Quantization Meets Spikes: Lossless Conversion in the First Timestep via Polarity Multi-Spike Mapping

Hangming Zhang[1*], **Zheng Li**[2*], **Qiang Yu**[1, 3†]
[1]College of Intelligence and Computing, Tianjin University,Tianjin, China
[2]School of Future Technology, Tianjin University, Tianjin, China
[3]College of Computer and Information Engineering, Tianjin Normal University, Tianjin, China
{zhm0755, li229049, yuqiang}@tju.edu.cn

Abstract

Spiking neural networks (SNNs) offer advantages in computational efficiency via event-driven computing, compared to traditional artificial neural networks (ANNs). While direct training methods tackle the challenge of non-differentiable activation mechanisms in SNNs, they often suffer from high computational and energy costs during training. As a result, ANN-to-SNN conversion approach still remains a valuable and practical alternative. These conversion-based methods aim to leverage the discrete output produced by the quantization layer to obtain SNNs with low latency. Although the theoretical minimum latency is one timestep, existing conversion methods have struggled to realize such ultra-low latency without accuracy loss. Moreover, current quantization approaches often discard negative-value information following batch normalization and are highly sensitive to the hyperparameter configuration, leading to degraded performance. In this work, we, for the first time, analyze the information loss introduced by quantization layers through the lens of information entropy. Building on our analysis, we introduce Polarity Multi-Spike Mapping (PMSM) and a hyperparameter adjustment strategy tailored for the quantization layer. Our method achieves nearly lossless ANN-to-SNN conversion at the extremity, i.e., the first timestep, while also leveraging the temporal dynamics of SNNs across multiple timesteps to maintain stable performance on complex tasks. Experimental results show that our PMSM achieves state-of-the-art accuracies of 98.5% on CIFAR-10, 89.3% on CIFAR-100 and 81.6% on ImageNet with only one timestep on ViT-S architecture, establishing a new benchmark for efficient conversion. In addition, our method reduces energy consumption by over 5× under VGG-16 on CIFAR-10 and CIFAR-100, compared to the baseline method.

Introduction

Spiking Neural Networks (SNNs) are a type of biologically inspired neural networks that encode and transmit information through discrete spike events. Compared with traditional artificial neural networks (ANNs), SNNs exhibit superior energy efficiency and neuromorphic hardware compatibility (Jang et al. 2019; Zhang et al. 2020). There are two mainstream paradigms for building deep SNNs: surrogate-gradient based direct training (Bohte, Kok, and La Poutré 2000; Neftci, Mostafa, and Zenke 2019) and ANN-to-SNN based conversion (Diehl et al. 2015; Cao, Chen, and Khosla 2015). Direct training methods approximate the non-differentiable spiking function by temporally unrolling the computational graph and employing differentiable surrogate gradients. However, previous studies (Neftci, Mostafa, and Zenke 2019; Ding et al. 2024) indicate that more timesteps are required to ensure the reliability of surrogate gradients, resulting in significant increases in computational and energy costs during training. In contrast, ANN-to-SNN conversion methods train ANNs using conventional backpropagation without temporally unrolling, then directly convert the pretrained weights to SNNs. These conversion-based methods enable efficient construction of deep SNNs while preserving the original model architecture, achieving performance comparable to ANNs on multiple large-scale datasets (Liu et al. 2022). In addition, due to the training efficiency and compatibility with existing architectures, ANN-to-SNN conversion methods are well-suited for deployment on resource-constrained neuromorphic hardware (Zhang et al. 2025a,b).

Although ANN-to-SNN conversion methods have demonstrated significant advantages in constructing efficient SNNs, several challenges remain. First, due to the fundamental difference between the continuous activations in ANNs and the discrete spiking activity in SNNs, direct conversion often leads to accuracy degradation (Jiang et al. 2023), with errors accumulating layer by layer as the network depth increases (Li et al. 2022). Second, to mitigate the adverse effects of this difference, existing methods typically introduce a longer inference time (Li et al. 2021) to approximate the continuous outputs, resulting in substantially increased inference latency and energy consumption (Deng et al. 2020).

