Towards Training-Free and Accurate ANN-to-SNN Conversion via Activation-Aware Redistribution

Honglin Cao[1], Shuai Wang[1,✉], Zijian Zhou[1], Ammar Belatreche[2], Wenjie Wei[1], Yu Liang[1], Yu Yang[1], Rui Xi[1], Malu Zhang[1,3], Haizhou Li[3,4,5]

[1]University of Electronic Science and Technology of China
[2]Northumbria University
[3]Shenzhen Loop Area Institute
[4]The Chinese University of Hong Kong, Shenzhen
[5]National University of Singapore
✉wangshuai718@std.uestc.edu.cn

Abstract

Conversion represents an effective approach for obtaining low-power models by transforming Artificial Neural Networks (ANNs) into event-driven Spiking Neural Networks (SNNs) without additional training. However, existing training-free conversion methods often incur substantial conversion errors. Here, we first reveal that these conversion errors primarily arise from a distributional mismatch, as the activation distributions of ANNs exhibit channel-wise shifts and scaling, whereas spike rates lack corresponding channel-specific characteristics. To address this limitation, we propose Adaptive Integrate-and-Fire (AIF) neurons with channel-specific thresholds and membrane-potential offsets that dynamically adjust spike rates. These parameters are optimized to jointly minimize conversion errors and maximize information entropy, enabling AIF neurons to capture the activation distribution characteristics of the original ANN. Moreover, AIF neurons can be seamlessly integrated into Transformer architectures with only negligible additional computational cost. Our method achieves state-of-the-art results on multiple vision and natural language processing benchmarks, in particular attaining a notable top-1 accuracy of 85.52% on ImageNet-1K.

Introduction

Spiking Neural Networks (SNNs), which simulate biological neuronal mechanisms, are recognized as third-generation neural networks (Maass 1997) due to their biological plausibility and computational efficiency (Zenke et al. 2021; Kheradpisheh et al. 2019). In contrast to Artificial Neural Networks (ANNs), SNNs emit binary spikes exclusively when membrane potentials exceed specific thresholds. This sparse spiking behavior results in substantially enhanced computational efficiency compared to ANNs (Zhang et al. 2021; Li et al. 2023; Liu et al. 2023; Xu et al. 2024), particularly when implemented on neuromorphic hardware, including TrueNorth (Akopyan et al. 2015), Loihi (Davies et al. 2018) and Tianjic (Pei et al. 2019). Nevertheless, compared with rapid advances in ANNs, the development of large-scale and high-precision SNNs remains largely underexplored (Zhang et al. 2025b).

To address this limitation, two primary training methodologies for SNNs have emerged. The first approach employs surrogate gradients to circumvent spike non-differentiability issues, enabling direct network training through backpropagation (Wu et al. 2018; Neftci, Mostafa, and Zenke 2019; Hu et al. 2024; Zhang et al. 2025a; Wang et al. 2025b; Xiao et al. 2025; Liang et al. 2025; Sun et al. 2025). However, these direct training methods are constrained by imprecise gradient approximations and substantial computational resource requirements, impeding the development of large-scale, high-performance models. The second approach is ANN-to-SNN conversion (Cao, Chen, and Khosla 2015; Wang et al. 2023; You et al. 2024; Bu et al. 2022b; Wang et al. 2024, 2025a; Bu, Li, and Yu 2025; Zhao et al. 2025). This conversion methodology exploits the mathematical equivalence between ANN activation functions and SNN neuronal dynamics, facilitating direct pre-trained weight transfer for inference. However, the practical effectiveness of existing training-free conversion methods is often limited by non-negligible conversion loss.

To mitigate conversion loss, two representative optimization approaches have been proposed: pre-conversion ANN structure optimization (Wang et al. 2022; You et al. 2024) and spiking neuron optimization (Zhang et al. 2023; Bu et al. 2022a; Hao et al. 2023a). While pre-conversion ANN structure optimization has been theoretically proven to reduce conversion error to zero (Bu et al. 2022b), it typically requires additional training of an ANN and cannot directly leverage pre-existing weights from the ANN community. In contrast, spiking neuron optimization calibrates neuron parameters according to ANN activation distributions, making it more resource-efficient. However, this strategy inevitably introduces conversion errors, and effectively reducing these errors remains a key challenge.

Therefore, we first analyze the primary causes of conversion loss and attribute them to discrepancies between ANN activation distributions and SNN spike-rate ranges. Our analysis reveals that ANN activations exhibit channel-wise shifts and scaling, whereas spike rates typically lack such channel-specific configurations, resulting in substantial

conversion loss. To address this issue, we propose an Adaptive Integrate-and-Fire (AIF) neuron model with channel-wise thresholds and membrane-potential offsets. AIF effectively captures the channel-wise characteristics of ANN activation distributions and dynamically adjusts spike rates, thereby minimizing conversion and information loss. Our main contributions are summarized as follows:

- We identify the mismatch between ANN activation distributions and SNN spike-rate ranges, which leads to substantial conversion loss. This mismatch primarily arises because SNN spike-rate ranges lack the channel-specific shifts and scaling present in ANN activations.
- We propose an AIF neuron incorporating channel-wise thresholds and membrane-potential offsets to reduce the mismatch between ANN activations and spike rates. Furthermore, we demonstrate that AIF can be seamlessly integrated into mainstream Transformer architectures.
- We conduct extensive experiments across visual and natural language processing benchmarks. Experimental results demonstrate that our approach achieves state-of-the-art performance with a notable top-1 accuracy of 85.52% on ImageNet-1K.

Related Works

CNN-based Conversion

Early conversion efforts primarily focus on CNN architectures. Cao et al. (Cao, Chen, and Khosla 2015) propose a conversion framework that involves training ANNs with ReLU activation functions and subsequently replacing them with Integrate-and-Fire (IF) neurons. Subsequently, weight scaling techniques (Diehl et al. 2015), soft reset mechanisms (Rueckauer et al. 2017; Han, Srinivasan, and Roy 2020) and membrane potential optimization (Bu et al. 2022a; Hao et al. 2023a) are introduced to reduce the performance gap. Works (Bu et al. 2022b; Hu et al. 2023) theoretically prove that conversion loss can be reduced to zero, though these approaches typically require additional training of pre-conversion ANNs. Zhang et al. (Zhang et al. 2023) develop pioneering dynamic threshold methodologies that adapt during conversion. Hao et al. (Hao et al. 2023b) introduce a novel approach that calibrates "offset spikes" by optimizing initial membrane potential shifts, effectively bridging the performance gap between ANNs and SNNs. Building upon these foundational works, our approach goes beyond considering the discrepancies of ANN activations and SNN spike rates, which further reduces performance gap.

