Training-Free ANN-to-SNN Conversion for High-Performance Spiking Transformers

Jingya Wang[1*], Xin Deng[1*], Wenjie Wei[1], Dehao Zhang[1], Shuai Wang[1], Qian Sun[1], Jieyuan Zhang[1], Hanwen Liu[1], Ning Xie[1], Malu Zhang[1,2 †]

[1]University of Electronic Science and Technology of China,
[2]Shenzhen Loop Area Institute
jiangyawang@std.uestc.edu.cn, dengbujiu@std.uestc.edu.cn, maluzhang@uestc.edu.cn

Abstract

Leveraging the event-driven paradigm, Spiking Neural Networks (SNNs) offer a promising approach for energy-efficient Transformer architectures. While ANN-to-SNN conversion avoids the high training cost of directly trained Spiking Transformers, existing approaches still struggle to handle the nonlinear operations within Transformer blocks, and often require additional fine-tuning of pretrained ANNs. To address these limitations, we propose a training-free and high-performance ANN-to-SNN conversion framework tailored for Transformer architectures. Specifically, we introduce a Multi-basis Exponential Decay (MBE) neuron that combines exponential decay with a multi-basis encoding strategy to effectively approximate nonlinear operations, eliminating the need for weight modifications in pretrained ANNs. Extensive experiments across diverse tasks (CV, NLU, NLG) and mainstream Transformer architectures (ViT, RoBERTa, GPT-2) demonstrate that our method achieves near-lossless conversion accuracy with significantly lower latency. This provides a promising pathway for the efficient and scalable deployment of Spiking Transformers in real-world applications.

Introduction

Spiking Neural Networks (SNNs) have garnered attention due to their sparse and spike-driven computing paradigm (Maass 1997). Unlike Artificial Neural Networks (ANNs), SNNs employ sparse binary spikes as information carriers, thereby offering superior energy efficiency for resource-limited devices (Zhang et al. 2025b; Liang et al. 2025; Wei et al. 2025). Recently, several studies (Wang et al. 2025b; Xiao et al. 2025) have integrated high-performance Transformer architectures (Vaswani et al. 2017) into SNNs to jointly exploit their expressive capacity and inherent energy efficiency. These methods substantially improve the performance of SNNs on complex tasks (Wang et al. 2025a,c; Shan et al. 2025; Cai et al. 2025).

To obtain Transformer-based SNNs, two mainstream approaches exist: Direct Training (DT) (Wu et al. 2018) and ANN-to-SNN conversion (A2S) (Diehl et al. 2015). DT employs surrogate gradients (Neftci, Mostafa, and Zenke 2019; Wei et al. 2024; Sun et al. 2025; Zhang et al. 2025a) and backpropagation through time (BPTT) (Hochreiter and Schmidhuber 1997), making the training of large-scale SNNs feasible. However, it suffers from inaccurate gradient approximation and $\mathcal{T}\times$ training overhead. In contrast, A2S establishes an equivalence between the firing rate of Integrate-and-Fire neurons and ReLU functions (Bu et al. 2023), enabling SNNs to inherit the pretrained weights of ANNs with nearly lossless conversion in CNNs. However, complex nonlinear operations in Transformers pose substantial challenges for converting them into SNN architectures.

To address this issue, prior research (Wang et al. 2023) examines nonlinear components in Transformers. According to their computational forms, these nonlinear operations can be categorized into single-variable (e.g., GELU, Tanh) and multi-variable operations (e.g., variable–variable floating-point multiplications, LayerNorm). SpikeZIP-TF (You et al. 2024) replaces single-variable functions and fine-tunes pretrained ANN weights, incurring additional training overhead. STA (Jiang et al. 2024) approximates all nonlinear functions using group operators, which increases inference latency. Thus, there is a need for a conversion method that balances training cost with inference efficiency.

Inspired by the temporal coding mechanisms observed in biological neurons, Few-spikes (FS) neurons (Stöckl and Maass 2021) are proposed. It leverages spike timing and patterns to represent neuronal activation states. This characteristic enables it to achieve near-lossless performance in small-scale CNN-based conversion (Mao et al. 2025). However, as model scale increases, FS-based conversion methods exhibit substantial performance gaps. Moreover, their single-input fitting design limits their ability to handle the multi-variable operations in Transformer architectures.

In this paper, we first theoretically and empirically analyze the key challenges associated with FS–based conversion, particularly in large-scale network architectures. To overcome these challenges, we propose a training-free A2S conversion framework tailored for Transformer architectures. Specifically, we introduce a novel Multi-basis Exponential Decay (MBE) neuron that effectively approximates diverse nonlinear operations, enabling our conversion framework to achieve both low inference latency and near-lossless accuracy. The main contributions are as follows:

[*]These authors contributed equally.
[†]Corresponding author.

- We systematically analyze the incompatibility between FS neurons and Transformer architectures, focusing on excessive dependence on initialization (EDI) and global suboptimality (GSO) problems, which hinder convergence and degrade conversion performance.
- We introduce a MBE neuron with exponential decay strategy and multi-basis encoding method. The decay strategy enables multi-resolution representations, while multi-basis encoding enhances the near-lossless approximation of diverse nonlinear operations.
- We propose an A2S framework based on MBE neurons that efficiently approximates various nonlinear operations in Transformer architectures, including variable-variable FP multiplications, GELU, Softmax, and LayerNorm. It achieves near-lossless conversion with fast inference and require no training on source ANNs.
- Extensive experiments on diverse tasks (CV, NLU, NLG) and architectures (ViT, RoBERTa, GPT-2) demonstrate that our method achieves near-lossless conversion with reduced latency, yielding competitive results among existing Transformer-based A2S methods.

Related Work

Existing conversion paradigms can be categorized into two types based on whether extra training on the source ANNs is required: training-dependent and training-free.