To bridge the fundamental mismatch between continuous activations and discrete spikes, several methods (Deng and Gu 2021; Li et al. 2021, 2022) calibrate the firing rates to better align outputs of SNNs with the pretrained ANNs. However, the firing rate calibration typically requires fine-grained, layer-wise parameter tuning and lacks adaptability across network architectures, resulting in limited generalization. Other approaches apply quantized activation functions to ANNs (You et al. 2024; Bu et al. 2022b; Yang et al. 2025; Li, Ma, and Furber 2022), discretizing continuous activa-

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

tions into bit-width representations to enable the converted SNNs to accurately approximate outputs of ANNs within fewer timesteps, thereby reducing the inference latency. Furthermore, while multi-spike (Song et al. 2021; Lan et al. 2023; Hao et al. 2024) and polarity spike (Yu et al. 2022b,a; Ma and Yu 2020) conversion schemes have been separately explored to improve conversion precision and inference latency, effectively combining both multi-spike and polarity in a unified mapping framework presents significant challenges. In particular, accurately reconstructing continuous-valued activations within a limited number of timesteps requires precise calculation of spike number and polarity per timestep, as both jointly determine the magnitude and sign of the represented value. Moreover, negative spike handling must remain compatible with existing SNNs dynamics and maintain low latency, further complicating its integration into conversion method. As a result, existing quantized activation functions tend to clip or ignore negative information, leading to loss of important information and degraded performance in the converted SNNs.

To address the aforementioned issues, we propose Polarity Multi-Spike Mapping (PMSM), a novel method that, to the best of our knowledge, is the first to achieve nearly lossless ANN-to-SNN conversion at the first timestep while maintaining stable performance across multiple timesteps. We conduct systematic experiments on representative architectures to evaluate the effectiveness of PMSM. The results show that our method achieves state-of-the-art with ultra-low latency and demonstrates strong generalization across diverse architectures. The main contributions of this work are as follows:

- **Nearly lossless ANN-to-SNN conversion at the first timestep:** To enable spiking neurons to convey rich information within a single timestep, we propose an Augmented Integrate-and-Fire (AIF) neuron that supports a polarity multi-spike firing mechanism. This design enables precise reconstruction of ANNs activations with ultra-low latency, achieving high-performance conversion at the first timestep.
- **Polarity quantized activation function:** To address information loss caused by conventional quantized activation functions, which truncate negative values in the outputs of batch normalization layers, we design a Polarity Quantized Activation (PQA) function. From an information-theoretic perspective, we show that PQA preserves the entropy of the original signal, ensuring lossless information propagation through quantized activations.
- **Generalization across diverse architectures:** We evaluate PMSM on three representative architectures: a deep plain CNN (VGG-16), a residual network (ResNet-20), and an attention-based model (Transformer). To the best of our knowledge, PMSM achieves state-of-the-art performance on CIFAR-10, CIFAR-100, and ImageNet, demonstrating strong generalization and scalability across diverse network paradigms. In addition, PMSM achieves over 5× reduction in energy consumption compared to the baseline method.