Transformer-based Conversion

With the impressive performance of Transformer architectures across various domains, significant research attention has focused on the conversion of Transformer-like models. Jiang et al. (Jiang et al. 2024) introduce the first training-free method for converting pretrained Transformer models by approximating complex operations such as self-attention in both temporal and spatial dimensions, though notable accuracy discrepancies. Huang et al. (Huang et al. 2024) propose an Expectation Compensation Module to preserve conversion accuracy for non-linear and self-attention, while introducing a Multi-Threshold Neuron to reduce network latency. You et al. (You et al. 2024) establish an exact equivalence between quantized Transformers and converted SNNs, thereby eliminating accuracy degradation. However, this method requires additional training of a quantized pre-conversion ANN Transformer. Building upon these pioneering works, our paper aims to develop a high-accuracy SNN Transformer without pretraining an ANN.

Preliminaries

Neurons in ANNs

The update process of neurons in ANNs can be represented as:

$$\mathbf{a}^l = f(\mathbf{W}^l \mathbf{a}^{l-1}), \quad 1 \leq l \leq n, \tag{1}$$

where $\mathbf{W}^l$ denotes the weight matrix of the l-th layer, $\mathbf{a}^l$ represents the activations propagated through the network, and $f(\cdot)$ signifies the activation function. A bias term b^l may optionally be incorporated into the pre-activation.

Neurons in SNNs

Specifically, the neuronal dynamics of the IF neuron at layer l during time step t can be mathematically expressed as:

$$\mathbf{v}^l_{temp}(t+1) = \mathbf{v}^l(t) + \mathbf{W}^l \mathbf{s}^{l-1}(t)\theta^{l-1}, \tag{2}$$

$$\mathbf{s}^l(t+1) = H(\mathbf{v}^l_{temp}(t+1), \theta^l), \tag{3}$$

$$\mathbf{v}^l(t+1) = \mathbf{v}^l_{temp}(t+1) - \mathbf{s}^l(t+1)\theta^l, \tag{4}$$

where $\mathbf{v}^l_{temp}(t+1)$ represents the membrane potential preceding spike generation, $\mathbf{v}^l(t)$ is the membrane potential, $\mathbf{s}^l(t)$ denotes the spike activation value at time step t, θ^l is the threshold for layer l, and $H(\cdot)$ is the Heaviside step function, defined as:

$$H(\mathbf{v}^l_{temp}(t+1), \theta^l) = \begin{cases} 1, & \text{if } \mathbf{v}^l_{temp}(t+1) \geq \theta^l, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Following established methodologies, soft-reset mechanism is employed to update the membrane potential, wherein θ^l is subtracted when a neuron triggers a spike. The neuronal dynamics in Transformer architectures exhibit slight variations from this approach, as detailed in the **Appendix A**.

ANN-to-SNN Conversion

To represent the rich floating-point activations of ANNs using binary spike values in SNNs, we accumulate neuronal activity over T time steps. By iterating through Equation (2)-(4) for T iterations, we obtain:

$$\mathbf{v}^l(T) - \mathbf{v}^l(0) = \theta^{l-1}\mathbf{W}^l \sum_t^T \mathbf{s}^{l-1}(t) - \theta^l \sum_t^T \mathbf{s}^l(t), \tag{6}$$

$$\begin{aligned} \sum_t^T \mathbf{s}^l(t) &= \frac{\theta^{l-1}\mathbf{W}^l \sum_t^T \mathbf{s}^{l-1}(t) - \mathbf{v}^l(T) + \mathbf{v}^l(0)}{\theta^l} \\ &\approx \text{clip}\left(\left\lfloor \frac{\mathbf{a}^l}{\theta^l} \right\rfloor, 0, T\right), \end{aligned} \tag{7}$$

where $\lfloor \cdot \rfloor$ represents the floor function, and $\text{clip}(\cdot)$ constrains values to the interval $[0, T]$. This approach enables SNNs to approximate ANN behaviors.

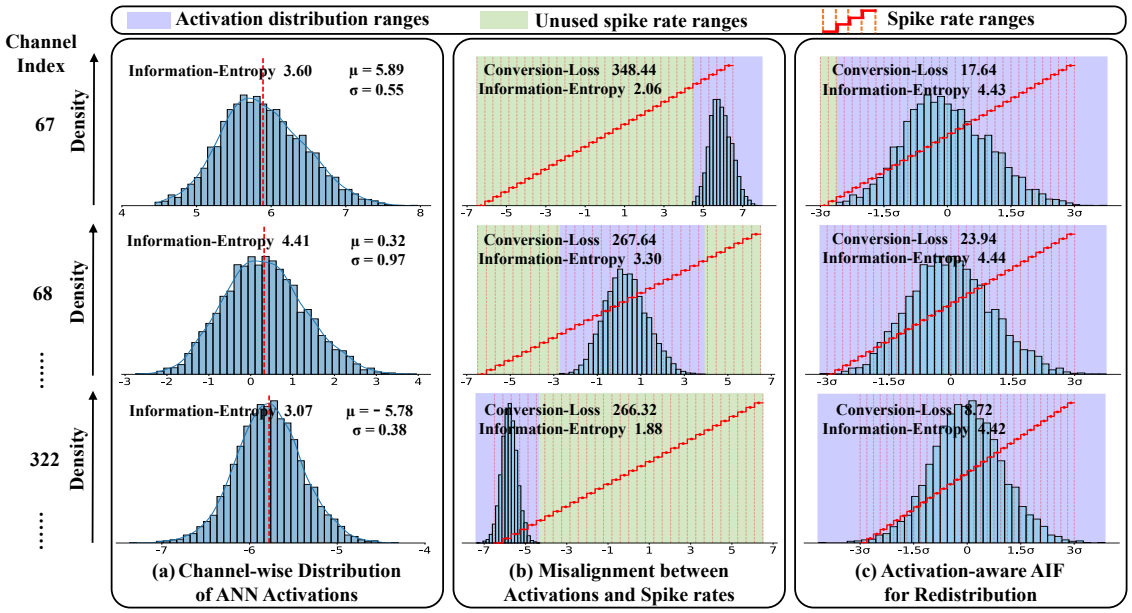

Figure 1: Distribution of ANN activation values and SNN spike rates in ViT-B/16 conversion. (a) Activations across different channels exhibit distinct shifts and scalings. (b) Spike rate ranges lack such channel-specific characteristics and are configured in a layer-wise manner. (c) AIF aligns spike rates with ANN activations.