Training-dependent conversion typically requires additional training of ANN based on theoretical ANN-SNN equivalence. Clip-Floor-Shift activations (Bu et al. 2023) replace ReLU to better match spiking neuron behavior. A two-phase training (Ding et al. 2021) first optimizes ANN weights, then adjusts neuron thresholds, while activation range constraints and membrane potential initialization (Bu et al. 2022) help further reduce conversion error. (Wang et al. 2022b) introduces a two-stage approach to handle quantization, pruning, and residual errors. (Jiang et al. 2023) proposes a unified framework treating spike-rate mapping as a differentiable problem. For Transformer architectures, a QANN is integrated with the spatiotemporal properties of spiking neurons for lossless conversion (You et al. 2024). While effective, these methods introduce computational overhead through additional training of source ANNs.

Training-free conversion directly transforms pre-trained ANNs via structure and parameter reuse without fine-tuning. Early works align SNN thresholds with ANN activations via threshold balancing (Diehl et al. 2015; Rueckauer et al. 2017), but longer timesteps are required. Signed spiking neurons (Wang et al. 2022a) support dynamic vision data. In (Jiang et al. 2024), training-free Transformer conversion is achieved using universal group operators and spatial rectification self-attention. SpikedAttention (Hwang et al. 2024) enables spike-driven Transformer A2S via trace-based matrix multiplication and winner-take-all spike shifting, yet retains non-spiking LayerNorm. ECMT (Huang et al. 2024) reduces latency in Transformer SNNs, yet still relies on floating-point operations. While retraining is avoided, these methods often require long timesteps or retain non-spiking components, limiting SNNs' energy efficiency.

Preliminary & Problem Analysis

Preliminary

Inspired by authentic electrophysiological characteristics of human brain neurons, FS neuron is proposed to address the inherent trade-off between spike count and accuracy in A2S (Stöckl and Maass 2021). By introducing learnable parameters including neuron threshold, membrane potential reset value, and spike intensity, FS neuron approximates ANN activation functions with very few spikes. Mathematically, the membrane potential of FS neuron can be computed as:

$$u[t+1] = u[t] - r[t] \cdot s[t], \tag{1}$$

where $t \in [0, T-1]$ denotes timesteps, $u[t]$ represents the membrane potential, $s[t] \in \{0, 1\}$ is the binary spike, and $r[t]$ is the learned reset value. The membrane potential has no decay, and its initial value is typically set to the input from the previous layer, i.e., $u[0] = x$. When the membrane potential exceeds a learned threshold, a spike is generated, which is described as follows:

$$s[t] = \mathcal{H}\left(\left(x - \sum_{t'=0}^{t-1} r[t'] \cdot s[t']\right) - V_{th}[t]\right), \tag{2}$$

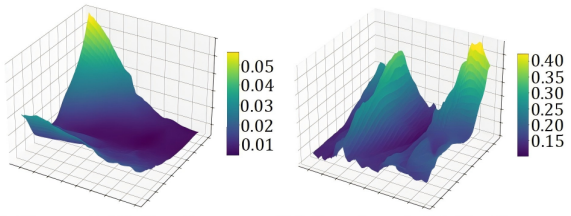
where $\mathcal{H}(\cdot)$ is the Heaviside step function and $V_{th}[t]$ is the learned firing threshold. After each spike emission, the reset mechanism is invoked to update the membrane potential, as described in Eq.(1). Noteworthy, the spike $s[t]$ emitted by FS neuron at time t is amplified by learned spike intensity $d[t]$. After T timesteps, the neuron integrates the weighted spike train and forwards it to the next layer:

$$\hat{f}(x) = \sum_{t=0}^{T-1} d[t]s[t] \approx f(x), \tag{3}$$

where $\hat{f}(x)$ is the approximation value produced by FS neuron and $f(x)$ is the output of activation function in ANNs.

Problem Analysis

We conduct a systematic analysis of FS neurons to understand their limitations in Transformer-based A2S tasks.



(a) Binary initialization (b) Random initialization

Figure 1: Loss landscape of FS neurons in approximating the nonlinear activation function(i.e., ReLU). The 3D landscape of $\mathcal{L}_{Binary}$ and $\mathcal{L}_{Random}$ from two different initialization.

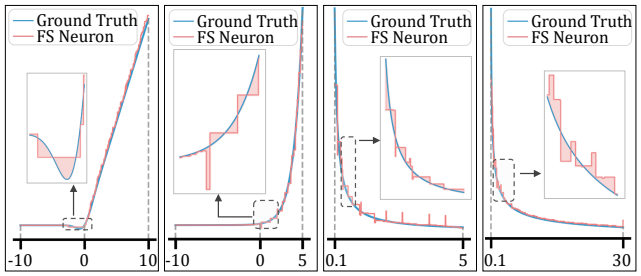
Approximation Error Analysis To qualitatively analyze how the FS neuron design impacts performance, following (Jiang et al. 2024), we consider errors from three sources: insufficient sampling, limited parameterization, and spiking quantization. This yields the following error bound:

Theorem 1 (FS Neuron Error Bound). *Let $f : [a, b] \to \mathbb{R}$ be a target activation function, and let $\hat{f}_T^{(M)}(x)$ denote the output of an FS neuron with T timesteps trained on M samples. Then the total approximation error satisfies:*

$$\varepsilon \leq \mathcal{O}\Bigg(\underbrace{\sqrt{\frac{T \log T \log M}{M}}}_{\text{Empirical Gap}} + \underbrace{\frac{\mathcal{L}_f |y|_{\max}}{T}}_{\text{Parametric Gap}} + \underbrace{\frac{\|d\|_1}{T}}_{\text{Quantization Gap}}\Bigg),$$

where $\varepsilon = \mathbb{E}[|f(x) - \hat{f}_T^{(M)}(x)|]$, $\mathcal{L}_f$ is the Lipschitz constant of f, $|y|_{\max}$ the maximum value of $f(x)$, and $\|d\|_1$ is the L_1-norm of spike intensities. Proof in Appendix A.