Related Works

ANN-to-SNN Conversion: ANN-to-SNN conversion is one of the main approaches to achieving nearly lossless and low-latency SNNs. Recent efforts have focused on minimizing conversion error through two main strategies. Some researchers focus on parameter calibration. Early work introduced SNN-adapted modules and analyzed conversion errors theoretically (Rueckauer et al. 2017), laying the groundwork for low-latency conversion. Subsequent studies showed that proper membrane potential initialization enables lossless conversion (Bu et al. 2022a), and further alleviated unevenness errors by optimizing both the initial and residual membrane potentials (Hao et al. 2023a,b). Other efforts combined threshold balancing and soft reset to achieve nearly lossless conversion (Deng and Gu 2021), and applied second-order analysis and minimized activation mismatch via layer-wise parameter calibration (Li et al. 2021, 2022). Others focus on activation function design. Several continuous-valued ANNs activation functions have been proposed to improve conversion effect (Ding et al. 2021; Wang et al. 2023a; Han et al. 2023; Jiang et al. 2023). Furthermore, the quantized-clip-shift activation function (Bu et al. 2022b) produces discrete outputs that allow the firing rates of SNNs to approximate the outputs of ANNs within few timesteps, laying the foundation for converting large-scale models such as Transformer architectures (Wang et al. 2023b; You et al. 2024; Hwang et al. 2024).

Spiking Neural Models in Conversion Process: Integrate-and-Fire (IF) neurons, which directly accumulate input currents without incorporating leakage terms, can better approximate ANNs activations and are widely adopted in converted SNNs (Han, Srinivasan, and Roy 2020). However, the binary spike mechanism of IF neurons limits representational capacity, and a recent study (Liu et al. 2025) introduces a compensation mechanism to align the firing rate more closely with the original ANNs, yielding promising results. To enhance representational capacity, prior works (Wang et al. 2022, 2024a; Guo et al. 2024; Zhou et al. 2024; Yu et al. 2022b,a; Ma and Yu 2020; Song et al. 2021; Wang et al. 2024b) have introduced spike polarity, enabling neurons to emit negative spikes when the membrane potential falls below a negative threshold. This polarity mechanism allows the encoding of negative information. In this paper, we refer to neurons capable of representing both positive and negative information as polarity neurons. Meanwhile, inspired by the burst firing behavior observed in biological neurons (Kepecs, Wang, and Lisman 2002), several studies (Song et al. 2021; Lan et al. 2023; Hao et al. 2024) have proposed neuron models capable of emitting multiple spikes within a single timestep, and one of those studies (Gao et al. 2023) further incorporates a compensation mechanism on top of this design, thereby enhancing instantaneous representational capacity. We refer to these neurons as multi-spike neurons. Furthermore, to combine the advantages of polarity and multi-spike neurons, Yu et al. (Yu et al. 2022a) proposed TerMapping and AugMapping, which were designed to mitigate the loss of negative information during ANN-to-SNN conversion.