Method

Challenge Analysis

The aforementioned conversion method often fails to accurately approximate ANN activations. We attribute this failure to neglecting the role of normalization layers in Transformer architectures. By sampling activation values from Vision Transformer models, as illustrated in **Figure 1(a)**, we identify two critical phenomena:

Intra-channel normality: Activation values within individual channels consistently approximate normal distributions, enabling efficient statistical characterization via Gaussian parameters. **Inter-channel distributional diversity**: Distribution parameters across different channels exhibit significant variations in both means and variances, reflecting distinct offsets and ranges. Outliers are predominantly concentrated within a subset of channels.

However, the parameters of the IF neurons are configured in a layer-wise manner. This leads to a mismatch between the SNN spike-rate ranges and the ANN activation distributions, as shown in **Figure 1(b)**. This mismatch results in significant conversion and information loss. To quantify the conversion loss caused by this distributional discrepancy, we define the following loss function:

$$\mathbf{L}_{con} = \left\| \frac{\theta^l}{T} \text{clip}\left(\left\lfloor \frac{\mathbf{a}_c^l \cdot T}{\theta^l} \right\rfloor, -T, T \right) - \mathbf{a}_c^l \right\|_2, \quad (8)$$

where $\mathbf{a}_c^l$ represents the activations of channel c in layer l. The conversion loss results are presented in **Figure 1**. This conversion method, which neglects inter-channel activation differences, leads to substantial discrepancies between spike representations and ANN activation distributions, resulting in considerable conversion loss. Additionally, we employ information entropy (Guo et al. 2024), which enables the assessment of information capacity, to quantify the information loss.

Proposition 1. *For a continuous random variable X with probability density function $p(x)$, the information entropy $\mathcal{H}(X)$ is defined as:*

$$\mathcal{H}(X) = -\int p(x)\log(p(x))dx, \quad (9)$$

When X follows a normal distribution with mean μ and variance σ^2, i.e., $X \sim \mathcal{N}(\mu, \sigma^2)$, and X is quantized into T intervals of width δ, the discrete information entropy of X equals:

$$\mathcal{H}(X) = \frac{1}{2}\log(2\pi e\sigma^2) - \log(\delta), \quad (10)$$

where e is the base of the natural logarithm. The detailed proof can be found in **Appendix B**.

We leverage **Proposition 1** to analytically determine the theoretical entropy of ANN activation distributions. Conversely, the information entropy of the converted SNN is computed through direct numerical integration, as the converted distributions cannot be adequately approximated by normal distributions. The results shown in **Figure 1** reveal substantial information degradation during conversion. Neurons in channels with small μ generate insufficient spikes, resulting in wasted spike rate ranges at both extremes, while neurons with large μ produce excessive spikes, creating homogeneous representations. These findings underscore the importance of developing an activation-aware methodology that appropriately matches spike rate ranges with activation distributions.

Adaptive Integrate-and-Fire Neuron

Neuronal dynamics We introduce the Adaptive Integrate-and-Fire (AIF) neuron as an activation-aware redistribution approach to address this challenge, specifically aimed at reducing information loss and conversion loss. The AIF incorporates channel-specific parameters: a pre-membrane potential offset $\mathbf{Z}^l = (z_1^l, z_2^l, \ldots, z_c^l) \in \mathbb{R}^c$ and an activation threshold $\mathbf{\Theta}^l = (\theta_1^l, \theta_2^l, \ldots, \theta_c^l) \in \mathbb{R}_+^c$. The neuronal dynamics can be described as:

$$\mathbf{v}_{temp}^l(t+1) = \mathbf{v}^l(t) + \mathbf{W}^l \mathbf{s}^{l-1}(t) \odot \mathbf{\Theta}^{l-1} - \mathbf{Z}^l, \quad (11)$$

$$\mathbf{s}^l(t+1) = H(\mathbf{v}_{temp}^l(t+1), \mathbf{\Theta}^l), \quad (12)$$

$$\mathbf{v}^l(t+1) = \mathbf{v}_{temp}^l(t+1) - \mathbf{s}^l(t+1) \odot \mathbf{\Theta}^l, \quad (13)$$

where $\odot$ represents the Hadamard product. As illustrated in **Figure 1(c)**, our approach first shifts pre-membrane potential of each channel to the middle of the spike rate ranges, then performs channel-wise scaling (which can also be considered as scaling the threshold), ensuring that the spike rate ranges optimally match the activation distributions, thereby reducing conversion loss and enhancing information representation capacity.

Theoretical analysis of AIF neuron's parameters Due to the channel-wise diversity in ANN activations, AIF introduces channel-wise offset $\mathbf{Z}^l$ and channel-wise threshold $\mathbf{\Theta}^l$. Directly optimizing $\mathbf{Z}^l$ and $\mathbf{\Theta}^l$ to simultaneously maximize information entropy and minimize conversion loss is impractical. Therefore, we present below how to search for optimal z_c^l and θ_c^l on a per-channel basis to maximize information entropy while minimizing conversion loss.

