Spiking Transformer:
Introducing Accurate Addition-Only Spiking Self-Attention for Transformer

Yufei Guo,[*] Xiaode Liu,[*] Yuanpei Chen, Weihang Peng, Yuhan Zhang, Zhe Ma[†]

Intelligent Science & Technology Academy of CASIC, China

yfguo@pku.edu.cn, lxde@pku.edu.cn, mazhe_thu@163.com

Abstract

Transformers have demonstrated outstanding performance across a wide range of tasks, owing to their self-attention mechanism, but they are highly energy-consuming. Spiking Neural Networks have emerged as a promising energy-efficient alternative to traditional Artificial Neural Networks, leveraging event-driven computation and binary spikes for information transfer. The combination of Transformers' capabilities with the energy efficiency of SNNs offers a compelling opportunity. This paper addresses the challenge of adapting the self-attention mechanism of Transformers to the spiking paradigm by introducing a novel approach: Accurate Addition-Only Spiking Self-Attention (A^2OS^2A). Unlike existing methods that rely solely on binary spiking neurons for all components of the self-attention mechanism, our approach integrates binary, ReLU, and ternary spiking neurons. This hybrid strategy significantly improves accuracy while preserving non-multiplicative computations. Moreover, our method eliminates the need for softmax and scaling operations. Extensive experiments show that the A^2OS^2A-based Spiking Transformer outperforms existing SNN-based Transformers on several datasets, even achieving an accuracy of 78.66% on ImageNet-1K. Our work represents a significant advancement in SNN-based Transformer models, offering a more accurate and efficient solution for real-world applications.

1. Introduction

Spiking Neural Networks (SNNs) have gained significant traction as an efficient neural network model, finding applications in diverse areas such as object recognition [15, 16, 34, 56], object detection [29, 44], and pose estimation [66]. These networks utilize binary spike signals for information transfer. A spiking neuron generates a spike, denoted by 1, when its membrane potential surpasses a cer-

tain threshold; conversely, it does not spike, represented by 0, when the threshold is not met. This distinctive approach to information processing is notably energy-efficient, as it substitutes the complex multiplications involved in weight and activation interactions with simpler addition operations. Furthermore, SNNs can be effectively implemented using event-driven computations on neuromorphic hardware [1, 7, 39, 43, 52]. In this setup, the computational units are activated solely upon the occurrence of a spike, conserving energy by remaining inactive in the absence of spikes. Research indicates that SNNs can achieve significantly greater energy savings compared to their Artificial Neural Network (ANN) counterparts [1, 7].

Despite the clear advantages of SNNs in terms of energy efficiency, their practical application is often constrained by limited task accuracy. Meanwhile, Transformers have demonstrated exceptional performance across a wide range of tasks, largely due to their self-attention mechanism [30, 42, 50]. Combining the strengths of Transformers with the energy efficiency of SNNs presents a promising opportunity. However, adapting the self-attention mechanism to SNNs is non-trivial. In the vanilla self-attention framework (VSA), three key components are involved: Query (Q), Key (K), and Value (V). As illustrated in Figure 1(a), the VSA process begins by computing the dot product of the floating-point representations of Q and K, resulting in a matrix. This matrix is then normalized using a softmax function, which involves exponential and division operations, to produce an attention map that determines the weighting of V. These operations in the VSA are incompatible with the operational principles of SNNs, which aim to minimize multiplication. Consequently, to implement a Transformer architecture in SNNs, it is essential to develop a novel, efficient self-attention mechanism that avoids multiplication. Ongoing research efforts are exploring solutions in this domain. Approaches such as Spikformer [64, 65], Spikingformer [63], and Spike-driven Transformer [58] have focused on transforming Q, K, and V into spike representations prior to performing matrix operations. This transformation enables the replacement of matrix multiplications with addition, aligning

---

[*]Equal contribution.
[†]Corresponding author.

better with the principles of SNNs.

While previous approaches utilize spike-form Q, K, and V to avoid multiplications, simply introducing spiking neurons to convert Q, K, and V into spike form prior to performing matrix multiplications does not fully exploit the advantages of the addition-only operation inherent in SNNs. It is unnecessary to convert all of Q, K, and V into spike form in order to transition from matrix multiplications to matrix additions. However, this method can lead to significant information loss, as demonstrated in Section 4. To address this issue, we propose the Accurate Addition-Only Spiking Self-Attention (A^2OS^2A). The distinctions between A^2OS^2A and the vanilla Spiking Self-Attention (VSSA) are illustrated in Figure 1(b) and Figure 1(c). Unlike VSSA, which applies the same binary spiking neuron to generate Q, K, and V, our approach employs a binary spiking neuron for Q, a ReLU activation for K, and a ternary spiking neuron for V. This design allows A^2OS^2A to retain the multiplication-addition transformation while reducing information loss. Moreover, it eliminates the need for both the scaling and softmax functions, enabling the output of floating-point values similar to the VSA, whereas VSSA produces only integer outputs. The key contributions of this paper are as follows:

- We present a theoretical framework for understanding the information loss associated with vanilla Spiking Self-Attention mechanisms. To the best of our knowledge, this is the first detailed examination of this issue within the context of Spiking Self-Attention, which opens the door for future advancements in SNN-based Transformers.

- We introduce A^2OS^2A, an innovative and efficient approach that incorporates binary spiking neurons, ReLU activations, and ternary spiking neurons, moving beyond the traditional binary-only spiking neuron model. This method reduces information loss while retaining the advantages of addition-only processing in SNNs. Additionally, it eliminates the need for scaling and softmax functions.

- Comprehensive experiments demonstrate that our proposed architecture either surpasses or matches the performance of State-of-the-Art (SoTA) SNN-based Transformer models on various datasets. Notably, we achieved an accuracy of 78.66% on ImageNet-1K, setting a new benchmark in the SNN domain.