Building upon these works, we propose a Polarity Quan-

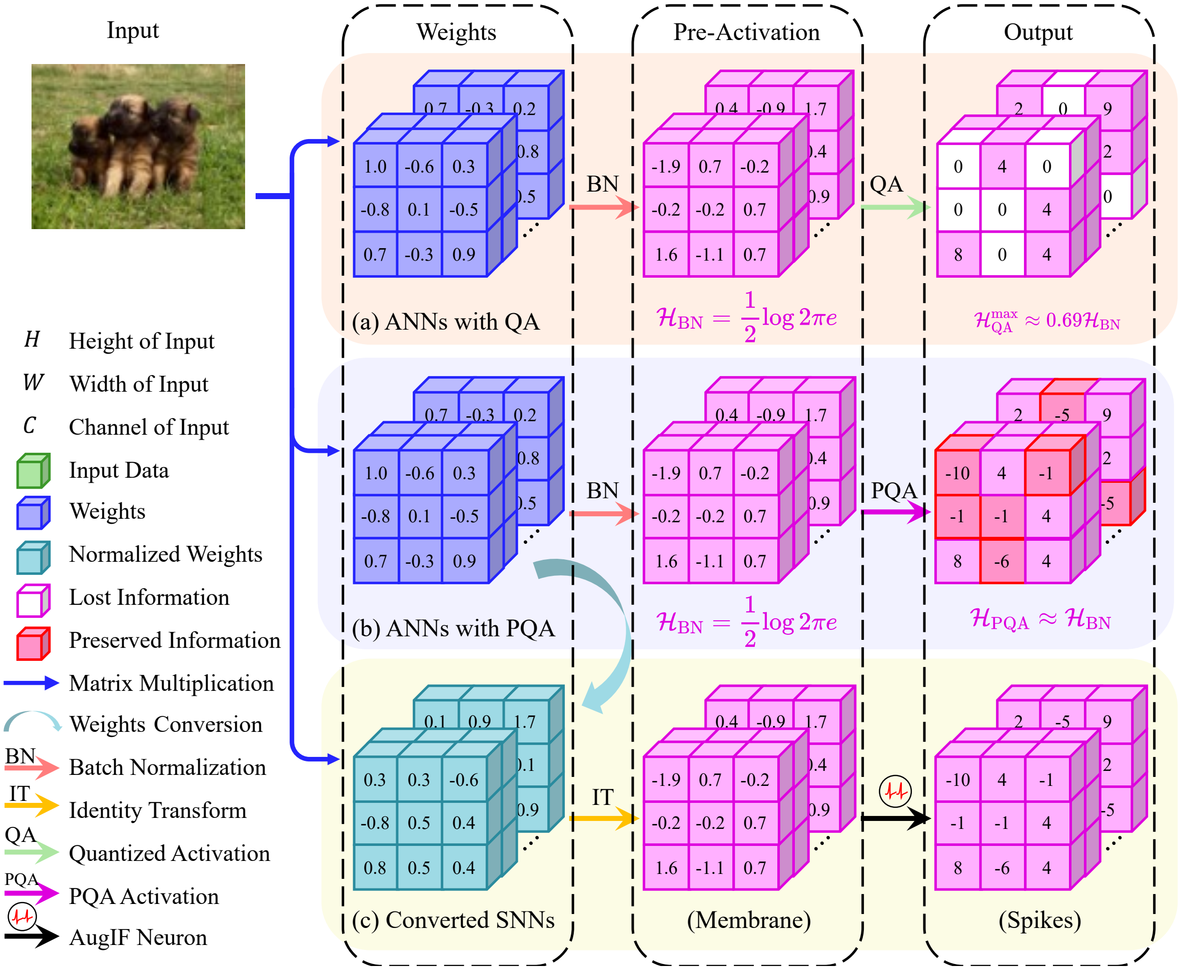

Figure 1: Illustration of the information transmission and conversion process. Under identical weights and inputs, panel (a), the first row, uses a quantized activation (QA) function with a maximum entropy of approximately $0.69\mathcal{H}_{\text{BN}}$; white cubes indicate locations of information loss. Panel (b) applies a Polarity Quantized Activation (PQA) function with entropy close to $\mathcal{H}_{\text{BN}}$; red cubes denote lossless transmission. During conversion, the parameters of BN layers are first folded into the weights of preceding layers, and PQA parameters are subsequently transferred to AIF neurons. Panel (c) shows the resulting computation graph of the converted SNNs.

tized Activation function to enhance the ability of ANNs to encode both positive and negative information. In addition, we design a polarity multi-spike neuron model, which, to the best of our knowledge, is the first to enable accurate and nearly lossless ANN-to-SNN conversion at the first timestep, thereby providing a promising direction for the development of ultra-low latency SNNs architectures.

Polarity Multi-Spike Mapping

To achieve nearly lossless ANN-to-SNN conversion under low-latency constraints, we propose the Polarity Multi-Spike Mapping (PMSM) method, which consists of three core components. First, we introduce a Polarity Quantized Activation (PQA) function that preserves the information of Batch Normalization (BN) layers during discretization. Second, we design an Augmented Integrate-and-Fire (AIF) neuron model capable of emitting multiple polarity spikes within a single timestep, enhancing the instantaneous representational capacity of SNNs under extreme latency constraints. Finally, we conduct a theoretical analysis of the conversion error sources under both single- and multi-timestep conditions, demonstrating the effectiveness of PQA across different temporal configurations. The following sections detail these three components in turn.