Proposition 2. *The information entropy of AIF spikes is maximized when the offset z_c^l is set to the mean μ of the ANN activation distribution and the threshold θ_c^l is set to half of the truncation range for each channel c.*

Proof 2. When the ANN distribution and SNN spike rate ranges are optimally matched, and the conversion interval $\delta \to 0$, we can formulate that the random variable X_S (cumulative sum of spikes) distributed within the interval $[\alpha, \alpha + \Delta]$ converges to a truncated Gaussian measure $X_S \sim \mu_S$, where $\mu_S(dx) = \frac{p(x)}{\Psi} dx$, and we define:

$$\Psi(\alpha, \Delta; \mu, \sigma) = \int_\alpha^{\alpha+\Delta} p(x)dx \quad (14)$$

$$= \Phi\left(\frac{\alpha + \Delta - \mu}{\sigma}\right) - \Phi\left(\frac{\alpha - \mu}{\sigma}\right) \quad (15)$$

where $p(x)$ denotes the probability density function of the ANN random variable $X \sim \mathcal{N}(\mu, \sigma^2)$, Φ represents the cumulative distribution function of standard normal distribution, and Ψ denotes the normalization coefficient. Consequently, the information entropy of the truncated Gaussian measure $X_S \sim \mu_S$ can be expressed as:

$$\begin{aligned}\mathcal{H}(X_S) &= -\int_\alpha^{\alpha+\Delta} \frac{p(x)}{\Psi} \log\left(\frac{p(x)}{\Psi}\right) dx \\ &= -\frac{1}{\Psi}\int_\alpha^{\alpha+\Delta} p(x)\log(p(x))dx + \log(\Psi) \quad (16) \\ &= -\frac{1}{\Psi}\mathbb{E}_{x\in[\alpha,\alpha+\Delta]}[\log(p(x))] + \log(\Psi).\end{aligned}$$

To maximize the information entropy of X_S, we first optimize $log(\Psi)$. Applying the Leibniz integral rule, we establish the following variational equality:

$$\frac{\partial \Psi}{\partial \alpha} = \frac{\partial}{\partial \alpha}\int_\alpha^{\alpha+\Delta} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = 0 \quad (17)$$

Since the exponential function is strictly monotonically increasing, this equality holds if and only if the arguments of the exponential terms are equal: $(\alpha + \Delta - \mu)^2 = (\alpha - \mu)^2$, which yields $\alpha = \mu - \frac{\Delta}{2}$ and consequently $\alpha + \Delta = \mu + \frac{\Delta}{2}$. At this stationary point, the normalization constant Ψ attains its maximum, thereby optimizing the logarithmic term $\log(\Psi)$ in the entropy functional. For the expectation term in the entropy expression, we demonstrate that it reaches its maximum when the integration domain exhibits symmetry with respect to μ (i.e., $[\alpha, \alpha + \Delta] = [\mu - \frac{\Delta}{2}, \mu + \frac{\Delta}{2}]$), as proven in the **Appendix C**. This confirms that the offset z_c^l should be set to the mean of the ANN activations, and the threshold θ_c^l should be set to $\frac{\Delta}{2}$ to maximize information entropy.

Furthermore, we consider the impact of conversion loss and formulate a constrained optimization problem to search for optimal parameters:

$$\min_{z_c^l, \theta_c^l} \frac{1}{N}\left\| \frac{\theta_c^l}{T} \cdot \text{clip}\left(\left\lfloor \frac{T(\mathbf{a}_c^l - z_c^l)}{\theta_c^l} \right\rfloor, -T, T\right) + z_c^l - \mathbf{a}_c^l \right\|_2, \quad (18)$$

where N represents the number of elements in $\mathbf{a}_c^l$. It is noteworthy that this optimization formulation incorporates non-differentiable operators $\lfloor \cdot \rfloor$ and clip$(\cdot)$ within an ℓ_2-norm objective, precluding the direct application of gradient-based methods for locating extremal points. Therefore, guided by **Proposition 2**, we employ a grid search approach, selecting candidate points near the mean for the offset z_c^l. To address the presence of extreme values, we apply the "3σ" principle and evaluate candidate points near 3σ for the threshold θ_c^l. Grid search only needs a few minutes, and the detailed runtime of grid search is discussed later. Through this approach, we endeavor to minimize conversion loss while maximizing information entropy, thereby deriving optimal activation-aware redistribution parameters.

Conversion Framework with AIF Neuron

Given that AIF introduces additional channel-wise offsets and thresholds, it is essential to demonstrate that our method produces results equivalent to the original calculations while preserving spike-driven characteristics. As illustrated in **Figure 2**, mainstream network architectures incorporate three fundamental types of operation: Weight-Activation (WA) multiplication, Activation-Activation (AA) multiplication, and Non-linear operations.

WA-multiplication For WA multiplication, such as the convolutional layers and FC layers in the MLP of Transformers, we have the following correspondence:

$$\mathbf{a}^l \approx \frac{1}{T}\sum_t^T \mathbf{s}^l(t) \odot \mathbf{\Theta}^l + \mathbf{Z}^l, \quad (19)$$

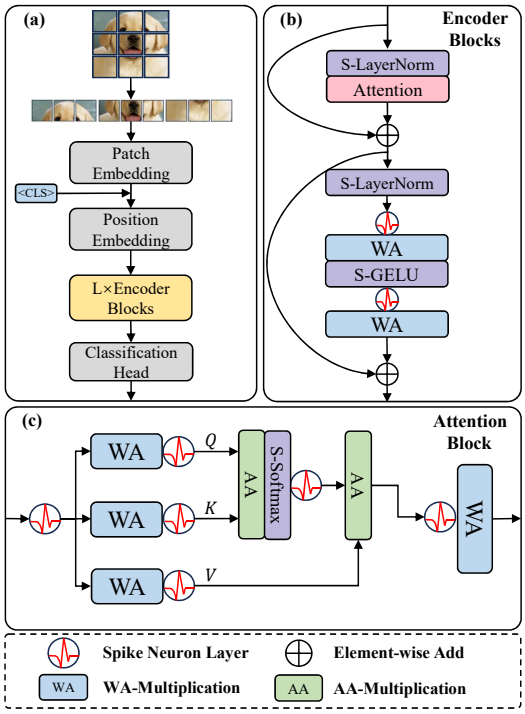

Figure 2: Training-free conversion framework with AIF neuron. (a) Training-free conversion framework on ViT-like architecture. (b) Transformer encoder blocks incorporate WA, AA operations and diverse non-linear functions. Each attention and MLP block is preceded by a normalization layer. (c) The expanded schematic of the Attention block.

$$\mathbf{W}^l \otimes \mathbf{a}^l \approx \frac{1}{T}\sum_t^T (\mathbf{W}^l \odot \mathbf{\Theta}^l) \otimes \mathbf{s}^l(t) + \mathbf{W}^l \otimes \mathbf{Z}^l, \quad (20)$$

where $\mathbf{W}^l \otimes \mathbf{Z}^l$ can be pre-computed and incorporated into the bias term, while the weights are scaled using $\mathbf{\Theta}^l$.