Excessive Dependence on Initialization From Theorem 1, the quantization gap $\frac{\|d\|_1}{T}$ depends on spike intensities $d[t]$, which are determined through optimization from initialization. Poor initialization may produce suboptimal $d[t]$ that increase $\|d\|_1$, thereby enlarging the approximation error. We conduct experiments to validate this: random initialization yields $\text{MSE}_{\text{random}} = 1.5 \times 10^{-3}$, while binary initialization ($V_{\text{th}}[t] = r[t] = d[t] = 2^{T-t}$) achieves $\text{MSE}_{\text{binary}} = 9.4 \times 10^{-5}$, representing a 93.29% performance degradation under random initialization. As shown in Fig. 1, binary initialization results in a smoother and more stable optimization landscape, while random initialization leads to multiple local minima. This further corroborates the sensitivity of FS neurons to initialization quality.



(a) GELU (b) e^x (c) $1/x$ (d) $1/\sqrt{x}$

Figure 2: Critical interval approximation failure of nonlinear operations in Transformer using FS neurons.

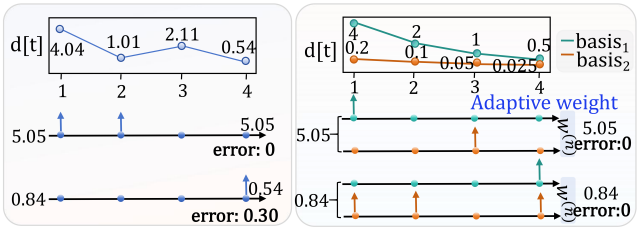
Global Suboptimality GSO stems from the parametric gap term $\frac{\mathcal{L}_f |y|_{\max}}{T}$ in Theorem 1, revealing FS neurons' limitations when approximating functions with non-uniform complexity. Transformer nonlinearity inputs concentrate near zero (Statistical analysis in Appendix C), where activation functions such as GELU and SiLU have high curvature and large local Lipschitz constants $\mathcal{L}_f^{\text{local}} \gg \mathcal{L}_f$. This creates an amplified local gap $\varepsilon_{\text{param}}^{\text{local}} = \mathcal{O}(\mathcal{L}_f^{\text{local}} |y|_{\max}^{local} / T)$, which dominates the approximation error and highlights the inadequacy of uniform time allocation. Experimental results support this analysis: as shown in Fig. 2, although the overall approximation error is small, the fitting degrades significantly in the near-zero region where outputs vary most rapidly. This region is crucial for Transformer performance, and the poor local approximation dominates overall behavior (FS-based Transformer conversion results in Appendix D).

Method

In this section, we first propose the MBE neuron, which employs exponential decay parameter update strategy and multi-basis encoding method to map activation values at multiple resolutions. Based on the MBE neuron, we further design an A2S conversion framework to overcome key challenges for Transformer architectures, achieving near-lossless conversion that is training-free and low-latency.

Multi-Basis Exponential Decay Neuron

To address the limitations of FS neurons, we propose a novel MBE neuron with an exponential decay parameter update strategy and multi-basis encoding scheme. Unlike FS neurons requiring learning multiple parameters $(d[t], r[t], V_{\text{th}}[t])$ per timestep, MBE neurons only learn decay rates and discrete timesteps, reducing parameter overhead and gradient instability (Details in Appendix E).



(a) FS neuron (b) MBE neuron with 2 basis

Figure 3: FS neuron and MBE neuron encoding methods.

The exponential decay mechanism enables progressive refinement from coarse to fine resolution, allowing adaptive approximation of functions with varying granularity and rapid focus on limited value ranges over shorter timesteps, as illustrated in Fig. 3. The parameter update is defined as:

$$Para(\tau_n, t) = \alpha \cdot exp\left(-\frac{t\Delta t}{\tau_n}\right), \tag{4}$$

where $Para(\tau_n, t)$ represents the parameter value at time t, α is a hyperparameter typically set to the target function's maximum value, Δt is the discrete timestep, and $\tau_n \in \{\tau_{d_n}, \tau_{r_n}, \tau_{V_{thn}}\}$ are the corresponding decay rates.

To further enhance the representational capacity for approximating diverse nonlinear functions, we introduce multi-basis encoding as the second core feature. As shown in Fig.4, each MBE neuron comprises n basis components contributing distinct functional components to the overall response. The dynamics of the n-th basis are defined as:

$$u_n[t+1] = u_n[t] - s_n[t] \cdot r_n[t], \tag{5}$$

$$s_n[t] = \mathcal{H}\big(u_n[t] - V_{thn}[t]\big), \tag{6}$$

$$o_n[t+1] = o_n[t] + s_n[t] \cdot d_n[t], \tag{7}$$

where $u_n[t]$ is the membrane potential of basis n at time t, $o_n[t]$ is the accumulated output, $d_n[t]$ denotes the spike intensity, $r_n[t]$ is the reset value, $V_{thn}[t]$ is the firing threshold, and $\mathcal{H}(\cdot)$ is the Heaviside step function. When $u_n[t] \geq V_{thn}[t]$, the neuron fires and emits a spike with intensity $d_n[t]$ while reducing the membrane potential by $r_n[t]$.

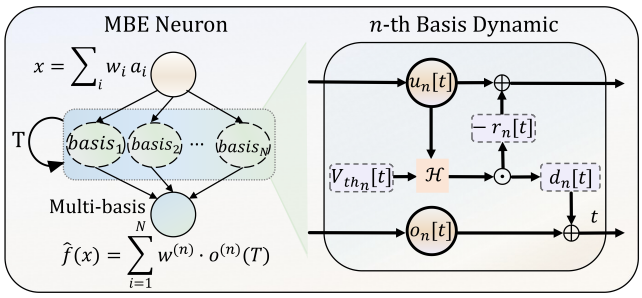


Figure 4: Multi-basis exponential decay neuron.

