. Preliminaries

Spiking Neuron As the basic unit of SNNs, the spiking neuron receives the resultant current and accumulates membrane potential. This potential is then compared with the threshold to determine whether a spike should be generated. In our work, we consistently use LIF spike neurons. The dynamic model of LIF is described as:

$$H[t] = V[t-1] + \frac{1}{\tau}\left(X[t] - (V[t-1] - V_{\text{reset}})\right), \quad (1)$$

$$S[t] = \Theta\left(H[t] - V_{th}\right), \quad (2)$$

$$V[t] = H[t](1 - S[t]) + V_{\text{reset}}\, S[t], \quad (3)$$

where τ is the membrane time constant, and $X[t]$ is the input current at time step t. When the membrane potential $H[t]$ exceeds the firing threshold V_{th}, the spike neuron will trigger a spike $S[t]$. $\Theta(v)$ is the Heaviside step function which equals 1 for $v \geq 0$ and 0 otherwise. $V[t]$ represents the membrane potential after the trigger event which equals $H[t]$ if no spike is generated, and otherwise equals to the reset potential V_{reset} .

Spiking Transformer As an emerging SNN model, the Spiking Transformer first process the input image I by:

$$X = \text{MP}(\text{SN}(\text{BN}((\text{Conv2d}(I))))) \quad (4)$$

where $I \in \mathbb{R}^{T\times C\times H\times W}$, T, C, H, and W refer to the timesteps, channels, height, and width, respectively. The BN, Conv2d and MP represent the batch normalization layer, 2D convolution layer and max-pooling, respectively. SN refers to the spiking neuron. After this processing , I is split and transformed into an image patches sequence $X \in \mathbb{R}^{T\times N\times D}$. Then the patches sequence are used to compute query (Q), key (K), and Value (V) through the corresponding linear transform and spiking neuron as follows:

$$Q = \text{SN}_Q\left(\text{BN}\left(XW_Q\right)\right), \quad (5)$$

$$K = \text{SN}_K\left(\text{BN}\left(XW_K\right)\right), \quad (6)$$

$$V = \text{SN}_V\left(\text{BN}\left(XW_V\right)\right), \quad (7)$$

where $Q, K, V \in \mathbb{R}^{T\times N\times D}$. W_Q, W_K, W_V are corresponding learnable weights for query, key and value. SN_Q, SN_K, SN_V are spiking neurons for Q, K and V, respectively. Then the most fundamental component of Spiking Transformer, i.e. Spiking Self Attention (SSA) is computed as below:

$$\text{SSA}(Q, K, V) = \text{SN}\left(QK^{\text{T}}V * s\right) \quad (8)$$

where s serves as a scaling factor to control the magnitude of the output results as mentioned in Spikformer (Zhou et al., 2023b).

After being processed by the Spiking Transformer encoder, which comprises multiple Spiking Self-Attention (SSA) and Feed-forward Network (FFN) layers, the resultant features are forwarded through a Global Average Pooling (GAP) and a classification head for prediction.

4. Methodology

In this section, we highlight the limitations of several commonly used metrics for token sparsification in the current Spiking Transformer. Then we propose Sparsification with Timestep-wise Anchor Token and dual Alignments (STATA) to efficiently train the Spiking Transformer. Firstly, we introduce the Timestep-wise Anchor Token into the spiking transformer structure, enabling an accurate assessment of the importance among tokens for sparsification. Subsequently, we devise dual Alignments for training, including Intra-Alignment and Inter-Alignment for attention maps, which further enhance the performance.

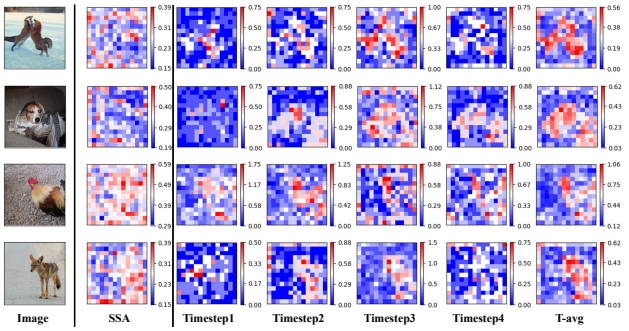

Figure 1. Comparison of the attention maps between Spiking Self Attention (SSA) and our STATA method on ImageNet. We average the value of SSA in original Spiking Transformer to demonstrate the attention maps of SSA. Meanwhile, we demonstrate the attention maps of STATA across different timesteps. Furthermore, T-avg shows the average attention map among timesteps in our STATA. The attention maps obtained from STATA better reflect the foreground regions, which is crucial for identifying and preserving the informative tokens in sparsification.

4.1. How to prune tokens in Spiking Transformer

In the current Spiking Transformers, images are split into tokens with multiple timesteps. These tokens are then fed into a sequence of Spiking Self Attention (SSA) and Feed Forward Networks (FFN) for processing. A direct strategy to

accelerate the training process of the Spiking Transformer is reducing the number of tokens. However, identifying which tokens to be pruned among multiple timesteps poses a non-trivial challenge within the current Spiking Transformer framework. As illustrated in Table 1, we have tried several pruning metrics for token sparsification in Spiking Transformer (Zhou et al., 2023a) on CIFAR100. We perform token sparsification directly based on the ranking scores derived from these metrics, and maintain their training overhead comparable. Nevertheless, these metrics result in noticeable performance degradation.