2. Related Work

2.1. Spiking Neural Networks

SNNs are a class of neural networks that more closely emulate the behavior of biological neurons compared to ANNs. The primary distinction lies in their use of discrete spikes for information transmission, as opposed to the continuous signals employed by ANNs. Foundational work in this area can be traced back to Hodgkin and Huxley [26], who developed models to simulate the action potentials of biological neurons. Since then, numerous frameworks and models have been introduced to enhance the capabilities and learning efficiency of SNNs. A notable example is the Leaky Integrate-and-Fire (LIF) model [40], which provides a simple yet effective means of simulating neuronal behavior. More recently, the incorporation of Spike-Timing-Dependent Plasticity (STDP) as a learning rule has enabled SNNs to learn temporal patterns effectively [4].

In the past decade, there has been a significant increase in interest regarding the training of SNNs using techniques adapted from ANNs, as well as their application to practical tasks. Two prevalent learning paradigms in current SNN research are the conversion from ANN to SNN (ANN2SNN) [2, 3, 22–25, 32, 34, 48] and supervised learning [12–14, 18–21, 35, 47, 59]. The ANN-SNN conversion method involves initially training an ANN and subsequently converting it into a homogeneous SNN by transferring the trained weights and substituting ReLU neurons with temporal spiking neurons. However, this method often fails for neuromorphic datasets, as ReLU neurons do not adequately capture the complex temporal dynamics required for processing sequential information. In contrast, supervised learning [11, 17, 53, 61] employs alternative functions during backpropagation to approximate the firing process, enabling the direct training of SNNs as if they were ANNs. This approach leverages the advantages of gradient-based optimization, achieving impressive performance with just a few time steps, even on large-scale datasets. Furthermore, supervised learning has proven to be effective in handling temporal data, establishing it as a preferred methodology in SNN research.

2.2. Vision Transformers

Vision Transformers (ViTs) have recently emerged as a powerful alternative to convolutional neural networks (CNNs) for a wide range of visual tasks, demonstrating remarkable efficacy and flexibility. Introduced by Dosovitskiy et al. [10], ViTs leverage the transformer architecture—originally designed for natural language processing—to process images in a novel manner. The architecture innovatively splits images into patches and treats these patches as sequences, enabling the model to capture long-range dependencies and relationships within the visual data, a limitation often encountered in traditional CNNs. The introduction of ViTs has spurred significant advancements in various computer vision tasks, including image classification [41], object detection [38], and segmentation [28]. These advancements highlight the potential of ViTs to tackle complex visual recognition problems with unprecedented accuracy. Notably, it has been shown that with appropriate training techniques and larger datasets, ViTs can not only match but surpass the performance of state-of-the-art CNNs, even in data-scarce scenarios, presenting a
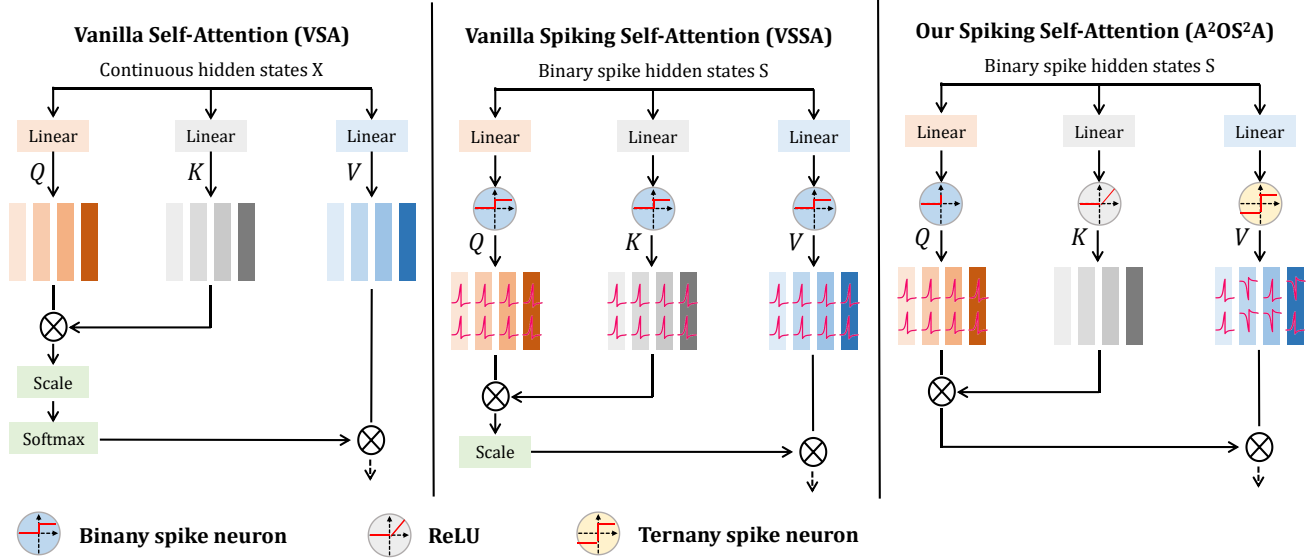
Figure 1. The difference between our Spiking Self-Attention and the vanilla Spiking Self-Attention. Our Spiking Self-Attention differs significantly from the vanilla Spiking Self-Attention. In the vanilla version, only binary spikes are employed, which can result in considerable information loss. In contrast, our approach integrates a combination of binary spiking neurons, ReLU, and ternary spiking neurons. This hybrid structure effectively reduces information loss, while preserving the advantages of addition-only processing. Moreover, our method eliminates the need for both scaling and the softmax function, streamlining the computation.

significant advantage in real-world applications [49].

Furthermore, the adaptation of transformers for visual tasks has inspired numerous variations and improvements across the field. Researchers have explored strategies such as integrating convolutional layers into the architecture [6] to retain spatial information more effectively. Other approaches include the use of hierarchical representations [36], which facilitate multi-scale feature extraction, and the incorporation of attention mechanisms tailored specifically for vision tasks [5], improving the model's ability to focus on relevant parts of the input data. These developments underscore the versatility and robustness of the transformer architecture in addressing complex visual challenges, paving the way for further innovations in computer vision.