Lossless Quantized Activation Function Based on Information Entropy

In typical ANN-to-SNN conversion methods based on quantized activation functions, ReLU is usually replaced with a discretized activation function to approximate the firing rate. However, since the outputs of BN layers contain negative values, traditional quantization functions often employ a truncation strategy that discards negative values, thereby preventing the transmission of negative information, as illustrated in Fig. 1(a). Such information loss weakens the network's ability to capture negative signals and ultimately limits the representation fidelity and classification accuracy of the resulting SNNs. To systematically assess the extent of information loss between BN outputs and their quantized forms, we introduce information entropy $\mathcal{H}$ as a quantitative measure. Specifically, assuming that the activation values follow a discrete distribution $p(x)$, the entropy is defined as:

$$\mathcal{H}(x) = -\sum_{i} p(x_i) \log p(x_i) \tag{1}$$

Information entropy reflects both the uncertainty of a variable and its capacity to represent information. Therefore, we aim for the quantized activations to preserve the entropy of the original activation distribution to the greatest extent possible. Considering that BN standardizes inputs to a zero-mean, unit-variance Gaussian distribution, the corresponding theoretical entropy is $\mathcal{H}_{\text{BN}} = \frac{1}{2}\log(2\pi e)$. The output of BN is subsequently processed by conventional clipping-based quantization functions. These functions discard negative information, and their output entropy is upper bounded by that of ReLU. However, even under this favorable condition, the theoretical entropy of the ReLU output is limited to $\mathcal{H}_{\text{ReLU}} \approx 0.69\mathcal{H}_{\text{BN}}$, suggesting a non-negligible loss of information inherent to traditional asymmetric quantization strategies. To address this issue, we propose the Polarity Quantized Activation (PQA) function, which retains positive activation values and introduces a quantized representation for negative activation values, thereby improving information completeness while preserving the advantages of discretization. The definition of PQA is as follows:

$$y^l = \vartheta^l clip(\frac{1}{L}\lfloor \frac{x^l L}{\vartheta^l} \rceil, \alpha, \beta), \quad x^l = W^l y^{l-1} \tag{2}$$

Here y^{l-1} denotes the input of the l-th layer as well as the output of the $l-1$-th layer. W^l is the weight matrix of the l-th layer, and ϑ^l is a learnable quantization threshold. L specifies the number of discrete quantization levels, while $\lfloor \cdot \rceil$ represents the rounding operation. The parameters α and β define the lower and upper bounds of the quantization range, satisfying $-1 < \alpha \leq 0$ and $0 < \beta \leq 1$. We theoretically analyze how the output entropy of the PQA function varies with different configurations of ϑ^l and L. The analysis shows that when $\vartheta \ll L$, most outputs concentrate toward the clipping boundaries. Thus, the distribution of outputs becomes effectively binarized, and the entropy approaches $\mathcal{H}_{\text{PQA}} \approx 0.49\mathcal{H}_{\text{BN}}$. In contrast, when $\vartheta \gg L$, most input values are collapsed to zero, leading to an almost complete loss of information entropy. As a result, $\mathcal{H}_{\text{PQA}} \approx 0$. As $\lim_{\epsilon \to 0} \vartheta = L + \epsilon$, there exist locally optimal combinations of (α, β) such that the output distribution of PQA preserves the maximum amount of information from the original activations, achieving $\mathcal{H}_{\text{PQA}} \approx \mathcal{H}_{\text{BN}}$. To determine the optimal combinations of (α, β), we perform a grid search within a bounded interval to systematically evaluate entropy values under different parameter configurations. The result of the grid search further supports the theoretical analysis.