The performance degradation primarily arises from the inability of these metrics to effectively identify and retain important tokens in the training of the Spiking Transformer. Additionally, We attribute this issue to two factors:

(a). Failure to capture semantic information. We consider the foreground region of an image to be important, as it always carries abundant semantic information and category-specific details. However, random sampling fails to distinguish between the foreground and background, resulting in the loss of numerous important tokens. Moreover, the l_1-norm metric is also inadequate in capturing semantic information and identifying the foreground region.

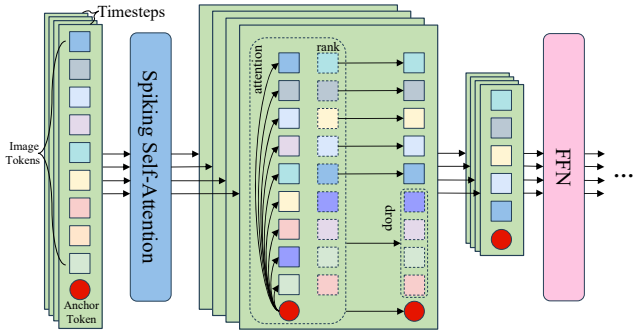

Figure 2. Illustration of Timestep-wise Anchor Token for token sparsification. This sparsification component is plug-and-play and the computational cost is negligible.

(b). Lack of standardized and consistent criteria of importance. As for Spiking Self Attention (SSA) in the original Spiking Transformer, it possesses the ability to capture semantic information. However, it may still not serve as a good criterion. This is because the values of Self Attention are influenced by the similarity between tokens. Consequently, tokens located in the background region may exhibit high attention values if there are numerous other tokens in the background region that are similar to them. This phenomenon is illustrated in Figure 1, where the attention maps of SSA fail to accurately identify the foreground. In the SSA-based method, we recognize that the importance score of each image token is determined collectively by the other image tokens. It is evident that different tokens employ inconsistent criteria when assigning scores. Hence, there is a need for a standardized scoring method to ensure consistency.

Token Pruning Metric	Top-1 Accuracy (%)	Training Time Overhead
Original	79.98	100%
Random	78.35	~67%
l_1-norm	78.51	~67%
SSA	78.47	~67%

Table 1. Different metrics for token sparsification in the current Spiking Transformers all lead to significant performance degradation. Original: the original model without any pruning. Random: randomly pruning token across timesteps. l_1-norm: applying l_1-norm to rank and prune tokens. SSA: ranking and pruning tokens based on the averaged value of Spiking Self Attention (SSA) across all image tokens.

4.2. Sparsification with Timestep-wise Anchor Token

To consistently compare each token across different timesteps, we introduce the timestep-wise anchor token to accurately extract important foreground regions across each timestep. Under the Spiking Transformer (Zhou et al., 2023b) paradigm, we denote the input image as $I \in \mathbb{R}^{T\times C\times H\times W}$. The image is first split to several tokens $X \in \mathbb{R}^{T\times N\times D}$, where $N = p^2, D = C \times \frac{H}{p} \times \frac{W}{p}$. Then we add our Anchor Token at the beginning position to obtain new patches sequence $X \in \mathbb{R}^{T\times(N+1)\times D}$. Then, each token is mapped to one vector through a linear layer with weights $W \in \mathbb{R}^{D\times D'}$, where D' is the hidden dimension. We denote the i-th token in the j-th timestep as X_i^j and introduce the Anchor Token in the j-th timestep to be X_0^j. Then, we can calculate the importance score of the image tokens in timestep j by Anchor Token as follows:

$$A^j = \mathbb{E}_h \left[Q_0^j K_{1:}^{j\top} \cdot s \right] \tag{9}$$

where $\mathbb{E}_h$ means taking the average attention value of multiple heads (Vaswani et al., 2017). Q_0^j means the query vector of the Anchor Token in timestep j, which can be calculated by $Q_0^j = \mathrm{SN}_Q(\mathrm{BN}(X_0^j W_Q))$ and W_Q refers to the learnable weights for the query. $K_{1:}^j$ is the key matrix of timestep j excluding the first row (Anchor Token), which is calculated as Equation (6). s serves as a scaling factor.

As the Anchor Token calculates importance score separately for each timestep, we refer to it as the Timestep-wise Anchor Token (TAT). The main rationale behind the timestep-wise design is rooted in the inherent temporal nature of the Spiking Transformer. Hidden tokens at different timesteps may exhibit distinct characteristics, which may require different treatment. It is also worth noting that these importance

scores can be obtained by directly extracting the spiking self-attention tensor that corresponds to the anchor token in the 0-th row. This process does not incur any additional computational or storage overhead.