AA-multiplication AA multiplication primarily exists in the attention calculations within Transformer blocks. For ANN activation values $\mathbf{q}^l, \mathbf{k}^l$ and their corresponding SNN spike values $\mathbf{q}_s^l, \mathbf{k}_s^l$, the following correspondence exists:

$$\mathbf{q}^l \approx \frac{1}{T}\sum_t^T \hat{\mathbf{q}}_s^l(t) = \frac{1}{T}\sum_t^T (\mathbf{q}_s^l(t) \odot \Theta_q^l + \mathbf{Z}_q^l), \quad (21)$$

$$\mathbf{k}^l \approx \frac{1}{T}\sum_t^T \hat{\mathbf{k}}_s^l(t) = \frac{1}{T}\sum_t^T (\mathbf{k}_s^l(t) \odot \Theta_k^l + \mathbf{Z}_k^l). \quad (22)$$

Therefore, when calculating attention, we ensure equivalence of values through the following method:

$$\mathbf{Q}^l(t) = \sum_i^t \hat{\mathbf{q}}_s^l(i), \mathbf{K}^l(t) = \sum_i^t \hat{\mathbf{k}}_s^l(i), \quad (23)$$

$$\mathbf{q}^l \otimes \mathbf{k}^l \approx \frac{1}{T^2}\sum_t^T (-\hat{\mathbf{q}}_s^l(t) \otimes \hat{\mathbf{k}}_s^l(t) + \mathbf{Q}^l(t) \otimes \hat{\mathbf{k}}_s^l(t) + \hat{\mathbf{q}}_s^l(t) \otimes \mathbf{K}^l(t)), \quad (24)$$

where $\mathbf{Q}^l(t), \mathbf{K}^l(t)$ represent the cumulative sum of spikes over the first t time steps. The first term in Equation (24) can be calculated using the following formula:

$$\hat{\mathbf{q}}^l \otimes \hat{\mathbf{k}}^l = \mathbf{q}_s^l \otimes \mathbf{k}_s^l \odot (\Theta_q^l \odot \Theta_k^l) + \mathbf{q}_s^l \otimes \mathbf{Z}_k^l \odot \Theta_q^l + \mathbf{Z}_q^l \otimes \mathbf{k}_s^l \odot \Theta_k^l + \mathbf{Z}_q^l \otimes \mathbf{Z}_k^l, \quad (25)$$

where $\Theta_q^l \odot \Theta_k^l$ and $\mathbf{Z}_q^l \otimes \mathbf{Z}_k^l$ can be pre-computed, and the Hadamard product of the scaling factors does not participate in matrix operations, rendering its computational overhead negligible. Beyond these factors, the remaining operations $\mathbf{q}_s^l \otimes \mathbf{k}_s^l, \mathbf{q}_s^l \otimes \mathbf{Z}_k^l, \mathbf{Z}_q^l \otimes \mathbf{k}_s^l$ are all spike-driven operations. The second term $\mathbf{Q}^l(t) \otimes \hat{\mathbf{k}}_s^l(t)$ and third term $\hat{\mathbf{q}}_s^l(t) \otimes \mathbf{K}^l(t)$ follow computational patterns similar to the first term, as demonstrated in the **Appendix E**.

Non-linear operations To implement traditional GELU, Softmax, and LayerNorm operations within the SNN paradigm, we have developed Spike-GELU (S-GELU), Spike-Softmax (S-Softmax), and Spike-LayerNorm (S-LayerNorm) operators similar to methodologies employed in (You et al. 2024; Huang et al. 2024). The procedural mechanism for these spiking operators can be represented as follows:

$$\mathbf{S}_t = \sum_{i=0}^t \mathbf{s}_i, \quad \mathbf{O}_t = \sigma(\frac{\mathbf{S}_t}{t}) \cdot t - \sigma(\frac{\mathbf{S}_{t-1}}{t-1}) \cdot (t-1). \quad (26)$$

In this formulation, σ denotes the GELU, Softmax or LayerNorm function. The variables $\mathbf{s}_i$ represent the respective input of the operator at each discrete time step. The term $\mathbf{S}_t$ indicates the temporal summation of inputs across t time steps. $\mathbf{O}_t$ represents the functional output. Functional equivalence between spike-based non-linear operations and the standard counterpart is achieved by performing temporal summation of the spike outputs across all time-steps.

Experiment

In this section, we validate the efficacy of our proposed methodology. The detailed experimental setup and extended experimental results can be found in **Appendix F**.

Comparative Analysis of Vision Datasets

ImageNet Our experimental results on ImageNet demonstrate significant advancements in performance across diverse architectures as shown in Table 1. When applied to Vision Transformers, our approach delivers 82.83% accuracy with ViT-B and 85.52% with ViT-L at 64 time steps, outperforming conversion methods like ECMT (84.6% with 8 time steps) and Spike-ZIP (83.82% with 64 time steps). These results demonstrate the efficacy of our approach when applied to large-scale Transformer architectures and complex image datasets.

CIFAR Table 2 presents a comprehensive comparison of various SNN approaches on the CIFAR10 and CIFAR100 datasets. Our proposed method on ViT-B/16 achieves superior performance with accuracy rates of 98.7% and 91.8% on CIFAR10 and CIFAR100, respectively, at a time step of

Method	Architecture	Type	Training-free	Param(M)	T	Acc(%)
(Dosovitskiy et al. 2020)	ViT-S	ANN	-	22.05	-	81.14
	ViT-B	ANN	-	86.57	-	84.51
	ViT-L	ANN	-	304.33	-	85.82
(Hao et al. 2023a)	VGG-16	ANN-to-SNN	✗	138.42	64	69.43
(Bu et al. 2022b)	VGG-16	ANN-to-SNN	✗	138.42	64	72.85
(Oh and Lee 2024)	VGG-16	ANN-to-SNN	✓	138.42	64	67.04
	ResMLP-S24	ANN-to-SNN	✓	30.00	128	77.99
(Yao et al. 2025)	E-SpikeFormer	Direct-Training	✗	52.0	4	80.6
		Direct-Training	✗	83.0	8	84.0
		Direct-Training	✗	173.0	8	85.1
(Huang et al. 2024)	ViT-S	ANN-to-SNN	✓	22.05	8(10)	76.03(77.07)
	ViT-B	ANN-to-SNN	✓	86.57	8(10)	79.40(80.12)
	ViT-L	ANN-to-SNN	✓	304.33	4(8)	83.20(84.60)
(You et al. 2024)	SViT-S-32Level	ANN-to-SNN	✗	22.05	64	81.45
	SViT-B-32Level	ANN-to-SNN	✗	86.57	64	82.71
	SViT-L-32Level	ANN-to-SNN	✗	304.33	64	83.82
Ours	ViT-S	ANN-to-SNN	✓	22.05	64	78.74
	ViT-B	ANN-to-SNN	✓	86.57	64	**82.83**
	ViT-L	ANN-to-SNN	✓	304.33	64	**85.52**