Finally, all basis outputs are weighted by a w to form $\hat{f}(x)$, passed to the next layer to approximate $f(x)$:

$$\hat{f}(x) = \sum_{n=1}^{N} w^{(n)} \cdot \mathrm{o}^{(n)}(T) \approx f(x). \tag{8}$$

Approximation Error Analysis Following the same analytical framework in Theorem 1, we derive error bounds for MBE neurons, accounting for the representational capacity:

Theorem 2 (MBE Neuron Error Bounds)**.** *For a target activation function $f : [a, b] \to \mathbb{R}$ and MBE neuron output $\hat{f}^{(M)}_{N,T}(x)$ with N basis components, T timesteps trained on M samples, the total approximation error satisfies:*

$$\varepsilon^* \leq O\left(\underbrace{\sqrt{\frac{N\log(N)\log M}{M}}}_{\textit{Empirical Gap}} + \underbrace{\frac{\mathcal{L}_f|y|_{max}}{NT}}_{\textit{Parametric Gap}} + \underbrace{\frac{\|w\|_1|\alpha|\tau_{max}}{T\Delta t}}_{\textit{Quantization Gap}}\right),$$

where $\varepsilon^* = \mathbb{E}[|f(x) - \hat{f}^{(M)}_{N,T}(x)|]$, $\mathcal{L}_f$ is the Lipschitz constant of f, $|y|_{max}$ is the maximum absolute value of the target function, $\|w\|_1$ is the L_1 norm of basis weights, $|\alpha|$ is the positive scaling parameter, $\tau_{max} = \max_n\{\tau_{d_n}, \tau_{r_n}, \tau_{V_{th_n}}\}$ is the maximum time constant, and Δt is the discrete timestep. The proof is provided in Appendix B.

The theoretical analysis shows MBE advantages and guides our conversion framework through three key insights:

Basis component design: The Parametric Gap $\mathcal{O}(1/NT)$ outperforms FS neurons' $\mathcal{O}(\mathcal{L}_f|y|_{\max}/T)$ by enabling multi-basis encoding to adaptively focus on high-curvature regions, solving the GSO problem without extra timesteps.

Initialization stability: The Quantization Gap $\frac{\|w\|_1|\alpha|\tau_{max}}{T\Delta t}$ mitigates the EDI problem through controlled scaling via $|\alpha|$ and time constant effects via $\tau_{max}/\Delta t$, enabling initialization-stability parameter updates.

Hyperparameter determination: The $1/NT$ scaling reveals a principled trade-off: increasing basis components N can effectively compensate for limited timesteps T, thereby achieving a balance between accuracy and low latency.

Conversion Framework

Based on MBE neurons, we propose a framework for Transformer-to-SNN conversion. As shown in Fig. 5, by decomposing non-spiking components (nonlinear activations, FP multiplication, Softmax, LayerNorm) into basic functions and designing corresponding MBE neurons for equivalent approximation, we enable spiking-based representation.

Approximation of Nonlinear Activation Functions To enable accurate and efficient spike-based approximation of nonlinear activation functions such as GELU and Tanh in Transformers, we employ MBE neurons with $N = 4$ basis components (in Eq.(8)). Since LayerNorm compresses activations into a narrow range, we constrain the GELU input domain to $(-120, 10)$ to enhance both approximation accuracy and training stability. Within this interval, we uniformly sample $M = 10{,}000$ points from the target function $f(x)$ and compute the spike-based output $\hat{f}(x)$ over T timesteps.

Approximation of Floating-Point Multiplication The FP multiplication operands x_1 and x_2 are transformed through approximate identity mapping by the MBE neuron MBE_{Id}, yielding the spike-train representations:

$$x \approx MBE_{Id}(x) = \sum_{t=0}^{T-1} d[t]s[t], \quad x \in \{x_1, x_2\}. \tag{9}$$

Then, the multiplication of x_1 and x_2 is expressed as:

$$x_1 \cdot x_2 \approx \sum_{i=0}^{T-1}\sum_{j=0}^{T-1} d[i]d'[j] \cdot s[i]s'[j]. \tag{10}$$

We represent the spike sequences of two MBE neurons as vectors $\mathbf{s} = \{s[t]\}_{t=0}^{T-1}$ and $\mathbf{s}' = \{s'[t]\}_{t=0}^{T-1}$ respectively, and denote their corresponding intensity sequences as vectors $\mathbf{d} = \{d[t]\}_{t=0}^{T-1}$ and $\mathbf{d}' = \{d'[t]\}_{t=0}^{T-1}$. Based on these temporal vectors, we construct the intensity matrix $\mathbf{D} \in \mathbb{R}^{T\times T}$ and the spike matrix $\mathbf{S} \in \{0, 1\}^{T\times T}$ as follows:

$$\mathbf{D} = \mathbf{d}^\top\mathbf{d}', \qquad \mathbf{S} = \mathbf{s}^\top\mathbf{s}'. \tag{11}$$

To achieve multiplication in the form of spikes, we employ the Hadamard product followed by summation. Consequently, Eq.(10) can be expressed as:

$$x_1 \cdot x_2 \approx \sum_{i=0}^{T-1}\sum_{j=0}^{T-1} \mathbf{D}_{ij} \cdot \mathbf{S}_{ij} = \sum_{i=0}^{T-1}\sum_{j=0}^{T-1} (\mathbf{D} \odot \mathbf{S})_{ij}, \tag{12}$$

where $\mathbf{D}_{ij} = d[i]d'[j]$ and $\mathbf{S}_{ij} = s[i]s'[j]$. The intensity matrix $\mathbf{D}$ can be precomputed and shared across the entire network. With the binary matrix $\mathbf{S}$, the entire FP multiplication approximation process maintains its spike-driven characteristics (Details in Appendix. F.1). By replacing FP multiplications in self-attention with our spike-driven operations, we enable fully spike-based attention matrix computation.