After obtaining the importance scores of the tokens, we can selectively divide the tokens into two sets, i.e., informative set $\mathcal{I}$ and non-informative set $\mathcal{N}$, where $\mathcal{I}$ mainly includes the tokens with larger importance score and $\mathcal{N}$ is composed of the remaining tokens. We introduce $\gamma = \frac{|\mathcal{I}|}{|\mathcal{I}|+|\mathcal{N}|}$ to control the sparsity. As depicted in Figure 2, the sparsification module is seamlessly integrated into the Spiking Transformer framework, and we use $\mathbb{P}_I$ to denote the set of insert locations for this module.

4.3. Intra-Alignment of attention

According to Figure 1, it can be observed that there exist noticeable variations in the quality of the attention maps across the timestep dimension. A similar phenomenon also emerges in the head dimension due to the nature of multi-head design (Vaswani et al., 2017). However, excessive variations are not desirable, as inferior attention may fail to accurately identify important tokens. On the one hand, excessive variations across timesteps hinder the effectiveness of the token identification and sparsification process described in Section 4.2. On the other hand, as the timestep-wise attention map is derived by averaging attention across heads, significant disparities among the heads also have a detrimental effect on the timestep-wise attention map, which hampers the accurate sparsification of tokens. To address these issues, we propose Intra-Alignment of attention during the training phase, which uses the superior attention map to supervise the inferior ones. Specifically, the Intra-Alignment consists of two dimensions: timestep and head. In terms of timestep alignment, we first partition the attention maps based on the timestep dimension and sort them according to the l_p-norm (Liu et al., 2017)($p = 2$). Subsequently, we leverage the top half of the attention maps to facilitate the learning of the bottom half ones. Similarly, in terms of head alignment, we partition, sort, and align the attention maps based on the head dimension. Finally, we choose to align the attention maps located in the position set $\mathbb{P}_a$, and the intra-layer alignment loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{intra}} &= \frac{1}{K}\sum_{i=1}^{K}\left(\mathcal{D}(\widetilde{\boldsymbol{\phi}_{\boldsymbol{T}_i}}, \widehat{\boldsymbol{\phi}_{\boldsymbol{T}_i}}) + \mathcal{D}(\widetilde{\boldsymbol{\phi}_{\boldsymbol{H}_i}}, \widehat{\boldsymbol{\phi}_{\boldsymbol{H}_i}})\right) \\ &= \frac{1}{K}\sum_{i=1}^{K}\left(\|\widetilde{\boldsymbol{\phi}_{\boldsymbol{T}_i}} - \widehat{\boldsymbol{\phi}_{\boldsymbol{T}_i}}\|_2 + \|\widetilde{\boldsymbol{\phi}_{\boldsymbol{H}_i}} - \widehat{\boldsymbol{\phi}_{\boldsymbol{H}_i}}\|_2\right),\end{aligned} \tag{10}$$

where $K = |\mathbb{P}_a|$ denotes the number of positions for Intra-Alignment, $\mathcal{D}(\cdot)$ represents the distance function (we utilize l_2 loss in our case), $\widetilde{\boldsymbol{\phi}_{\boldsymbol{T}_i}}$ and $\widehat{\boldsymbol{\phi}_{\boldsymbol{T}_i}}$ refer to the bottom half and the top half attention maps, respectively, based on the timestep dimension. Similarly, $\widetilde{\boldsymbol{\phi}_{\boldsymbol{H}_i}}$ and $\widehat{\boldsymbol{\phi}_{\boldsymbol{H}_i}}$ represent the ones based on the head dimension. Since the number of timesteps and heads is small, it is worth noting that our Intra-Alignment does not significantly increase training time and incurs no additional cost during inference.

Algorithm 1 STATA

Input: Spiking Transformer model $f(\cdot)$, training dataset $\mathcal{T}$, Anchor Token X_0, location sets $\mathbb{P}_I$, $\mathbb{P}_a$, $\mathbb{P}_r$, number of transformer layers L, Timestep T.

for $(x, y) \in \mathcal{T}$ **do**
  $X = patchify(x)$
  for k **in** $0, \ldots, L-1$ **do**
    $X = SSA_k(X)$
    if $k \in \mathbb{P}_I$ **then**
      for j in $1, \ldots, T$ **do**
        ranking tokens according to Equation (9)
      end for
    else if $k \in \mathbb{P}_a$ **then**
      collect attention map $\boldsymbol{\phi}_{\boldsymbol{T}_k}$ and $\boldsymbol{\phi}_{\boldsymbol{H}_k}$ for Intra-Alignment loss $\mathcal{L}_{intra}$ in Equation (10)
    else if $k \in \mathbb{P}_r$ **then**
      collect attention map $\boldsymbol{\phi}_k$ for computing Inter-Alignment loss $\mathcal{L}_{inter}$ in Equation (11)
    end if
    $X = FFN_k(X)$
  end for
  obtain final loss as Equation (12) to backpropagate
end for