Table 1: Comparative experimental results on the ImageNet dataset. All inference was conducted with an image size of 224×224. Training-free indicates whether the method requires direct training or training a pre-conversion ANN.

Method	Type	T	Acc.(%)	
			CIFAR10	CIFAR100
(Zhou et al. 2023a)	DT	4	95.8	79.2
(Yao et al. 2024)	DT	4	95.6	78.4
(Zhou et al. 2023b)	TL	4	97.0	83.8
(Shi, Hao, and Yu 2024)	TL	4	97.4	86.0
(Huang et al. 2024)	A2S	8	98.6	91.5
(Wang et al. 2023)	A2S	64	96.3	85.4
(Jiang et al. 2024)	A2S	32	95.5	84.2
(You et al. 2024)	A2S	32	98.7	89.7
Ours	A2S	32	**98.7**	**91.8**
		64	**98.9**	**92.0**

Table 2: Comparative results on CIFAR datasets. DT, TL and A2S indicate direct training, transfer learning and ANN-to-SNN conversion methodologies, respectively.

32, surpassing previous approaches. Increasing the time step to 64 further enhances performance to 98.9% and 92.0%, respectively, without requiring any additional training. The superior performance demonstrates the robust generalization capability of our method across smaller datasets.

Comparative Analysis of NLP Tasks

Table 3 presents a comprehensive evaluation of our method against BERT and directly trained methods on the GLUE dataset. A significant advantage of our approach lies in the direct conversion of fine-tuned ANN BERT weights without training a pre-conversion BERT. In comparison to state-of-the-art directly trained SNNs, our method achieves a substantial performance improvement of 6.2% in average accuracy. When configured with 64 time steps, our ANN-to-SNN conversion demonstrates remarkable fidelity, experiencing only a negligible 0.2% accuracy reduction compared to the original ANN BERT model (82.7% versus 82.9%). These empirical findings highlight the potential for deploying energy-efficient SNNs in sophisticated natural language processing tasks and large language models.

Ablation Study

Table 4 presents a systematic evaluation of channel-wise threshold and channel-wise offset strategies across varying time steps. The baseline model exhibits substantial accuracy degradation across all time steps. Employing either CW-threshold or CW-offset individually fails to simultaneously maximize information entropy and minimize conversion loss, resulting in significant performance degradation at lower time steps (T=4, T=8). The proposed AIF approach, which integrates both strategies, consistently outperforms all other methods across all time steps. Most remarkably, AIF maintains substantial performance even at extremely compressed temporal windows (T=4 and T=8).

Analysis of Grid Search Runtime

The complexity of grid search is $O(L \times B \times C \times N \times M)$, which is related to network depth L, batch size B, number of channels C, sequence length N, and number of candidate points M. Since both the number of network parameters and input dimensions are fixed, the actual runtime of the search remains essentially constant across multiple experiments, as shown in Table 5. Grid search performs calibration using only a tiny portion of the training set, and the calibrated

Model	Type	Time	$\text{MNLI}_{\text{m/mm}}$	QQP_{F1}	QNLI	SST-2	CoLA	STS-B	MRPC_{F1}	RTE	Avg.
$\text{BERT}_{\text{base}}$	ANN	-	84.3/84.7	88.4	91.4	92.9	57.0	88.7	89.2	69.0	82.9
LIF-BERT*	Direct Training	4	56.8/55.2	70.0	60.6	80.6	14.6	20.0	82.3	53.8	54.9
SpikingFormer*	Direct Training	4	70.2/70.6	80.9	79.5	83.9	12.8	77.0	83.0	62.1	68.9
SpikeBERT	Direct Training	4	71.4/71.0	68.2	66.4	85.4	16.9	18.7	82.0	57.5	59.7
$\text{SpikeLM}_{\text{6L}}$	Direct Training	4	75.1/75.3	83.5	84.6	87.4	33.7	84.5	86.5	64.3	75.0
SpikeLM	Direct Training	4	77.1/77.2	83.9	85.3	87.0	38.8	84.9	85.7	69.0	76.5
Ours	ANN-to-SNN	8	79.3/78.3	84.7	85.2	89.1	46.0	86.0	87.0	58.1	**77.1**
	ANN-to-SNN	16	83.7/83.9	88.0	89.8	92.1	56.1	87.9	90.3	63.2	**81.6**
	ANN-to-SNN	64	84.8/85.1	88.6	91.1	92.4	56.5	88.7	89.6	67.9	**82.7**

Table 3: Comparative results on the GLUE dataset. The results of BERT, SpikeBERT and SpikeLM are obtained from (Devlin et al. 2019; Lv et al. 2023; Xing et al. 2024), respectively. * Results are obtained from SpikeLM.

Method	T=4	T=8	T=16	T=32	T=64
Baseline	0.09	0.42	27.49	63.45	73.17
Loss (%)	-85.73	-85.40	-58.33	-22.37	-12.65
+CW-threshold	1.73	53.15	74.69	81.60	84.27
Loss (%)	-84.90	-32.67	-11.13	-4.22	-1.55
+CW-offset	0.17	66.34	81.12	83.69	84.49
Loss (%)	-85.65	-19.48	-4.70	-2.13	-1.33
AIF	**58.18**	**82.36**	**84.96**	**85.33**	**85.52**
Loss (%)	**-27.64**	**-3.46**	**-0.86**	**-0.49**	**-0.30**

Table 4: Ablation study on channel-wise threshold (CW-threshold) and channel-wise offset (CW-offset). Loss denotes the accuracy gap relative to ViT-Large, which achieves 85.82% top-1 accuracy on ImageNet. The baseline implements layer-wise thresholds without offset adjustments.