2.3. Spiking Neural Networks for Transformers

The integration of SNNs into transformer architectures represents an exciting and rapidly developing area of research. This field aims to leverage the distinct advantages of both paradigms: the energy-efficient characteristics of SNNs and the robust contextual representation capabilities of transformers. Recent studies [33, 37, 51] have explored replacing certain neurons within transformers with spiking neurons, marking a significant step towards this integration. While these efforts contribute to mitigating the accuracy loss associated with introducing spiking neurons into transformers, they still face challenges in fully realizing the low energy consumption benefits of SNNs. This is primarily due to

their reliance on hybrid computing methodologies, which continue to require traditional Multiply-Accumulate (MAC) operations, such as dot products, softmax calculations, and scaling operations.

To address this issue, Spikformer [64, 65] proposes converting the Q, K, and V components into spike form before performing matrix multiplications similar to those used in VSA. This innovative approach allows spike matrix multiplications to be transformed into addition operations, eliminating the need for softmax computations. However, when considering the residual connections in these architectures, Spikformer [64, 65] still incorporates non-spiking computations within the ConvBN layers. To overcome this limitation, Spikingformer [63] and Spike-driven Transformer [58] have restructured the residual connections in standard transformer architectures, placing them before activation functions to ensure that all neurons convey binary spike signals.

The computation of spike-form Q, K, and V in these models circumvent multiplications, relying solely on additions. However, simply placing spiking neurons in front of Q, K, and V to convert them into spike form before performing matrix multiplications does not fully exploit the addition-only advantage of SNNs. In this work, we propose a novel, addition-only spiking self-attention mechanism for transformers, aiming to push the boundaries of what can be achieved through the integration of these two technologies.

3. Preliminary

3.1. Spiking Neuron Layer

The spiking neuron layer plays a crucial role in integrating both spatial and temporal information, which is then encoded into the membrane potential and subsequently transformed into binary spikes. These spikes drive further computations in subsequent layers of the network. We model the dynamics of the spiking neuron using the LIF model [40]. The evolution of the membrane potential and the spike generation mechanism are described by the following equations:

$$U[t] = H[t-1] + X[t], \qquad (1)$$
$$S[t] = \text{Hea}(U[t] - V_{\text{th}}), \qquad (2)$$
$$H[t] = V_{\text{reset}}S[t] + \beta U[t](1 - S[t]), \qquad (3)$$

where: $X[t]$ is the spatial input current at time t, $U[t]$ is the membrane potential at time t, combining the spatial input $X[t]$ and the temporal input $H[t-1]$ from the previous time step, $\text{Hea}(\cdot)$ is the Heaviside step function, which outputs 1 if its argument is non-negative, and 0 otherwise, $S[t]$ is the spike output at time t, V_{th} is the spike threshold, V_{reset} is the reset potential, and β is a decay factor controlling the membrane potential's decay.

When the membrane potential $U[t]$ exceeds a predefined threshold V_{th}, the neuron emits a spike, $S[t] = 1$, and the internal state $H[t]$ is reset to V_{reset}. If $U[t]$ does not surpass the threshold, the membrane potential decays towards the previous state $H[t-1]$ at a rate governed by β. To simplify notation, we represent the spiking neuron layer as $\mathcal{SN}(\cdot)$, where the input is the membrane potential tensor U, and the output is the spike tensor S.

3.2. Vanilla Self-Attention Mechanism (VSA)

The Vanilla Self-Attention (VSA) mechanism enables a model to focus on different parts of the input sequence while constructing its output representations. The self-attention operation can be mathematically expressed as:

$$\text{VSA}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \qquad (4)$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. The softmax function is applied to normalize the attention scores, ensuring they are non-negative. The resulting weighted sum of the values reflects the importance of each value, based on the similarity between the query and its corresponding key.

The input to the self-attention mechanism consists of a sequence of embeddings $x_1, x_2, ..., x_n$, which are linearly projected into Q, K, and V using learned weight matrices:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \qquad (5)$$

where W_Q, W_K, and W_V are the learned weight matrices corresponding to the Q, K, and V, respectively.

However, the standard VSA is not directly compatible with SNNs due to the conflict between the floating-point operations required for matrix multiplication of Q, K, and V, and the energy efficiency of SNN computations. What's more, the softmax operation, which involves exponentiation and division, does not align with the computation paradigm of SNNs too.

3.3. Vanilla Spiking Self-Attention Mechanism (VSSA)

To address the incompatibility between the VSA and SNNs, the Vanilla Spiking Self-Attention (VSSA) mechanism is proposed in [64, 65], which is more suitable for SNNs, as shown in Figure 1(b). The Q, K, and V are initially derived through learnable weight matrices. These are then transformed into spiking sequences by applying binary spiking neuron layers:

$$Q = \mathcal{SN}_Q(\text{BN}(XW_Q)), \qquad (6)$$
$$K = \mathcal{SN}_K(\text{BN}(XW_K)), \qquad (7)$$
$$V = \mathcal{SN}_V(\text{BN}(XW_V)). \qquad (8)$$

In this approach, the computation of the attention matrix is performed using binary spike-form queries and keys (which contain only 0s and 1s), thus replacing traditional matrix multiplication with addition-based operations.

To further address the challenges of large values resulting from the matrix multiplication, a scaling factor s is introduced to regulate the magnitude of the result. The spike-efficient VSSA is formulated as follows:

$$\text{VSSA}(Q, K, V) = \mathcal{SN}\left(Q\ K^T\ V \cdot s\right). \qquad (9)$$

In this formulation, the scaling factor s adjusts the magnitude of the result from the matrix product, while all other operations in VSSA—such as the attention calculation—are performed using addition, in line with the spike-based computation model.