Output: Token pruned Spiking Transformer $f_p(\cdot)$

4.4. Inter-Alignment of attention

In addition to the quality variations within the attention map of a layer, there are also variations in the attention maps across different layers. Inspired by (Wolchover, 2018; Chen et al., 2021a), which suggests that deep features always capture more semantic visual concepts than shallow ones, we introduce the attention alignment among layers, which utilizes the attention maps derived from deep layers to enhance the attention ability of shallow layers. We select the attention maps of layers located in the position set $\mathbb{P}_r$, and the loss of Inter-Alignment among Layers can be written as:

$$\begin{aligned}\mathcal{L}_{\text{inter}} &= \frac{1}{\Gamma}\sum_{i\in\mathbb{P}_r}\sum_{j\in\mathbb{P}_r, j>i}\mathcal{D}\left(T_i\left(\boldsymbol{\phi}_{\boldsymbol{i}}\right), T_j\left(\boldsymbol{\phi}_{\boldsymbol{j}}\right)\right) \\ &= \frac{1}{\Gamma}\sum_{i\in\mathbb{P}_r}\sum_{j\in\mathbb{P}_r, j>i}\left\|T_i\left(\boldsymbol{\phi}_{\boldsymbol{i}}\right) - T_j\left(\boldsymbol{\phi}_{\boldsymbol{j}}\right)\right\|_2,\end{aligned} \tag{11}$$

where Γ denotes the number of pair loss, l_2 distance is used for function $\mathcal{D}(\cdot)$, $\boldsymbol{\phi}$ refers to the attention map and the transformation function T is utilized to process the attention maps. Specifically, interpolation is employed to

Methods	Architecture	OPs (G)	Energy (mJ)	TimeStep	Acc (%)
Hybrid training (Rathi et al., 2020)	ResNet-34	-	-	250	61.48
TET (Deng et al., 2021b)	SEW-ResNet-34	-	-	4	68
Spiking ResNet (Hu et al., 2021a)	ResNet-34	65.28	59.295	350	71.61
	ResNet-50	78.29	70.934	350	72.75
STBP-tdBN (Zheng et al., 2021)	Spiking-ResNet-34	6.5	6.393	6	63.72
SEW ResNet (Fang et al., 2021a)	SEW-ResNet-50	4.83	4.89	4	67.78
	SEW-ResNet-101	9.3	8.913	4	68.76
	SEW-ResNet-152	13.72	12.891	4	69.26
Spikformer(Zhou et al., 2023b)	Spikformer-8-768	22.09	21.48	4	74.81
Random Token	Spikformer-8-768	10.62	11.11	4	70.13
SSA-based	Spikformer-8-768	10.64	11.12	4	70.45
STATA (ours)	Spikformer-8-768	10.70	11.16	4	74.03

Table 2. Experiments on ImageNet. We compared STATA with other pruning-based training acceleration strategies, all of which maintain the similar acceleration ratios. OPs refers to Synaptic Operations (SOPs) in SNN and Floating-point Operations (FLOPs) in ANNs. Energy represents the average theoretical energy consumption.

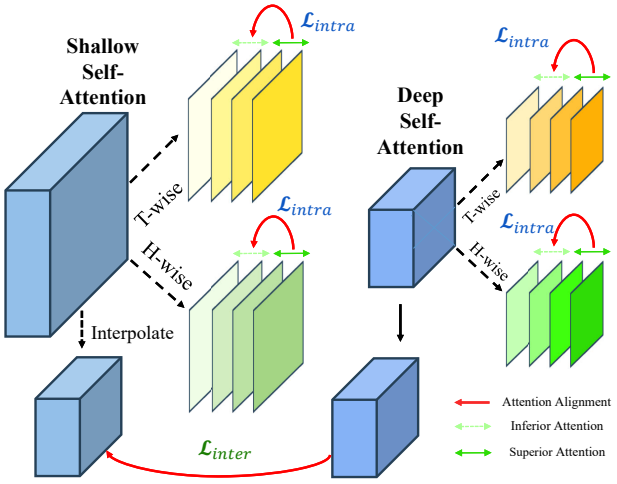

Figure 3. Illustration of dual Alignments which include Intra-Alignment and Inter-Alignment of attention maps. To illustrate, we simply depict two layers' attention, labeled as Shallow Self-Attention (in the shallow layer) and Deep Self-Attention (in the deep layer). We illustrate the item used for calculating the Intra-Alignment loss $\mathcal{L}_{intra}$ and Inter-Alignment loss $\mathcal{L}_{inter}$. T-wise means Timestep-wise division, H-wise means head-wise division.

align and match the attention maps of different sizes. It is important to note that ϕ_j in Equation (11) is detached during back-propagation. Due to the number of related layers and the dimension of the attention maps are small, the computational burden associated with this loss term can be disregarded. Finally, we employ the following loss to train the token-pruned Spiking Transformer:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}\left(f_\theta; x, y\right) + \alpha\mathcal{L}_{\text{intra}} + \beta\mathcal{L}_{\text{inter}}\,, \qquad (12)$$

where $\mathcal{L}_{\text{CE}}$ represents cross entropy loss, f_θ refers to the Spiking Transformer model f with parameter θ. α and β are hyper-parameters. x and y are sample and label, respectively. By employing the aforementioned design, we can efficiently train the Spiking Transformer as Algorithm 1 and subsequently obtain a token-pruned Spiking Transformer for efficient inference.