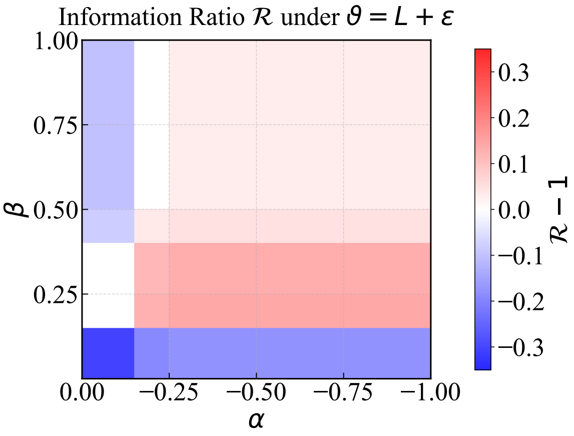

Figure 2: The grid search results of $\mathcal{R} = \frac{\mathcal{H}_{\text{PQA}}}{\mathcal{H}_{\text{BN}}}$ under the condition $L = \vartheta = 8$, with $\alpha \in [-1, 0]$ and $\beta \in (0, 1]$. The colormap visualizes the value of $\mathcal{R} - 1$: Bluer regions indicate more information loss during transmission. White regions denote lossless transmission. Redder regions reflect stronger distortion during transmission.

As illustrated in Fig. 2, appropriate settings of L, ϑ, α, and β can maximize the representational capacity of discretized activations. This, in turn, enhances feature retention in the quantized activation function, as illustrated in Fig. 1(b). Excessive expansion of the output range increases the encoding base, but paradoxically reduces effective resolution, leading to information distortion. The distribution of the entropy ratio $\mathcal{R}$, visualized through color gradients in Fig. 2, thereby offers practical guidance for parameter tuning in future quantization strategies.

Augmented Multi-Spiking IF neurons

Prior works explored neurons with polarity and multi-spike mechanism (Wang et al. 2022, 2024a; Yu et al. 2022a; Han et al. 2023), which enhanced neuronal expressiveness but still required a large number of timesteps to approximate ANNs activations. To overcome this limitation, we propose an Augmented Integrate-and-Fire (AIF) neuron model that enables neurons to emit multiple spikes within a single timestep. This design significantly enhances the instantaneous representational capacity of spiking neurons while preserving their event-driven nature, thereby laying a theoretical foundation for nearly lossless ANN-to-SNN conversion. Specifically, an AIF neuron at layer l receives the output $s^{l-1}(t)$ from the previous layer $l-1$ at timestep t:

$$o^l(t) = W^l_{\text{SNN}} s^{l-1}(t), \quad W^l_{\text{SNN}} = W^l \vartheta^l_{\text{SNN}} \tag{3}$$

$$m^l(t) = v^l(t-1) + o^l(t) \tag{4}$$

Here W^l_{SNN} denotes the weight of the l-th layer in the SNNs, converted from the corresponding weight matrix W^l in the ANNs scaled by ϑ^l_{SNN}. ϑ^l_{SNN} denotes the threshold of the l-th layer. The membrane potential of the AIF neuron at layer l before spike firing is denoted by $m^l(t)$. When

$m^l(t) > 0$, the neuron emits spikes proportional to the ratio between the membrane potential and the threshold ϑ^l_{SNN}. Conversely, when $m^l(t) < 0$, the neuron emits negative spikes until the membrane potential returns to the interval $[0, \vartheta^l_{\text{SNN}})$ or $(-\vartheta^l_{\text{SNN}}, 0]$. To ensure stable firing, we impose upper bounds on the number of positive and negative spikes per timestep, denoted by C_{pos} and C_{neg}, respectively. The output of the AIF neuron at timestep t is thus defined as:

$$s^l(t) = g\left(m^l(t)\right) = \text{clip}\left(\left\lfloor \frac{m^l(t)}{\vartheta^l_{\text{SNN}}} \right\rceil, C_{\text{neg}}, C_{\text{pos}}\right) \quad (5)$$