However, simply converting the queries, keys, and values into spike form prior to performing matrix multiplication does not fully leverage the advantages of SNNs, which thrive on addition-only and event-driven computations. It is unnecessary to convert all of Q, K, and V into binary spike sequences to transition from matrix multiplication to addition. Furthermore, converting all of the values to binary spikes can result in significant information loss. Thus, the challenge remains to strike a balance between maintaining the richness of the data while taking full advantage of the addition operation-based nature of SNNs.
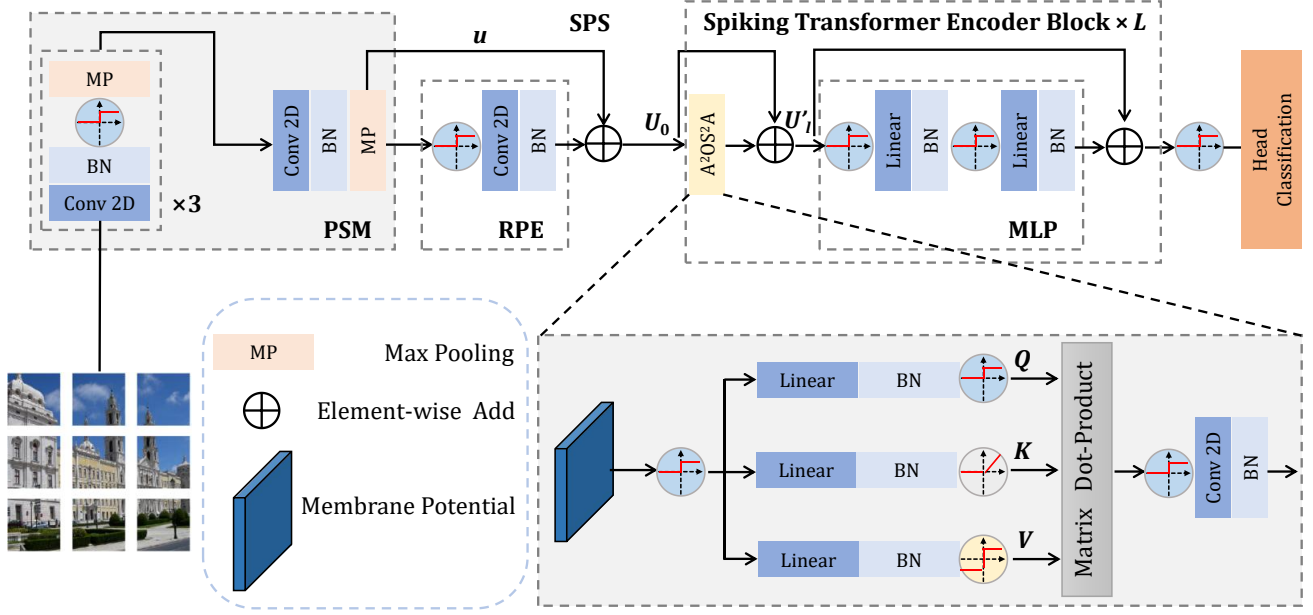
Figure 2. The overview of Spiking Transformer.

4. Method

4.1. Overall Architecture

Figure 2 illustrates the architecture of our Spiking Transformer, which comprises four primary components: Spiking Patch Splitting (SPS), Accurate Addition-Only Spiking Self-Attention (A^2OS^2A), Multi-Layer Perceptron (MLP), and a linear classification head. The design follows the approach outlined in [58]. For the SPS module, we adopt the structure from [63, 64]. Given a 2D image sequence $I \in \mathbb{R}^{T \times C \times H \times W}$, the Patch Splitting Module (PSM) performs a linear projection and splits the input into a sequence of N flattened spike patches with D-dimensional channels, where T denotes the number of timesteps (images are repeated T times in the dataset), C is the number of channels, and H and W represent the height and width of the image sequence. Additionally, a convolutional layer generates Relative Position Embeddings (RPE) as described in [58]. Together, the SPS part is written as:

$$u = \mathrm{PSM}(I), I \in \mathbb{R}^{T \times C \times H \times W}, x \in \mathbb{R}^{T \times N \times D}, \quad (10)$$

$$s = \mathcal{SN}(u), s \in \mathbb{R}^{T \times N \times D}, \quad (11)$$

$$\mathrm{RPE} = \mathrm{BN}((\mathrm{Conv2d}(s))), \mathrm{RPE} \in \mathbb{R}^{T \times N \times D}, \quad (12)$$

$$U_0 = u + \mathrm{RPE}, U_0 \in \mathbb{R}^{T \times N \times D}, \quad (13)$$

where u and U_0 are the output membrane potential tensors from the PSM and SPS modules, respectively, and $\mathcal{SN}(\cdot)$

represents the spike neuron layer. The output U_0 is then passed to the L-block Spiking Transformer encoder, which consists of the A^2OS^2A and MLP blocks. Residual connections are applied to the membrane potentials in both the A^2OS^2A and MLP blocks. To avoid multi-bit spike outputs, we adopt the design from [58, 63], ensuring that the values before the convolutional layers are binary, thus allowing the spike and weight matrix multiplication to be simplified to addition operations.

The A^2OS^2A mechanism models both local and global information of the input sequence using binary spike Q, full-precision K, and ternary spike V. This approach reduces information loss and eliminates the need for scaling and softmax operations, thereby improving task accuracy. A Global Average Pooling (GAP) is applied to the processed feature from the Spiking Transformer encoder, and the resulting D-dimensional channel is passed to a fully connected Classification Head (CH) for final prediction. The A^2OS^2A, MLP, and CH components are as follows:

$$S_0 = \mathcal{SN}(U_0), S_0 \in \mathbb{R}^{T \times N \times D}, \quad (14)$$

$$U'_l = A^2OS^2A(S_{l-1}) + U_{l-1}, U'_l \in \mathbb{R}^{T \times N \times D}, l = 1...L, \quad (15)$$

$$S'_l = \mathcal{SN}(U'_l), S'_l \in \mathbb{R}^{T \times N \times D}, \quad (16)$$

$$S_l = \mathcal{SN}(\mathrm{MLP}(S'_l) + U'_l), S_l \in \mathbb{R}^{T \times N \times D}, l = 1...L, \quad (17)$$

$$Y = \mathrm{CH}(\mathrm{GAP}(S_L)). \quad (18)$$

4.2. Information Loss in Spiking Transformers

In this section, we analyze the limitations of binary spikes in SNNs with respect to information representation, which motivates the focus of this paper. While binary spikes offer energy efficiency, they inherently suffer from reduced representational capacity, leading to information loss when used for all Q, K, and V in VSSA, which results in accuracy degradation.