5. Experiments

5.1. Setups

Models and datasets We evaluate our method, Sparsification with Timestep-wise Anchor Token and dual Alignments (STATA), using three Spiking Transformers: Spikformer (Zhou et al., 2023b), Spike-driven Transformer (Yao et al., 2023a) and Spikingformer (Zhou et al., 2023a). In certain cases, we denote the corresponding Spiking Transformer with B blocks and D dimension as ST-B-D. As for datasets, to demonstrate the superior performance of our method in terms of accuracy, training costs, and inference efficiency, we conduct experiments on various datasets including static datasets CIFAR-10/100 (Krizhevsky, 2009), ImageNet (Deng et al., 2009) and neuromorphic datasets CIFAR10-DVS (Li et al., 2017) and DVS128 Gesture (Amir et al., 2017).

Implementation The experimental setup follows (Zhou et al., 2023b). For the ImageNet dataset, the size of the input image is set to 224 × 224, and the image is divided into 196 patches. We utilize the AdamW optimizer and perform training on 8 GPUs, with a batch size of 128, over a total of 310 training epochs. Standard data augmentation techniques, such as random augmentation, mixup, are also

employed during the training process. As for CIFAR10/100, the input size is set to 32×32, and each image is divided into 64 patches. We conduct the training on a single GPU using a batch size of 192. The timestep is set to 4 and the training epoch is 310, which are consistent with that of ImageNet. For neuromorphic datasets CIFAR10-DVS and DVS128 Gesture, the image size is set to 128×128, the patch size is 16×16, and the timestep is 16. For token sparsification, we insert the token sparsification module every 3 blocks, beginning from the block-$(L/3)$, where L is the number of layers. The alignment location sets $\mathbb{P}_a$ and $\mathbb{P}_r$ are defined as $\{0, L/3+1, L-1\}$. The factor α and β are set to $1e-2$. All experiments are conducted on RTX-3090 GPUs

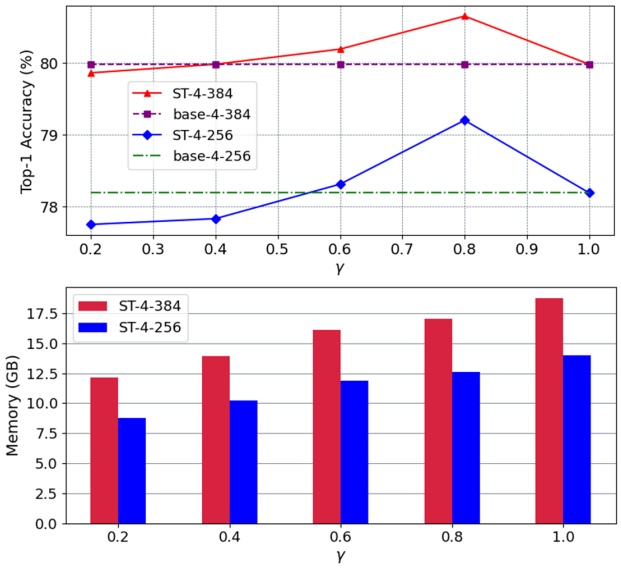

Figure 4. Ablation study of different γ. Top: impact of γ on CIFAR100 Top-1 Accuracy. Bottom: impact of γ on the memory consumption during training.

5.2. Token Sparsification of Spiking Transformers

Comparison with traditional pruning methods on ImageNet. For the purpose of comparison, we evaluate several pruning-based training acceleration strategies, including Random Token and SSA-based methods, based on the Spikformer on the ImageNet dataset. Random Token refers to the random selection and pruning of tokens. SSA-based represents pruning tokens dependent on the average of the original Spiking Self Attention (SSA). We ensure that the training speedup remains approximately consistent ($\sim$1.53$\times$) across these methods. As demonstrated in Table 2, the traditional pruning methods encounter noticeable performance degradation on ImageNet. In contrast, our method achieves a significant improvement compared to the other pruning-based training acceleration methods. Furthermore, we evaluate the inference efficiency using OPs (Merolla et al., 2014) and theoretical energy consumption. Our STATA can achieve $\sim$52% OPs reduction and $\sim$48% energy reduction compared to the original model, demonstrating the improved inference efficiency of our method.

Methods	Acc (%)	Mem (GB)	Training Speedup	Energy (mJ)
Original	79.98	18.74	1$\times$	1.29
SSA-based	78.47	12.10	1.48$\times$	0.70
STATA	79.86	12.16	1.48$\times$	0.71

Table 3. Experiments on accuracy and comprehensive efficiency comparison of Spiking Transformer on CIFAR100.

Comparison with different Spiking Transformers on CIFAR10/100 and neuromorphic Datasets. To further evaluate the effectiveness of our methods among different architectures of Spiking Transformers, we apply our token sparsification approach to Spikformer(Zhou et al., 2023b), Spike-driven Transformer (Yao et al., 2023a) and Spikingformer (Zhou et al., 2023a). As illustrated in Table 4, our STATA demonstrates considerable performance across different Spiking Transformers not only on static datasets such as CIFAR10/100, but also on neuromorphic datasets including CIFAR10-DVS and DVS128 Gesture. These results highlight the transferability of our method across different architectures of Spiking Transformers.