Approximation of Softmax The $\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$, where i ,j denote the indices of elements in the sequence. Since softmax depends on all inputs, it cannot be directly implemented by a single MBE neuron. As shown in Fig. 5(b), we decompose the softmax computation into three components: exponential (e^x), reciprocal ($1/x$), and FP multiplication, each approximated by MBE neurons. For the term e^{x_i}, we apply the change-of-base formula to decompose it into integer and decimal components as:

$$e^{x_i} = 2^{x_i\cdot\log_2 e} = 2^{\lfloor x_i\cdot\log_2 e\rfloor} \cdot 2^{x_i\cdot\log_2 e - \lfloor x_i\cdot\log_2 e\rfloor}, \tag{13}$$

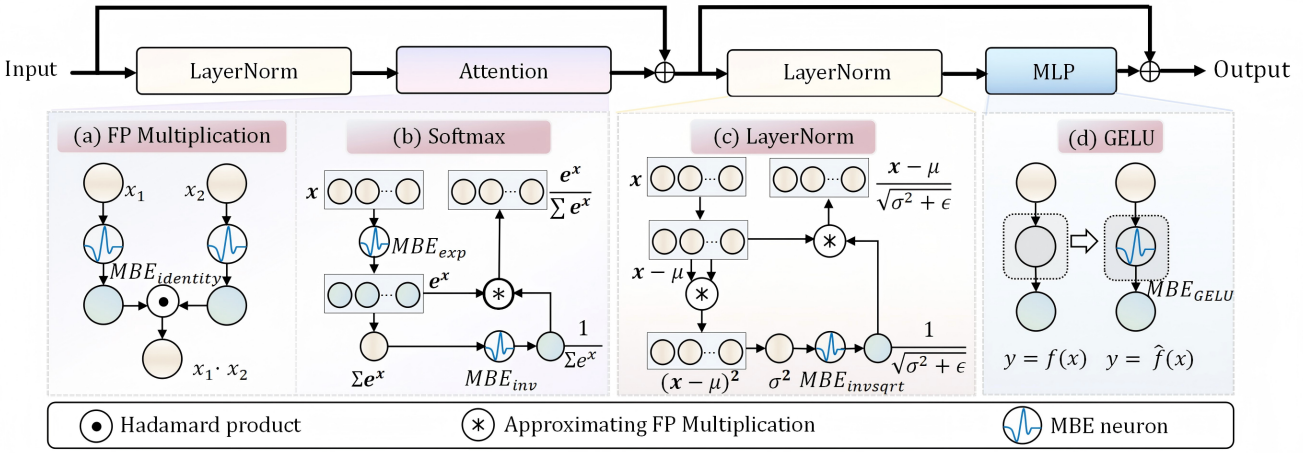


Figure 5: Overview of our framework, which depicts the approximations of FP multiplication, Softmax, LayerNorm, and GELU.

where $\lfloor \cdot \rfloor$ denotes the floor operation. The fractional part $2^{x_i \cdot \log_2 e - \lfloor x_i \cdot \log_2 e \rfloor}$ is approximated by MBE neurons, and the integer exponent $2^{\lfloor x_i \cdot \log_2 e \rfloor}$ is achieved by hardware-efficient addition (Li and Gu 2023). For the reciprocal term $1/\sum_j e^{x_j}$, we follow the IEEE 754 standard (Kahan 1996) to extract the exponent E and mantissa M through the corresponding bits of $\sum_j e^{x_j}$ (i.e., $\sum_j e^{x_j} = M \cdot 2^E$). The reciprocal $1/M$ is directly approximated using an MBE neuron, and the final result is obtained as $1/\sum_j e^{x_j} = 2^{-E}/M$.

Combining the approximations of e^{x_i} and $1/\sum_j e^{x_j}$, together with our FP multiplication approximation scheme, we compute the final softmax output in a spike-based manner (Detailed Softmax algorithm table in Appendix F.2).

Approximation of LayerNorm The $\mathrm{LN}(x_i) = \gamma \cdot \frac{x_i - \mu}{\sqrt{\sum (x_i - \mu)^2 / n + \epsilon}} + \beta$, where μ is the input mean, n is the number of elements, ϵ is a stability constant, and γ, β are learnable parameters. As shown in Fig. 5(c), we also decompose LayerNorm into three spike-unfriendly operations: squaring (x^2), reciprocal square root ($1/\sqrt{x}$), and FP multiplication, each approximated by MBE neurons.

For the squared term $\sum (x_i - \mu)^2$, we adopt the proposed FP multiplication approximation with the intensity matrix $\mathbf{D}$ pre-scaled by $1/n$ to directly yield $(x_i - \mu)^2/n$.

For the inverse square root $1/\sqrt{\sum (x_i - \mu)^2/n + \epsilon}$, we decompose the variance term into exponent E and mantissa M. We adjust E and M based on E's parity to ensure $-E/2$ is an integer. The term $1/\sqrt{M}$ is approximated directly using MBE neuron, and the inverse square root is obtained by multiplying with $2^{-E/2}$, similar to the Softmax approximation. Finally, we multiply $(x_i - \mu)$ with the approximated $1/\sqrt{\sum (x_i - \mu)^2/n + \epsilon}$ using our FP multiplication approximation to obtain the LayerNorm output.

Based on the aforementioned conversion of all non-spike-friendly components into equivalent spiking forms, we replace the corresponding modules to construct the Transformer-based SNN without modifying the original network parameters. Detailed pseudocode is in Appendix F.3.

Experiments

Experimental Setup

We conduct experiments on a GPU (RTX 4090) environment using the PyTorch framework, evaluating both pre-trained Vision Transformer (ViT) models, including ViT-Base-Patch16 (ViT-B/16) (Dosovitskiy et al. 2020) and ViT-Medium-Patch16-Reg4-Gap-256 (ViT-M/16) (Darcet et al. 2023), as well as CNN models, including VGG16 (Simonyan and Zisserman 2014) and ResNet34 (He et al. 2016) on the ImageNet dataset (Deng et al. 2009). To validate the generalizability of our method across different language domains and tasks, we adapt RoBERTa (Liu et al. 2019; Zhang et al. 2025c) for natural language understanding (NLU) tasks and employ GPT-2 (Radford et al. 2019) for natural language generation (NLG) task evaluation (Experimental and implementation details in Appendix G.1).