To support this claim, we first perform a theoretical analysis using the concept of entropy. The representational capability $\mathcal{C}(\mathbf{X})$ of a set $\mathbf{X}$ is determined by the maximum entropy of $\mathbf{X}$, expressed as:

$$\mathcal{C}(\mathbf{X}) = \max \mathcal{H}(\mathbf{X}) = -\sum_{x \in \mathbf{X}} p_{\mathbf{X}}(x) \log p_{\mathbf{X}}(x), \quad (19)$$

where $p_{\mathbf{X}}(x)$ is the probability of a sample x from $\mathbf{X}$. We now present the following proposition:

Proposition 1 *For a set $\mathbf{X}$, its representational capacity is $\mathcal{C}(\mathbf{X}) = \max \mathcal{H}(\mathbf{X})$. When the probability distribution of $\mathbf{X}$ is uniform, i.e., $p_{\mathbf{X}}(x) = \frac{1}{N}$, where N is the total number of samples in $\mathbf{X}$, the entropy $\mathcal{H}(\mathbf{X})$ reaches its maximum value of $\log(N)$. Hence, we conclude that $\mathcal{C}(\mathbf{X}) = \log(N)$.*

Using Proposition 1, we can evaluate the representational capacity of binary spike layers in SNNs and compare them with real-valued layers in ANNs. Let $\mathbf{F}_{\mathrm{B}} \in \mathbb{B}^{C \times H \times W}$ denote the binary spike layer of the SNN, and $\mathbf{F}_{\mathrm{R}} \in \mathbb{R}^{C \times H \times W}$ denote the real-valued feature map in the corresponding ANN. For a binary spike output s, it requires 1 bit, and thus the number of possible samples from s is 2. Therefore, the number of samples for $\mathbf{F}_{\mathrm{B}}$ is $2^{(C \times H \times W)}$, leading to:

$$\mathcal{C}(\mathbf{F}_{\mathrm{B}}) = \log \left(2^{(C \times H \times W)} \right) = C \times H \times W. \quad (20)$$

In contrast, a real-valued output requires 32 bits per sample, yielding 2^{32} possible values. Hence, the representational capacity for real-valued layers is:

$$\mathcal{C}(\mathbf{F}_{\mathrm{R}}) = \log \left(2^{32 \times (C \times H \times W)} \right) = 32 \times C \times H \times W. \quad (21)$$

This highlights the limited representational capacity of the binary spike layer. By transforming the real-valued $Q_{\mathbb{R}}, K_{\mathbb{R}}, V_{\mathbb{R}}$ into binary $Q_{\mathbb{B}}, K_{\mathbb{B}}, V_{\mathbb{B}}$, significant information loss occurs, reducing accuracy. Therefore, we propose increasing the bit precision of Q, K, and V in spiking attention to minimize information loss, while preserving the advantages of multiplication-addition transformations.

4.3. Accurate Addition-Only Spiking Self-Attention Mechanism (A²OS²A)

To maintain the integrity of information within spiking attention while leveraging the advantages of multiplication-addition transformations, we propose the Accurate Addition-Only Spiking Self-Attention (A²OS²A). This mechanism employs binary Q, full-precision K, and ternary V in the self-attention process. Specifically, we utilize a binary spiking neuron after the linear layer to produce Q, a ReLU function for generating K, and a ternary spiking neuron for producing V, as illustrated in Figure 1(c). The formulations for Q, K, V in A²OS²A are defined as follows:

$$Q = \mathcal{SN}_Q^b(\mathrm{BN}(XW_Q)), \quad (22)$$

$$K = \mathrm{ReLU}_K(\mathrm{BN}(XW_K)), \quad (23)$$

$$V = \mathcal{SN}_V^t(\mathrm{BN}(XW_V)), \quad (24)$$

where $\mathcal{SN}^b(\cdot)$ denotes the binary spiking neuron as used in standard SNNs, $\mathrm{ReLU}_K(\cdot)$ is the ReLU activation function typical in ANNs, and $\mathcal{SN}^t(\cdot)$ represents a ternary spiking neuron that outputs values in the set $\{-1, 0, 1\}$.

The evolution of the membrane potential and the spike generation mechanism within $\mathcal{SN}^t(\cdot)$ can be described by the following equations:

$$U[t] = H[t-1] + X[t], \quad (25)$$

$$S[t] = \mathrm{Hea}(|U[t]| - V_{\mathrm{th}}), \quad (26)$$

$$H[t] = V_{\mathrm{reset}} S[t] + \beta U[t](1 - |S[t]|). \quad (27)$$

The formulation of the A²OS²A is expressed as:

$$\mathrm{A^2OS^2A}(Q, K, V) = \mathcal{SN}\left(Q \cdot K^T \cdot V\right). \quad (28)$$

In A²OS²A, since Q takes values in $\{0, 1\}$ and K is in $\mathbb{R}$, the matrix multiplication $Q \cdot K^T$ can be transformed into equivalent addition operations, resulting in $Q \cdot K^T \in \mathbb{R}$. Moreover, because both Q and K are naturally non-negative, this preserves a non-negative attention map $Q \cdot K^T$. Consequently, the mechanism eliminates the need for a softmax operation to enforce non-negativity within the attention map. Furthermore, since K is a real-valued vector without boundaries, there is no necessity for a scaling factor s to manage large values resulting from the matrix multiplication, unlike the approaches in [64, 65]. Additionally, given that V can also take values in $\{-1, 0, 1\}$, the matrix product $Q \cdot K^T \cdot V$ can similarly be expressed through addition operations. At the same time, unlike other matrix multiplication methods outlined in [64, 65], our framework allows for the result to include negative values, akin to what is seen in traditional ANNs.