We also present a comprehensive analysis of the efficiency of our STATA method on Spikingformer from multiple perspectives, as shown in Table 3. Our STATA method maintains considerable efficiency in terms of training speed, training memory consumption, and inference energy consumption, which is comparable to the efficiency achieved by SSA-based pruning. However, our approach exhibits a noticeable accuracy advantage over the SSA-based method. Moreover, when compared to the original Spiking Transformer, our STATA method demonstrates superior efficiency in both the training and inference phases.

5.3. Ablation Study

Influence of different γ. Since γ governs the sparsity in our work, we aim to evaluate the influence of various values of γ from multiple perspectives. We investigate five γ configurations: $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, with $\gamma = 1.0$ representing the original Spikingformer. Our experiments are conducted based on two Spikingformer architectures, namely ST-4-384 and ST-4-256, using the CIFAR100 dataset. As depicted in Figure 4, when $\gamma < 0.8$, the accuracy of both two models steadily improves with the growth of γ and reaches the peak at $\gamma = 0.8$. It is noteworthy that our sparsification method can even enhance the model performance compared with the original model in the case of $\gamma = 0.6$ and 0.8. In terms of memory consumption during training, a decrease

Methods	Architecture	CIFAR10		CIFAR100		CIFAR10-DVS		DVS128 Gesture	
		T	Acc	T	Acc	T	Acc	T	Acc
Rollout (Kugele et al., 2020)	VGG-11	-	-	-	-	48	66.8	240	97.2
Hybrid training (Rathi et al., 2020)	VGG-11	125	92.2	125	67.9		-		-
Dspike (Li et al., 2021)	ResNet-19	6	94.3	6	74.2	10	75.4	10	-
STBP-tdBN (Zheng et al., 2021)	ResNet-19/17	4	92.9	4	70.9	10	67.8	10	96.9
PLIF (Fang et al., 2021b)	Spike-based BP	8	93.5	8	70.9	20	74.8	20	97.6
TA-SNN (Yao et al., 2021)	CNN-based SNN	-	-	-	-	10	72	60	98.6
DSR (Meng et al., 2022)	ResNet-19	20	95.4	20	78.5	20	77.3	20	-
TCJA (Zhu et al., 2022a)	VGGSNN	-	-	-	-	10	80.7	20	99
Spikformer (Zhou et al., 2023b)	Spikformer	4	95.2	4	77.9	16	80.9	16	98.3
Spike-driven T (Yao et al., 2023a)	Spike-driven T	4	95.6	4	78.4	16	80.0	16	97.9*
Spikingformer (Zhou et al., 2023a)	Spikingformer	4	95.8	4	80.0	16	81.4	16	98.6
STATA (ours)	Spikformer	4	95.0	4	77.7	16	80.7	16	98.3
	Spike-driven T	4	95.4	4	78.1	16	79.7	16	97.6
	Spikingformer	4	95.8	4	79.9	16	81.2	16	98.6

Table 4. Experiments on different Spiking Transformers. Note that our STATA uses less training resources. Compared to the original Spiking Transformers, our STATA can achieve comparable performance with up to $\sim$1.5$\times$ training acceleration, $\sim$36% memory reduction and $\sim$45% energy saving. $*$ denotes our reproduction. We evaluate not only on static datasets CIFAR10/100, but also on the neuromorphic dataset CIFAR10-DVS and DVS128 Gesture.

TAT	Intra Align	Inter Align	Top-1 Acc (%)	Top-5 Acc (%)
/	/	/	78.31	93.82
AT	/	/	78.98	94.05
✓	/	/	79.23	94.11
✓	✓	/	79.58	94.19
✓	✓	✓	79.86	94.28

Table 5. Ablation study of each proposed component in STATA on CIFAR100. We use random token pruning as the baseline. TAT refers to Timestep-wise Anchor Token, while AT is Anchor Token without timestep-wise. Intra Align and Inter Align refers to Intra-Alignment loss and Inter-Alignment loss during training.

in the value of γ is associated with a reduction in memory usage. When $\gamma = 0.2$, our STATA method achieves $\sim$36% memory reduction during training, while still maintaining performance comparable to the original model. Moreover, the ST-4-384 model with $\gamma = 0.2$ sparsification demonstrates enhanced memory efficiency compared to the original ST-4-256 model.

Effectiveness of each proposed component. To illustrate the contribution of each component, we conduct the ablation study on the CIFAR100 based on Spikingformer(Zhou et al., 2023a). We establish a baseline using random token pruning and gradually introduce each component of the STATA framework. As depicted in Table 5, the incorporation of Anchor Token brings about a significant performance improvement by effectively identifying important tokens. Additionally, the Timestep-wise Anchor Token (TAT) indeed enhances the model performance through its finer-grained identification at the timestep level. With respect to the dual Alignments, both Intra-Alignment and Inter-Alignment contribute to further improving the model performance.

γ	Top-1 Acc (%)	OPs Ratio	Power Ratio
original	79.98	100%	100%
0.8	80.65	82.72%	84.91%
0.6	80.19	72.99%	75.46%
0.4	79.98	63.69%	66.58%
0.2	79.86	54.87%	57.92%

Table 6. Experiments on the inference efficiency of STATA with multiple γ configurations. We illustrate the ratio of our method in terms of OPs and Power ratio compared to the original Spiking Transformer model.