Comparative Study

Comparison on CV We evaluate the performance of our conversion framework by applying it to convert both ViT and CNN architectures. As shown in Tab. 1, ViT-B/16 and ViT-M/16 achieve conversion losses of 0.44% and 0.64% respectively—significantly lower than most existing methods. The framework attains 85.31% accuracy for ViT and 75.57% for CNN, outperforming other SNNs. While ECMT achieves competitive accuracy at shorter timesteps, it retains FP multiplication in its expectation compensation module. In contrast, our A2S framework maintains crucial spike-based property, thereby avoiding the high energy consumption typically associated FP multiplication during inference. Furthermore, the training-free nature of our approach enables conversion with minimal computational overhead, eliminating the need for extensive retraining for source ANNs. Our method successfully accomplishes A2S conversion without requiring additional training of the original network, preserves the essential spiking properties, delivers superior performance at short timesteps, and achieves optimal results across all evaluation dimensions.

Category	Method	Arch.	SD	TF	Param[M]	Timesteps	ANN	Acc.[%](Δ)
DT	DSR(Meng et al. 2022)	ResNet-18	✔	-	12	50	-	67.74
	TET(Deng et al. 2022)	SEW-ResNet-34	✘	-	22	4	-	68.00
	AT-SNN(Yao et al. 2023b)	ResNet-104	✘	-	45	4	-	77.08
	Spikingformer(Zhou et al. 2023)	-4-384-400E	✔	-	66	4	-	75.85
	SDTv1(Yao et al. 2023a)	-8-768	✔	-	66	4	-	77.07
	Spikeformer(Li, Lei, and Yang 2022)	-7L/3×2×4	✘	-	38	4	-	78.31
	QKFormer(Zhou et al. 2024)	HST-10-768	✔	-	65	4	-	84.22
	SDTv3(Yao et al. 2025)	E-SpikeFormer	✔	-	173	8	-	85.10
A2S	QCFS (Bu et al. 2023)	VGG-16	✔	✘	138	64	74.92	72.85(-2.07)
	QFFS(Li, Ma, and Furber 2022)	VGG-16	✔	✘	138	8	73.08	73.10(+0.02)
	SNM(Wang et al. 2022a)	ResNet-18	✔	✔	12	64	73.18	71.50(-1.68)
	QCFS(Bu et al. 2023)	ResNet-34	✔	✘	22	64	74.23	72.35(-1.88)
	ECL(Liu et al. 2025)	ResNet-34	✔	✘	22	16	74.36	72.37(-1.99)
	AdaFire(Wang et al. 2025d)	ResNet-34	✔	✔	22	8	75.66	72.96(-2.70)
	Ours	VGG16	✔	✔	138	10	73.37	72.61 (-0.76)
		ResNet-34	✔	✔	22	10	76.31	**75.57(-0.74)**
	ECMT(Huang et al. 2024)	ViT-L/16	✘	✔	307	12	84.88	84.71(-0.17)
	STA(Jiang et al. 2024)	ViT-B/32	✘	✔	88	256	83.60	82.79(-0.81)
	SpikeZIP-TF(You et al. 2024)	SViT-L-32Level	✔	✘	304	64	85.41	83.82(-1.59)
	SpikedAttention(Hwang et al. 2024)	Swin-T(BN)	✘	✔	28	48	79.30	77.20(-2.10)
	Ours	ViT-B/16	✔	✔	86	16	83.44	83.00(-0.44)
		ViT-M/16	✔	✔	64	16	85.95	**85.31(-0.64)**

Table 1: ImageNet performance comparison. Here, "SD" and "TF" represent Spike-Driven and Training-Free.

Comparison on NLU To evaluate our method on NLU tasks, we conduct experiments on four benchmark datasets including SST-2, SST-5, MR, and Subj, following standard experimental settings from studies (You et al. 2024; Zhu et al. 2023). As shown in Table 2, our method achieves competitive performance across all datasets, outperforming existing methods on both RoBERTa-Base (125M) and RoBERTa-Large (355M). On SST-2, it attains 95.98% accuracy with $T = 16$, representing only 0.24% degradation from the original ANN (96.22%), while outperforming SpikeZIP-TF (You et al. 2024) by 2.19% and reducing timesteps by 87.5%. Similarly, on MR with RoBERTa-Base, our method achieves 89.00% accuracy at $T = 16$, exceeding SpikeZIP-TF by 2.87% with 75% fewer timesteps.

Category	Model	Param	SST-2	SST-5	MR	Subj	T
ANN	Roberta	125	94.49	55.46	89.39	96.45	1
		355	96.22	59.37	91.36	97.50	1
DT	Spikeformer	110	81.55	42.02	79.38	91.80	4
	SpikeBERT	109	85.39	46.11	80.69	93.00	4
	SpikeGPT	45	80.39	37.69	69.23	88.45	50
A2S	SpikeZIP-TF	125	92.81	52.71	86.13	95.55	64
		355	93.79	56.51	89.28	96.70	128
	Ours	125	93.46	55.11	89.00	96.30	16
		355	**95.98**	**58.31**	**90.96**	**97.45**	16

Table 2: NLU Performance Comparison

Comparison on NLG To demonstrate the applicability of our method to NLG tasks, we evaluate its performance on WikiText-2 and WikiText-103. As shown in Tab. 3, our method achieves 22.69 perplexity on WikiText-2 with only T=16 timesteps, incurring merely 0.35% conversion loss compared to GPT-2. More significantly, on WikiText-103, it attains 23.41 perplexity—a substantial 41.1% improvement over directly-trained SpikeGPT (39.75) while using dramatically fewer timesteps (T=16 vs. T=1024), closely approaching the original GPT-2 performance and demonstrating superior efficiency for long-context generation.

Category	Model	Param	Wiki-2 (↓)	Wiki-103 (↓)	T
ANN	GPT-2	346	22.34	22.65	1
DT	SpikeGPT	216	**18.01**	39.75	1024
A2S	Ours	346	22.69 (+0.35)	**23.41** (+0.76)	16

Table 3: NLG Performance Comparison. Both Wiki-2 and Wiki-103 use perplexity, ↓ indicates that lower is better.