In summary, the A²OS²A mechanism utilizes full-precision K and ternary spike V to enhance the representational capability of self-attention and minimize information loss. Additionally, all attention calculations are performed using addition operations, which align with the characteristics of SNNs.

5. Experiment

We evaluate our method on various datasets, including CIFAR-10/100 [31] and ImageNet-1K [8]. The network

Table 1. Ablation study for Spiking Transformer on CIFAR10/100. Param refers to the number of parameters. Spiking Transformer-*L-D* is a Spiking Transformer model with *L* Spiking Transformer encoder blocks and *D* feature embedding dimensions.

Methods	Param (M)	Time Step	CIFAR10 Acc	CIFAR100 Acc
Baseline-2-256	2.59	4	94.39	76.00
Spiking Transformer-2-256	2.59	4	**94.91**	**76.96**
Baseline-2-512	10.23	4	95.51	78.83
Spiking Transformer-2-512	10.23	4	**96.42**	**79.90**

Table 2. Comparison of the performance between Spiking Transformer and existing approaches on CIFAR10/100.

Methods	Param (M)	Time Step	CIFAR10 Acc	CIFAR100 Acc
Hybrid training[46]	9.27	125	92.22	67.87
Diet-SNN[45]	0.27	10/5	92.54	64.07
STBP[55]	17.54	12	89.83	-
STBP NeuNorm[54]	17.54	12	90.53	-
TSSL-BP[60]	17.54	5	91.41	-
STBP-tdBN[62]	12.63	4	92.92	70.86
TET[9]	12.63	4	94.44	74.47
Spikformer-4-256[64]	4.15	4	93.94	75.96
Spikformer-2-384[64]	5.76	4	94.80	76.95
Spikformer-4-384[64]	9.32	4	95.19	77.86
Spikingformer-4-256[63]	4.15	4	94.77	77.43
Spikingformer-2-384[63]	5.76	4	95.22	78.34
Spikingformer-4-384[63]	9.32	4	95.61	79.09
Spiking Transformer-4-256	4.15	4	**94.96**	**77.49**
Spiking Transformer-2-384	5.76	4	**95.70**	**78.59**
Spiking Transformer-4-384	9.32	4	**96.32**	**79.69**
Spiking Transformer-2-512	10.23	4	**96.42**	**79.90**

architecture and experimental setup follow the baseline from Spike-driven Transformer [58]. Further details regarding the experimental settings can be found in the Spike-driven Transformer [58].

5.1. Ablation Study

We conduct a series of ablation experiments to assess the effectiveness of the proposed Spiking Transformer, comparing it with the baseline model from [58] on the CIFAR-10 and CIFAR-100 datasets. The results are summarized in Table 1.

The baseline models used are Spike-driven Transformer-2-256 and Spike-driven Transformer-2-512. For the Transformer-2-256 with 4 timesteps, the baseline accuracy is 94.39% on CIFAR-10 and 76.00% on CIFAR-100, consistent with previous results. Our Spiking Transformer, using the same architecture, achieves notable improvements, with accuracies of 94.91% and 76.96% on CIFAR-10 and CIFAR-100, respectively—resulting in performance gains of approximately 0.5% and 1.0%. For the larger model, Transformer-2-512 with 4 timesteps achieves baseline accuracies of 95.51% and 78.43% on CIFAR-10 and CIFAR-100. Our model with the same architecture shows substantial improvements, reaching accuracy scores of 96.42% and 79.90%, corresponding to gains of approximately 0.9% and 1.1% on CIFAR-10 and CIFAR-100, respectively.

5.2. CIFAR

To further evaluate the performance of our method, we compare it with recent SNN-based Transformer approaches, including Spikformer [64], Spikingformer [63] and so on, using a range of models with varying embedding dimensions and numbers of transformer blocks on the CIFAR dataset. The CIFAR dataset consists of 50,000 training images and 10,000 test images, all with a resolution of 32×32. We maintain the experimental setup from [58, 63, 64], including network architecture, training configurations, etc. Table 2 presents the accuracy of our method compared to other models on the CIFAR dataset. As shown in Table 2, our Spiking Transformer-2-384 achieves an accuracy of 95.70% on the CIFAR-10 dataset, surpassing TET (94.44%) and even

Spikingformer-4-384 (94.97%). Notably, performance improves further with higher embedding dimensions and more transformer blocks. In particular, Spiking Transformer-4-384 shows a 1.13% improvement over Spikformer-4-384 and a 0.71% improvement over Spikingformer-4-384. The performance gain of our Spiking Transformer is even more pronounced on more complex datasets such as CIFAR-100. Specifically, Spiking Transformer-4-384 achieves a notable improvement of 1.83% over Spikformer-4-384.

5.3. ImageNet

We further evaluated our approach on the challenging ImageNet-1K dataset. ImageNet-1K [8] is a widely-used benchmark for image classification, consisting of 1,000 categories. The dataset includes approximately 1.28 million training images and 50,000 test images. For both training and evaluation, the images are resized to a default resolution of 224×224 pixels. To ensure consistency with previous work, we maintain the experimental setup from [58, 63, 64], including network architecture, training configurations, and other experimental details.

Table 3 presents a comparative analysis of recent SoTA methods evaluated on the ImageNet dataset. Notable models in this domain include Spikformer [64], Spikingformer [63], and Spike-driven Transformer [58], which achieve accuracies of 73.38%, 74.79%, and 74.57%, respectively, with Transformer-8-512 as the baseline configuration. In comparison, our approach demonstrates a significant improvement, achieving an accuracy of 76.28%, which represents a 1.49% increase over the best-performing transformer-based SNN model, Spikingformer. This improvement highlights the

Table 3. Comparison of the performance between Spiking Transformer and existing approaches on ImageNet-1k.