Model	Dataset	Param(M)	Runtime(s)
ViT-S	ImageNet	22.1	27.2
ViT-B	ImageNet	86.6	54.1
ViT-L	ImageNet	304.3	150.5
BERT	GLUE-RTE	110.0	80.1

Table 5: Empirical runtime of grid search across different model architectures. All reported runtimes are averaged over three independent runs.

neuron thresholds and offset data can be stored for reuse, eliminating the need for calibration during each inference. Consequently, this time overhead is negligible compared to training and inference.

Analysis of Grid Search Approach

We plot conversion loss and information entropy distributions within the ranges $z \in [\mu - 6\sigma, \mu + 6\sigma]$ and $\theta \in [0, 6\sigma]$, as shown in **Figure 3**. The conversion loss exhibits a valley-like structure centered at the mean, indicating that using the mean as an offset minimizes conversion loss when the threshold is low. As the threshold increases, the relative impact of the offset diminishes, though offsets near the mean still significantly reduce conversion loss. Regarding information entropy, it displays a peak-like structure centered at the mean, demonstrating that using the mean as an offset substantially enhances information representation. The influence of offsets decreases with increasing threshold values. Notably, we observe that the optimal offset values are very close to the mean, with optimal threshold values approximating 3σ, thus validating the effectiveness of our grid search approach.

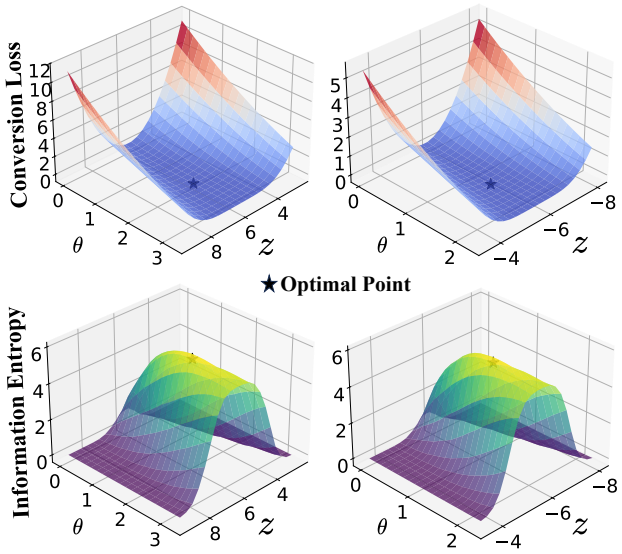

Figure 3: Conversion loss plane and information entropy plane of grid search. Left: Channel-67, $\mu = 5.89, \sigma = 0.55$; Right: Channel-322, $\mu = -5.78, \sigma = 0.38$.

Conclusion

This work identifies a critical mismatch between ANN activation distributions and SNN spike-rate ranges that leads to a substantial performance degradation during conversion. To address this problem, we develop an Adaptive Integrate-and-Fire (AIF) neuron with channel-wise thresholds and membrane-potential offsets. We optimize these parameters by jointly maximizing information entropy and minimizing conversion loss. AIF can be seamlessly integrated into Transformer architectures, substantially improving conversion performance without requiring the training of a separate pre-conversion ANN. Our work lays the foundation for future high-performance, large-scale SNN conversion.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62220106008 and 62576080), in part by the Shenzhen Science and Technology Program (Shenzhen Key Laboratory, Grant No. ZDSYS20230626091302006) and in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams, Grant No. 2023ZT10X044, in part by the State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Grant No. SKLBI-K2025010.

References

Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.-J.; et al. 2015. 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.

Bu, T.; Ding, J.; Yu, Z.; and Huang, T. 2022a. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 11–20.

Bu, T.; Fang, W.; Ding, J.; DAI, P.; Yu, Z.; and Huang, T. 2022b. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. In *International Conference on Learning Representations*.

Bu, T.; Li, M.; and Yu, Z. 2025. Inference-Scale Complexity in ANN-SNN Conversion for High-Performance and Low-Power Applications. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 24387–24397.

Cao, Y.; Chen, Y.; and Khosla, D. 2015. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113: 54–66.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Diehl, P. U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.-C.; and Pfeiffer, M. 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, 1–8. ieee.

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

Guo, Y.; Chen, Y.; Liu, X.; Peng, W.; Zhang, Y.; Huang, X.; and Ma, Z. 2024. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 12244–12252.

Han, B.; Srinivasan, G.; and Roy, K. 2020. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13558–13567.

Hao, Z.; Bu, T.; Ding, J.; Huang, T.; and Yu, Z. 2023a. Reducing ann-snn conversion error through residual membrane potential. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11–21.

Hao, Z.; Ding, J.; Bu, T.; Huang, T.; and Yu, Z. 2023b. Bridging the Gap between ANNs and SNNs by Calibrating Offset Spikes. In *The Eleventh International Conference on Learning Representations*.

Hu, Y.; Deng, L.; Wu, Y.; Yao, M.; and Li, G. 2024. Advancing spiking neural networks toward deep residual learning. *IEEE transactions on neural networks and learning systems*, 36(2): 2353–2367.

Hu, Y.; Zheng, Q.; Jiang, X.; and Pan, G. 2023. Fast-snn: Fast spiking neural network by converting quantized ann. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 14546–14562.

Huang, Z.; Shi, X.; Hao, Z.; Bu, T.; Ding, J.; Yu, Z.; and Huang, T. 2024. Towards High-performance Spiking Transformers from ANN to SNN Conversion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10688–10697.

Jiang, Y.; Hu, K.; Zhang, T.; Gao, H.; Liu, Y.; Fang, Y.; and Chen, F. 2024. Spatio-temporal approximation: A training-free snn conversion for transformers. In *The Twelfth International Conference on Learning Representations*.

Kheradpisheh, S. R.; Tavanaei, A.; Ghodrati, M.; Masquelier, T.; and Maida, A. 2019. Deep learning in spiking neural networks. *Neural Networks*, 111: 47–63.

Li, G.; Deng, L.; Tang, H.; Pan, G.; Tian, Y.; Roy, K.; and Maass, W. 2023. Brain inspired computing: A systematic survey and future trends. *Authorea Preprints*.