5.4. Efficiency evaluation

Training efficiency on acceleration ratio and memory. To demonstrate the training efficiency of our proposed method, we conduct a series of experiments to evaluate the training speedup and memory consumption. Specifically, we investigate the performance of ST-4-384 on CIFAR100 and ST-8-768 on ImageNet based on Spikingformer architecture, considering different values of pruning ratio factor γ. As illustrated in Figure 5, the training acceleration ratio increases as the value of γ decreases. Notably, when $\gamma = 0.2$, the speedup ratio reaches $\sim$1.6$\times$ on ImageNet and $\sim$1.48$\times$ on CIFAR100. Additionally, it is important to

highlight that the ImageNet model ST-8-768 demonstrates a higher acceleration ratio compared to the CIFAR model ST-4-384.

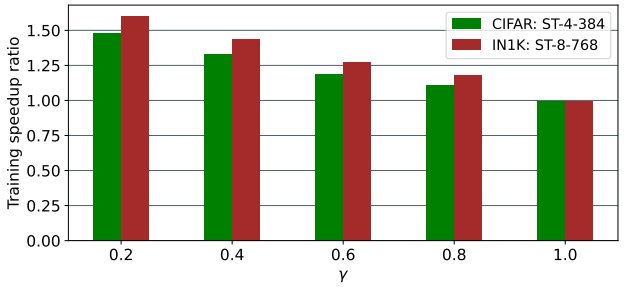

Figure 5. Training speedup ratios on different Spiking Transformer architectures and pruning ratio factor γ. We present the acceleration ratios based on ST-4-384 for CIFAR-100 and ST-8-768 for ImageNet.

Inference efficiency on energy. In addition to the training efficiency, our STATA model also demonstrates energy efficiency during inference. To evaluate this aspect, we conduct experiments using the Spikingformer-4-384 and evaluate its performance on the CIFAR100 test dataset by measuring the OPs and Power ratio. Then, we compare our STATA method with the original Spiking Transformer model to analyze the OPs ratio and Power ratio. The results are presented in Table 6. As shown in it, our STATA method achieves a significant reduction both in computational complexity and energy consumption compared to the original Spiking Transformer. Specifically, our method can save approximately 45% of the original model's OPs and 42% of its energy consumption.

6. Conclusion

In this paper, we propose Sparsification with Timestep-wise Anchor Token and dual Alignments (STATA) as a novel approach to simultaneously enhance the efficiency of both training and inference in the Spiking Transformers. STATA is a versatile method that can be seamlessly integrated with various Spiking Transformer architectures. By leveraging the low-cost Timestep-wise Anchor Token and dual Alignments, our method achieves outstanding efficiency in both training and inference phases while maintaining performance comparable to the original model.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No.2022ZD0160304), the Beijing Municipal Science and Technology Project (No.Z231100010323002), and the Key Research and Development Program of Jiangsu Province (No.BE2023016).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., Kusnitz, J., Debole, M., Esser, S., Delbruck, T., Flickner, M., and Modha, D. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7243–7252, 2017.

Chen, D., Mei, J.-P., Zhang, Y., Wang, C., Wang, Z., Feng, Y., and Chen, C. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7028–7036, 2021a.

Chen, J., Gao, C., Sun, L., and Sang, N. CCSD: cross-camera self-distillation for unsupervised person re-identification. *Vis. Intell.*, 1(1), 2023a. doi: 10.1007/S44267-023-00029-4. URL https://doi.org/10.1007/s44267-023-00029-4.

Chen, W., Wang, P., and Cheng, J. Towards mixed-precision quantization of neural networks via constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5350–5359, 2021b.

Chen, Y., Yu, Z., Fang, W., Huang, T., and Tian, Y. Pruning of deep spiking neural networks through gradient rewiring. *arXiv preprint arXiv:2105.04916*, 2021c.

Chen, Y., Yu, Z., Fang, W., Ma, Z., Huang, T., and Tian, Y. State transition of dendritic spines improves learning of sparse spiking neural networks. In *International Conference on Machine Learning*, pp. 3701–3715. PMLR, 2022.

Chen, Y., Ma, Z., Fang, W., Zheng, X., Yu, Z., and Tian, Y. A unified framework for soft threshold pruning. *arXiv preprint arXiv:2302.13019*, 2023b.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Deng, L., Wu, Y., Hu, Y., Liang, L., Li, G., Hu, X., Ding, Y., Li, P., and Xie, Y. Comprehensive snn compression using admm optimization and activity regularization. *IEEE transactions on neural networks and learning systems*, 2021a.

Deng, S., Li, Y., Zhang, S., and Gu, S. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. In *International Conference on Learning Representations (ICLR)*, 2021b.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representa- tions (ICLR)*, 2020.

Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., and Tian, Y. Deep Residual Learning in Spiking Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21056–21069, 2021a.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2661–2671, 2021b.

Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A. The spinnaker project. *Proceedings of the IEEE*, 102(5): 652–665, 2014.

Guo, G., Han, L., Wang, L., Zhang, D., and Han, J. Semantic-aware knowledge distillation with parameter-free feature uniformization. *Vis. Intell.*, 1(1), 2023. doi: 10.1007/S44267-023-00003-0. URL https://doi.org/10.1007/s44267-023-00003-0.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 784–800, 2018.