Category	Methods	Architecture	Param (M)	Time Step	Acc
ANN-to-SNN	Hybrid training[46]	ResNet-34	21.79	250	61.48
	Spiking ResNet[27]	ResNet-34	21.79	350	71.61
		ResNet-50	25.56	350	72.75
	QCFS[2]	VGG-16	138.42	64	72.85
	COS[24]	ResNet-34	21.79	8	74.17
Directly Learning	TET[9]	Spiking-ResNet-34	21.79	6	64.79
		SEW-ResNet-34	21.79	4	68.00
	STBP-tdBN[62]	Spiking-ResNet-34	21.79	6	63.72
		SEW-ResNet-34	21.79	4	67.04
	SEW ResNet[11]	SEW-ResNet-50	25.56	4	67.78
		SEW-ResNet-101	44.55	4	68.76
		SEW-ResNet-152	60.19	4	69.26
	Attention-SNN[57]	ResNet-104	78.37	4	77.08
Directly Learning	Spikformer[64]	Spikformer-8-384	16.81	4	70.24
		Spikformer-6-512	23.37	4	72.46
		Spikformer-8-512	29.68	4	73.38
		Spikformer-10-512	36.01	4	73.68
		Spikformer-8-768	66.34	4	74.81
Directly Learning	Spikingformer[63]	Spikingformer-8-384	16.81	4	72.45
		Spikingformer-8-512	29.68	4	74.79
		Spikingformer-8-768	66.34	4	75.85
Directly Learning	Spike-driven Transformer[58]	Spike-driven Transformer-8-384	16.81	4	72.28
		Spike-driven Transformer-6-512	23.37	4	74.11
		Spike-driven Transformer-8-512	29.68	4	74.57
		Spike-driven Transformer-10-512	36.01	4	74.66
		Spike-driven Transformer-8-768	66.34	4	77.07
Directly Learning	**Spiking Transformer**	Spiking Transformer-8-384	16.81	4	**74.04**
		Spiking Transformer-6-512	23.37	4	**76.22**
		Spiking Transformer-8-512	29.68	4	**76.28**
		Spiking Transformer-10-512	36.01	4	**78.66**

effectiveness of our method in leveraging the potential of spiking neuron models within the transformer architecture, resulting in enhanced performance on large-scale datasets like ImageNet-1K.

Furthermore, even with a smaller network architecture, our Spiking Transformer-10-512 model outperforms the other methods, including those using Transformer-8-768 as the baseline. This demonstrates that our approach is not only more accurate but also more efficient, as it achieves superior results with fewer parameters, highlighting its scalability and robustness.

6. Conclusion

In this paper, we have introduced the Accurate Addition-Only Spiking Self-Attention (A^2OS^2A) mechanism as a novel method for integrating Spiking Neural Networks with Transformer models. Our approach tackles the inherent challenge of adapting the self-attention mechanism to the spiking paradigm by proposing a hybrid solution that combines binary, ReLU, and ternary spiking neurons. This innovative hybrid strategy enables the elimination of both softmax and scaling operations, which are traditionally essential in Transformer models, while preserving the fundamental advantages of addition-only computations in SNNs.

Through extensive experiments, we have demonstrated that A^2OS^2A outperforms existing SNN-based Transformer models in terms of both accuracy and efficiency. Specifically, our method achieves a SoTA accuracy of 78.66% on the ImageNet-1K dataset. This performance, combined with the energy-efficient nature of SNNs, makes A^2OS^2A a promising candidate for real-world applications, especially those requiring high computational performance with low energy consumption.

Acknowledgment

References

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015. 1

[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. 2, 8

[3] Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11–20, 2022. 2

[4] Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a hebbian learning rule. *Annu. Rev. Neurosci.*, 31 (1):25–46, 2008. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[6] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3

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

[8] 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, 7

[9] 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, 8

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[11] 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. 2, 8

[12] Yufei Guo, Weihang Peng, Xiaode Liu, Yuanpei Chen, Yuhan Zhang, Xin Tong, Zhou Jie, and Zhe Ma. Enof-snn: Training accurate spiking neural networks via enhancing the output feature. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2

[13] Yufei Guo, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Yinglei Wang, Xuhui Huang, and Zhe Ma. IM-loss: Information maximization loss for spiking neural networks. In *Advances in Neural Information Processing Systems*, 2022.

[14] Yufei Guo, Yuanpei Chen, Liwen Zhang, YingLei Wang, Xiaode Liu, Xinyi Tong, Yuanyuan Ou, Xuhui Huang, and Zhe Ma. Reducing information loss for spiking neural networks. In *Computer Vision – ECCV 2022*, pages 36–52, Cham, 2022. Springer Nature Switzerland. 2

[15] Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 326–335, 2022. 1

[16] Yufei Guo, Liwen Zhang, Yuanpei Chen, Xinyi Tong, Xiaode Liu, YingLei Wang, Xuhui Huang, and Zhe Ma. Real spike: Learning real-valued spikes for spiking neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 52–68. Springer, 2022. 1

[17] Yufei Guo, Xiaode Liu, Yuanpei Chen, Liwen Zhang, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Rmp-loss: Regularizing membrane potential distribution for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17391–17401, 2023. 2

[18] Yufei Guo, Weihang Peng, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Xuhui Huang, and Zhe Ma. Joint a-snn: Joint training of artificial and spiking neural networks via self-distillation and weight factorization. *Pattern Recognition*, 142:109639, 2023. 2

[19] Yufei Guo, Yuhan Zhang, Yuanpei Chen, Weihang Peng, Xiaode Liu, Liwen Zhang, Xuhui Huang, and Zhe Ma. Membrane potential batch normalization for spiking neural networks. *arXiv preprint arXiv:2308.08359*, 2023.

[20] Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 12244–12252, 2024.