Liang, Y.; Wei, W.; Belatreche, A.; Cao, H.; Zhou, Z.; Wang, S.; Zhang, M.; and Yang, Y. 2025. Towards Accurate Binary Spiking Neural Networks: Learning with Adaptive Gradient Modulation Mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1402–1410.

Liu, H.; Chen, Y.; Zeng, Z.; Zhang, M.; and Qu, H. 2023. A Low Power and Low Latency FPGA-Based Spiking Neural Network Accelerator. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Lv, C.; Li, T.; Xu, J.; Gu, C.; Ling, Z.; Zhang, C.; Zheng, X.; and Huang, X. 2023. SpikeBERT: A Language Spikformer Learned from BERT with Knowledge Distillation. *arXiv preprint arXiv:2308.15122*.

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

Neftci, E. O.; Mostafa, H.; and Zenke, F. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6): 51–63.

Oh, H.; and Lee, Y. 2024. Sign Gradient Descent-based Neuronal Dynamics: ANN-to-SNN Conversion Beyond ReLU Network. In *Forty-first International Conference on Machine Learning*.

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

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

Shi, X.; Hao, Z.; and Yu, Z. 2024. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5610–5619.

Sun, Q.; Lu, C.; Chen, W.; Wei, W.; Wang, J.; Zhang, J.; Liu, X.; Ye, Y.; Yang, Y.; and Zhang, M. 2025. Temporal-coded Spiking Transformer. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2616–2624.

Wang, J.; Deng, X.; Wei, W.; Zhang, D.; Wang, S.; Sun, Q.; Zhang, J.; Liu, H.; Xie, N.; and Zhang, M. 2025a. Training-free ann-to-snn conversion for high-performance spiking transformer. *arXiv preprint arXiv:2508.07710*.

Wang, S.; Zhang, M.; Zhang, D.; Belatreche, A.; Xiao, Y.; Liang, Y.; Shan, Y.; Sun, Q.; Zhang, E.; and Yang, Y. 2025b. Spiking vision transformer with saccadic attention. *arXiv preprint arXiv:2502.12677*.

Wang, Y.; Liu, H.; Zhang, M.; Luo, X.; and Qu, H. 2024. A universal ANN-to-SNN framework for achieving high accuracy and low latency deep Spiking Neural Networks. *Neural Networks*, 174: 106244.

Wang, Y.; Zhang, M.; Chen, Y.; and Qu, H. 2022. Signed Neuron with Memory: Towards Simple, Accurate and High-Efficient ANN-SNN Conversion. In *IJCAI*, 2501–2508.

Wang, Z.; Fang, Y.; Cao, J.; Zhang, Q.; Wang, Z.; and Xu, R. 2023. Masked spiking transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1761–1771.

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

Xiao, Y.; Wang, S.; Zhang, D.; Wei, W.; Shan, Y.; Liu, X.; Jiang, Y.; and Zhang, M. 2025. Rethinking Spiking Self-Attention Mechanism: Implementing a-XNOR Similarity Calculation in Spiking Transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5444–5454.

Xing, X.; Zhang, Z.; Ni, Z.; Xiao, S.; Ju, Y.; Fan, S.; Wang, Y.; Zhang, J.; and Li, G. 2024. SpikeLM: towards general spike-driven language modeling via elastic bi-spiking mechanisms. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Xu, Q.; Fang, X.; Li, Y.; Shen, J.; Ma, D.; Xu, Y.; and Pan, G. 2024. RSNN: Recurrent Spiking Neural Networks for Dynamic Spatial-Temporal Information Processing. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10602–10610.

Yao, M.; Hu, J.; Zhou, Z.; Yuan, L.; Tian, Y.; Xu, B.; and Li, G. 2024. Spike-driven transformer. *Advances in neural information processing systems*, 36.

Yao, M.; Qiu, X.; Hu, T.; Hu, J.; Chou, Y.; Tian, K.; Liao, J.; Leng, L.; Xu, B.; and Li, G. 2025. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

You, K.; Xu, Z.; Nie, C.; Deng, Z.; Guo, Q.; Wang, X.; and He, Z. 2024. SpikeZIP-TF: Conversion is All You Need for Transformer-based SNN. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 57367–57383. PMLR.

Zenke, F.; Bohté, S. M.; Clopath, C.; Comşa, I. M.; Göltz, J.; Maass, W.; Masquelier, T.; Naud, R.; Neftci, E. O.; Petrovici, M. A.; et al. 2021. Visualizing a joint future of neuroscience and neuromorphic engineering. *Neuron*, 109(4): 571–575.

Zhang, A.; Shi, J.; Wu, J.; Zhou, Y.; and Yu, W. 2023. Low latency and sparse computing spiking neural networks with self-driven adaptive threshold plasticity. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhang, D.; Zhang, M.; Wang, S.; Wang, J.; Wei, W.; Ma, Z.; Wang, G.; Yang, Y.; and Li, H. 2025a. Dendritic Resonate-and-Fire Neuron for Effective and Efficient Long Sequence Modeling. *arXiv preprint arXiv:2509.17186*.

Zhang, M.; Wang, J.; Wu, J.; Belatreche, A.; Amornpaisannon, B.; Zhang, Z.; Miriyala, V. P. K.; Qu, H.; Chua, Y.; Carlson, T. E.; et al. 2021. Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE transactions on neural networks and learning systems*, 33(5): 1947–1958.

Zhang, M.; Wei, W.; Zhou, Z.; Liu, W.; Zhang, J.; Belatreche, A.; and Yang, Y. 2025b. Spike-Driven Lightweight Large Language Model With Evolutionary Computation. *IEEE Transactions on Evolutionary Computation*.

Zhao, L.; Huang, Z.; Ding, J.; and Yu, Z. 2025. TTFS-Former: A TTFS-based Lossless Conversion of Spiking Transformer. In *Forty-second International Conference on Machine Learning*.

Zhou, C.; Yu, L.; Zhou, Z.; Ma, Z.; Zhang, H.; Zhou, H.; and Tian, Y. 2023a. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; YAN, S.; Tian, Y.; and Yuan, L. 2023b. Spikformer: When Spiking Neural Network Meets Transformer. In *The Eleventh International Conference on Learning Representations*.