Hu, Y., Tang, H., and Pan, G. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021a.

Hu, Y., Wu, Y., Deng, L., and Li, G. Advancing residual learning towards powerful deep spiking neural networks. *arXiv preprint arXiv:2112.08954*, 2021b.

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

Kim, Y., Li, Y., Park, H., Venkatesha, Y., Yin, R., and Panda, P. Exploring lottery ticket hypothesis in spiking neural networks. In *European Conference on Computer Vision*, pp. 102–120. Springer, 2022.

Krestinskaya, O., James, A. P., and Chua, L. O. Neuromemristive circuits for edge computing: A review. *IEEE transactions on neural networks and learning systems*, 31 (1):4–23, 2019.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Kugele, A., Pfeil, T., Pfeiffer, M., and Chicca, E. Efficient Processing of Spatio-temporal Data Streams with Spiking Neural Networks. *Frontiers in Neuroscience*, 14:439, 2020.

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

Li, Y., Guo, Y., Zhang, S., Deng, S., Hai, Y., and Gu, S. Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 23426–23439, 2021.

Li, Y., Lei, Y., and Yang, X. Spikeformer: A novel architecture for training high-performance low-latency spiking neural network. *ArXiv*, abs/2211.10686, 2022.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Lotfi Rezaabad, A. and Vishwanath, S. Long short-term memory spiking networks and their applications. In *Proceedings of the International Conference on Neuromorphic Systems 2020 (ICONS)*, pp. 1–9, 2020.

Maass, W. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671, 1997.

Meng, Q., Xiao, M., Yan, S., Wang, Y., Lin, Z., and Luo, Z.-Q. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. *ArXiv preprint arXiv:2205.00459*, 2022.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

Perez-Nieves, N. and Goodman, D. Sparse spiking gradient descent. *Advances in Neural Information Processing Systems*, 34:11795–11808, 2021.

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

Roy, K., Jaiswal, A., and Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Wang, K., Liu, Z., Lin, Y., Lin, J., and Han, S. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8612–8620, 2019.

Wolchover, N. New theory cracks open the black box of deep learning. 2018.

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

Xiao, Y., Liu, A., Zhang, T., Qin, H., Guo, J., and Liu, X. Robustmq: benchmarking robustness of quantized models. *Vis. Intell.*, 1(1), 2023. doi: 10.1007/S44267-023-00031-W. URL https://doi.org/10.1007/s44267-023-00031-w.

Xu, W., He, X., Cheng, K., Wang, P., and Cheng, J. Towards fully sparse training: Information restoration with spatial similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2929–2937, 2022.

Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., and Li, G. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10221–10230, 2021.

Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., and Li, G. Spike-driven transformer. *arXiv preprint arXiv:2307.01694*, 2023a.

Yao, X., Hu, Q., Liu, T., Mo, Z., Zhu, Z., Zhuge, Z., and Cheng, J. Spiking nerf: Making bio-inspired neural networks see through the real world. *arXiv preprint arXiv:2309.10987*, 2023b.

Yin, H., Lee, J. B., Kong, X., Hartvigsen, T., and Xie, S. 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*, pp. 2017–2025, 2021.

Yin, R., Kim, Y., Li, Y., Moitra, A., Satpute, N., Hambitzer, A., and Panda, P. Workload-balanced pruning for sparse spiking neural networks. *arXiv preprint arXiv:2302.06746*, 2023.

Yue, S., Li, C., Zhuge, Z., and Song, R. Eesrnet: A network for energy efficient super-resolution. In *European Conference on Computer Vision*, pp. 602–618. Springer, 2022.

Zhao, T., Zhang, X. S., Zhu, W., Wang, J., Yang, S., Liu, J., and Cheng, J. Multi-granularity pruning for model acceleration on mobile devices. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2022.

Zheng, H., Wu, Y., Deng, L., Hu, Y., and Li, G. Going Deeper With Directly-Trained Larger Spiking Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 11062–11070, 2021.

Zhou, C., Zhang, H., Zhou, Z., Yu, L., Ma, Z., Zhou, H., Fan, X., and Tian, Y. Enhancing the performance of transformer-based spiking neural networks by improved downsampling with precise gradient backpropagation. *arXiv preprint arXiv:2305.05954*, 2023a.

Zhou, Z., Zhu, Y., He, C., Wang, Y., YAN, S., Tian, Y., and Yuan, L. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=frE4fUwz_h.

Zhu, R., Zhao, Q., Zhang, T.-J., Deng, H., Duan, Y., Zhang, M., and Deng, L.-J. Tcja-snn: Temporal-channel joint attention for spiking neural networks. *ArXiv*, abs/2206.10177, 2022a.

Zhu, Z., Peng, J., Li, J., Chen, L., Yu, Q., and Luo, S. Spiking graph convolutional networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2434–2440, 2022b. doi: 10.24963/ijcai.2022/338.

Zhuge, Z., Wang, J., Li, Y., Bao, Y., Wang, P., and Cheng, J. Patch-aware sample selection for efficient masked image modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17245–17253, 2024.