[21] Yufei Guo, Yuanpei Chen, Zecheng Hao, Weihang Peng, Zhou Jie, Yuhan Zhang, Xiaode Liu, and Zhe Ma. Take a shortcut back: Mitigating the gradient vanishing for training spiking neural networks. *Advances in Neural Information Processing Systems*, 37:24849–24867, 2025. 2

[22] Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision*, pages 388–404. Springer, 2020. 2

[23] Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ann-snn conversion error through resid-

ual membrane potential. *arXiv preprint arXiv:2302.02091*, 2023.

[24] Zecheng Hao, Jianhao Ding, Tong Bu, Tiejun Huang, and Zhaofei Yu. Bridging the gap between anns and snns by calibrating offset spikes. *arXiv preprint arXiv:2302.10685*, 2023. 8

[25] Nguyen-Dong Ho and Ik-Joon Chang. Tcl: an ann-to-snn conversion with trainable clipping layers. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 793–798. IEEE, 2021. 2

[26] Allan L Hodgkin and Andrew F Huxley. Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *The Journal of physiology*, 116(4):449, 1952. 2

[27] Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021. 8

[28] Asifullah Khan, Zunaira Rauf, Abdul Rehman Khan, Saima Rathore, Saddam Hussain Khan, Najmus Saher Shah, Umair Farooq, Hifsa Asif, Aqsa Asif, Umme Zahoora, et al. A recent survey of vision transformers for medical image segmentation. *arXiv preprint arXiv:2312.00634*, 2023. 2

[29] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection, 2019. 1

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer White-head, Alexander C Berg, Wan-Yen Lo, et al. Segment any-thing. In *Proceedings of the IEEE/CVF International Confer-ence on Computer Vision*, pages 4015–4026, 2023. 1

[31] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010. 6

[32] Yuxiang Lan, Yachao Zhang, Xu Ma, Yanyun Qu, and Yun Fu. Efficient converted spiking neural network for 3d and 2d classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9211–9220, 2023. 2

[33] Nathan Leroux, Jan Finkbeiner, and Emre Neftci. Online transformers with spiking neurons for fast prosthetic hand control. In *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–6. IEEE, 2023. 3

[34] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spik-ing neural networks calibration. In *International Conference on Machine Learning*, pages 6316–6325. PMLR, 2021. 1, 2

[35] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Ad-vances in Neural Information Processing Systems*, 34:23426–23439, 2021. 2

[36] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 3

[37] Yudong Li, Yunlin Lei, and Xu Yang. Spikeformer: a novel architecture for training high-performance low-latency spik-ing neural network. *arXiv preprint arXiv:2211.10686*, 2022. 3

[38] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object de-tection. In *European conference on computer vision*, pages 280–296. Springer, 2022. 2

[39] De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of Systems Architecture*, 77: 43–51, 2017. 1

[40] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10 (9):1659–1671, 1997. 2, 4

[41] ZHI Min and LU Jingfang. A review of vision transformer for image classification. *Journal of Zhengzhou University: Engineering Science*, 45(4), 2024. 2

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF Inter-national Conference on Computer Vision*, pages 4195–4205, 2023. 1

[43] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019. 1

[44] Jinye Qu, Zeyu Gao, Tielin Zhang, Yanfeng Lu, Huajin Tang, and Hong Qiao. Spiking neural network for ultra-low-latency and high-accurate object detection, 2023. 1

[45] Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encod-ing with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020. 7

[46] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backprop-agation. *arXiv preprint arXiv:2005.01807*, 2020. 7, 8

[47] Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xi-aode Liu, Yuhan Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *arXiv preprint arXiv:2310.06232*, 2023. 2

[48] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019. 2

[49] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through atten-tion. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 3

[50] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[51] Shuai Wang, Malu Zhang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Yu Liang, Yimeng Shan, Qian Sun, Enqi Zhang, and Yang Yang. Spiking vision transformer with saccadic attention. *arXiv preprint arXiv:2502.12677*, 2025. 3

[52] Wenjie Wei, Malu Zhang, Jilin Zhang, Ammar Belatreche, Jibin Wu, Zijing Xu, Xuerui Qiu, Hong Chen, Yang Yang, and Haizhou Li. Event-driven learning for spiking neural networks. *arXiv preprint arXiv:2403.00270*, 2024. 1

[53] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018. 2

[54] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1311–1318, 2019. 7

[55] Y. et al. Wu. Spatio-temporal learning with spiking neural networks. *Nature Communications*, 9, 2018. 7

[56] Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Yisen Wang, and Zhouchen Lin. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *Advances in Neural Information Processing Systems*, 34:14516–14528, 2021. 1

[57] Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9393–9410, 2023. 8

[58] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36, 2024. 1, 3, 5, 7, 8

[59] Malu Zhang, Jiadong Wang, Jibin Wu, Ammar Belatreche, Burin Amornpaisannon, Zhixuan Zhang, Venkata Pavan Kumar Miriyala, Hong Qu, Yansong Chua, Trevor E Carlson, et al. Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE transactions on neural networks and learning systems*, 33 (5):1947–1958, 2021. 2

[60] Wenrui Zhang and Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, 33:12022–12033, 2020. 7

[61] Yuhan Zhang, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yufei Guo, Xuhui Huang, and Zhe Ma. Enhancing representation of spiking neural networks via similarity-sensitive contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16926–16934, 2024. 2

[62] 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*, pages 11062–11070, 2021. 7, 8

[63] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023. 1, 3, 5, 7, 8

[64] 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. 1, 3, 4, 5, 6, 7, 8

[65] Zhaokun Zhou, Kaiwei Che, Wei Fang, Keyu Tian, Yuesheng Zhu, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *arXiv preprint arXiv:2401.02020*, 2024. 1, 3, 4, 6

[66] Shihao Zou, Yuxuan Mu, Xinxin Zuo, Sen Wang, and Li Cheng. Event-based human pose tracking by spiking spatiotemporal transformer, 2023. 1