Ablation Study

To compare MBE neurons and FS neurons, we evaluate their performance on key Transformer components. For GELU and FP multiplication, direct replacement is employed. As shown in Fig. 6(a), MBE neurons surpass FS neurons by 13.20% in GELU conversion and achieve improvements in

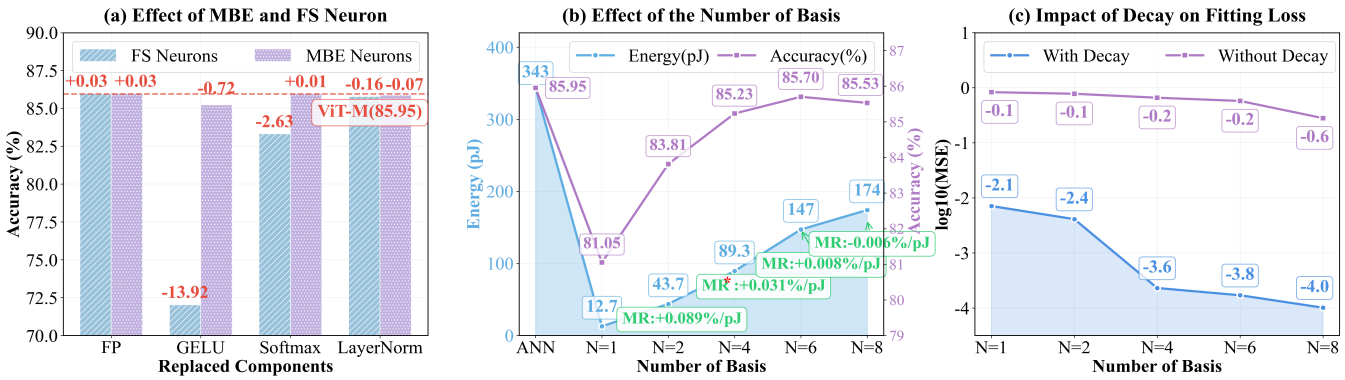


Figure 6: (a) Effect of MBE and FS neuron on components in Transformer, FP in (a) means floating multiplication. (b) Impact of MBE neurons with varying basis counts on energy consumption and accuracy when replacing GELU. MR represents marginal rate of change between accuracy and energy consumption. (c) Effect of the basis and decay. Loss comparison between models with (blue) and without (red) decay mechanism across basis numbers N=1,2,4,6,8. Y-axis shows log MSE loss (lower is better).

Softmax and LayerNorm, demonstrating superior approximation ability. In Fig. 6(b), increasing N leads to more accurate function approximations by MBE neurons. MBE neurons achieve high performance (e.g., less than 1% conversion loss when N $\geq$ 4) while offering significantly higher energy efficiency than ANNs. N=4 achieves the optimal accuracy-MR trade-off, representing the knee point where marginal gains plateau. The synergistic effect between exponential decay and multi-basis encoding is demonstrated in Fig. 6(c). Incorporating decay consistently reduces MSE by 1–2 orders of magnitude across all basis numbers, with significant performance improvements. This reveals that exponential decay creates a multiplicative synergy with multi-basis encoding, enabling exceptional approximation precision unattainable through basis expansion alone.

Dataset	Model	ANN	T			
			8	10	12	16
ImageNet	ViT-B/16	83.44	0.12	79.96	82.79	83.00
	ViT-M/16	85.95	1.17	21.99	84.43	85.31
MR	Roberta-B	89.39	50.17	71.30	88.55	89.00
	Roberta-L	91.36	49.66	68.62	90.52	90.96
Wiki-103 (↓)	GPT-2	22.65	41072	11992	33.56	23.41

Table 4: Performance across various timesteps and models. Wiki-103 uses perplexity, ↓ indicates that lower is better.

We evaluate ImageNet, MR, and Wiki-103 to analyze timestep requirements for A2S conversion across vision and language tasks. As shown in Tab. 4, our method achieves about 1% conversion loss for ViT, RoBERTa, and GPT-2 at T=16, with near-optimal performance at T=12, demonstrating efficient progressive encoding. Notably, the accuracy of ViT-M/16 drops to 21.99% at T=10, primarily due to its wider identity mapping range [0,62] causing amplified approximation errors under limited timesteps.

Energy Estimation

Unlike ANNs where energy depends on floating-point operations (FLOPs), the energy cost of SNNs is dominated by synaptic operations (SOPs). Following (Rathi and Roy 2020), the energy ratio between SNNs and ANNs is:

$$\frac{E_{\text{SNN}}}{E_{\text{ANN}}} = \frac{\text{SOPs} \cdot E_{\text{AC}}}{\text{FLOPs} \cdot E_{\text{MAC}}}, \tag{14}$$

where E_{MAC} = 4.6pJ, E_{AC} = 0.9pJ. For ViT-M/16, we measure the firing rates η of nonlinear operations across all layers. Taking GELU as an example, the ANN implementation requires 70 FLOPs (Jiang et al. 2024), while the MBE neuron achieves η=38.22% under N=4 and T=16. The synaptic operations achieve energy consumption of 13.7% according to $T * N * \eta * E_{AC}$. Other operations yield greater energy savings owing to lower MBE firing rates. Complete results and detailed firing rates are provided in Appendix G.4.

Conclusion

This paper proposes an efficient Spiking Transformers conversion method that requires no additional training on the source ANNs. We first theoretically and experimentally identify two key challenges in using FS neurons to approximate nonlinear operations: excessive dependence on initialization (EDI) and global suboptimality (GSO) problems. To address these issues, the MBE neuron employs an exponential decay strategy and a multi-basis encoding method to more accurately approximate the nonlinear operations in Transformer architecture. Based on the MBE neuron, a general ANN-to-SNN conversion framework is developed, which supports spiking nonlinear activation functions, spiking FP multiplications, spiking Softmax, and spiking LayerNorm. Extensive experiments on various models (CNN, ViT, RoBERTa, GPT-2) across tasks (CV, NLU, NLG) confirm that the proposed method achieves near-lossless conversion. Therefore, our method provides a promising approach for the efficient and scalable deployment of Transformer-based SNNs in real-world applications.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (62220106008 and 62576080), in part by the State Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Grant No. SKLBI-K2025010.

References

Bu, T.; Ding, J.; Yu, Z.; and Huang, T. 2022. 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. 2023. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*.

Cai, L.; Wang, H.; Ji, J.; ZhouMen, Y.; Ma, Y.; Sun, X.; Cao, L.; and Ji, R. 2025. Zooming In on Fakes: A Novel Dataset for Localized AI-Generated Image Detection with Forgery Amplification Approach. *arXiv preprint arXiv:2504.11922*.

Darcet, T.; Oquab, M.; Mairal, J.; and Bojanowski, P. 2023. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal efficient training of spiking neural network via gradient reweighting. *arXiv preprint arXiv:2202.11946*.

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.

Ding, J.; Yu, Z.; Tian, Y.; and Huang, T. 2021. Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks. *arXiv preprint arXiv:2105.11654*.

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*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

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.

Hwang, S.; Lee, S.; Park, D.; Lee, D.; and Kung, J. 2024. Spikedattention: Training-free and fully spike-driven transformer-to-snn conversion with winner-oriented spike shift for softmax operation. *Advances in Neural Information Processing Systems*, 37: 67422–67445.

Jiang, H.; Anumasa, S.; De Masi, G.; Xiong, H.; and Gu, B. 2023. A unified optimization framework of ANN-SNN conversion: towards optimal mapping from activation values to firing rates. In *International Conference on Machine Learning*, 14945–14974. PMLR.

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*.

Kahan, W. 1996. IEEE standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, 754(94720-1776): 11.

Li, C.; Ma, L.; and Furber, S. 2022. Quantization framework for fast spiking neural networks. *Frontiers in Neuroscience*, 16: 918793.

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

Li, Z.; and Gu, Q. 2023. I-vit: Integer-only quantization for efficient vision transformer inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17065–17075.

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, C.; Shen, J.; Ran, X.; Xu, M.; Xu, Q.; Xu, Y.; and Pan, G. 2025. Efficient ANN-SNN Conversion with Error Compensation Learning. *arXiv preprint arXiv:2506.01968*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Mao, R.; Tang, L.; Xia, Z.; Shen, A.; Wei, H.; Zhang, Z.; Lu, Y.; Long, Y.; Guo, J.; Zhou, L.; et al. 2025. FSNAP: An Ultra-Energy-Efficient Reconfigurable Few-Spikes-Neuron-Based SNN Processor Supporting Unified On-Chip Learning and Adaptive Time-Window Tuning. *IEEE Journal of Solid-State Circuits*.

Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2022. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12444–12453.

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.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rathi, N.; and Roy, K. 2020. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*.

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.

Shan, Y.; Ren, Z.; Wu, H.; Wei, W.; Zhu, R.-J.; Wang, S.; Zhang, D.; Xiao, Y.; Zhang, J.; Shi, K.; et al. 2025. Sdtrack: A baseline for event-based tracking via spiking neural networks. *arXiv preprint arXiv:2503.08703*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Stöckl, C.; and Maass, W. 2021. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3): 230–238.

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.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, S.; Zhang, D.; Belatreche, A.; Xiao, Y.; Qing, H.; Wei, W.; Zhang, M.; and Yang, Y. 2025a. Ternary spike-based neuromorphic signal processing system. *Neural Networks*, 187: 107333.

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, S.; Zheng, H.; Chen, Y.; Belatreche, A.; Wang, G.; Jin, Y.; Wu, J.; Zhang, M.; Yang, Y.; and Li, H. 2025c. SNN-FT: Temporal-Coded Spiking Neural Networks for Fourier Transform. *IEEE Transactions on Neural Networks and Learning Systems*.

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

Wang, Z.; Fang, Y.; Cao, J.; Ren, H.; and Xu, R. 2025d. Adaptive calibration: A unified conversion framework of spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1583–1591.

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.

Wang, Z.; Lian, S.; Zhang, Y.; Cui, X.; Yan, R.; and Tang, H. 2022b. Towards lossless ANN-SNN conversion under ultra-low latency with dual-phase optimization. *arXiv preprint arXiv:2205.07473*.

Wei, W.; Liang, Y.; Belatreche, A.; Xiao, Y.; Cao, H.; Ren, Z.; Wang, G.; Zhang, M.; and Yang, Y. 2024. Q-snns: Quantized spiking neural networks. In *Proceedings of the 32nd ACM international conference on multimedia*, 8441–8450.

Wei, W.; Zhang, M.; Zhou, Z.; Belatreche, A.; Shan, Y.; Liang, Y.; Cao, H.; Zhang, J.; and Yang, Y. 2025. Qp-snn: Quantized and pruned spiking neural networks. *arXiv preprint arXiv:2502.05905*.

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.

Yao, M.; Hu, J.; Zhou, Z.; Yuan, L.; Tian, Y.; Xu, B.; and Li, G. 2023a. Spike-driven transformer. *Advances in neural information processing systems*, 36: 64043–64058.

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*.

Yao, M.; Zhao, G.; Zhang, H.; Hu, Y.; Deng, L.; Tian, Y.; Xu, B.; and Li, G. 2023b. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8): 9393–9410.

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. *arXiv preprint arXiv:2406.03470*.

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.; Luo, X.; Wu, J.; Belatreche, A.; Cai, S.; Yang, Y.; and Li, H. 2025b. Toward Building Human-Like Sequential Memory Using Brain-Inspired Spiking Neural Models. *IEEE transactions on neural networks and learning systems*.

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

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

Zhou, C.; Zhang, H.; Zhou, Z.; Yu, L.; Huang, L.; Fan, X.; Yuan, L.; Ma, Z.; Zhou, H.; and Tian, Y. 2024. Qkformer: Hierarchical spiking transformer using qk attention. *arXiv preprint arXiv:2403.16552*.

Zhu, R.-J.; Zhao, Q.; Li, G.; and Eshraghian, J. K. 2023. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*.