To avoid information loss, we apply a soft reset mechanism (Han, Srinivasan, and Roy 2020), in which only the cumulative threshold associated with the emitted spikes is subtracted after firing, rather than resetting the membrane potential to zero. The membrane potential $v^l(t)$ is then updated as:

$$v^l(t) = m^l(t) - \vartheta^l_{\text{SNN}} \cdot s^l(t) \quad (6)$$

Conversion Error Analysis

The core mechanism of ANN-to-SNN conversion is to approximate activation values in ANNs using the average postsynaptic potential (Bu et al. 2022b; Huang et al. 2024; Rudnicka, Szczepanski, and Pregowska 2024; You et al. 2024; Wang et al. 2023b) (or the firing rate (Cao, Chen, and Khosla 2015; Rueckauer et al. 2017; Tan, Patel, and Kozma 2021)) of neurons in SNNs. Specifically, at the l-th layer, the average post-synaptic potential of neurons over T timesteps is denoted by $\phi^l(T)$ and defined as follows:

$$\phi^l(T) = \frac{\sum_{i=1}^{T} s^l(i)\vartheta^l_{\text{SNN}}}{T} \quad (7)$$

To analyze the conversion error, we assume that ANNs and SNNs receive the same weighted input $z^l = W^l x^{l-1} = W^l_{\text{SNN}} s^{l-1}(t)$ at the l-th layer, and compare their corresponding outputs. The absolute conversion error, denoted Err^l, is defined as $Err^l = \phi^l(T) - y^l$, where $\phi^l(T)$ is the average post-synaptic potential and y^l is the ANN's activation value:

$$Err^l = \vartheta^l_{\text{SNN}} \frac{\sum_{i=1}^{T} s^l(i)}{T} - \vartheta^l clip(\frac{1}{L}\left\lfloor \frac{z^l L}{\vartheta^l} \right\rceil, \alpha, \beta) \quad (8)$$

Here, $\vartheta^l = L \cdot \vartheta^l_{\text{SNN}}$ defines the relationship between the threshold values in SNNs and the ones in ANNs. Considering that AIF neurons can emit multiple spikes, we express C_{neg} and C_{pos} as αL and βL respectively. The above equation can thus be further simplified as follows:

$$\begin{aligned} Err^l = \vartheta^l_{\text{SNN}} \Bigg(\frac{1}{T} \sum_{i=1}^{T} \text{clip}\left(\left\lfloor \frac{v^l(i-1) + z^l(i)}{\vartheta^l_{\text{SNN}}} \right\rceil, \alpha L, \beta L\right) \\ - \text{clip}\left(\left\lfloor \frac{z^l}{\vartheta^l_{\text{SNN}}} \right\rceil, \alpha L, \beta L\right) \Bigg) \end{aligned} \quad (9)$$

Since both internal terms are constrained within the range $[\alpha L, \beta L]$, the clipping function $\text{clip}(\cdot)$ can be treated as the identity function, thereby leading to an approximate expression:

$$Err^l \approx \frac{1}{T} \sum_{i=1}^{T} \left\lfloor \frac{v^l(i-1) + z^l(i)}{\vartheta^l_{\text{SNN}}} \right\rceil - \left\lfloor \frac{z^l}{\vartheta^l_{\text{SNN}}} \right\rceil \quad (10)$$

The conversion error stems from SNNs' temporal dependency, where the membrane potential $v^l(i-1)$ modulates current inputs $x^l(i)$, causing deviations in postsynaptic potentials from ANNs activation values. According to our derivation, the error can be eliminated under the following three conditions: the PQA function and the spiking neuron threshold satisfy $\vartheta = L \cdot \vartheta^l_{\text{SNN}}$; the clipping bounds satisfy $C_{\text{neg}} = \alpha L$ and $C_{\text{pos}} = \beta L$; and the neuron, receiving a static input, is initialized with $v^l(0) = \frac{1}{2}\vartheta_{\text{SNN}}$. Under these conditions, the activation values of ANNs can be accurately reconstructed. Moreover, as the number of timesteps increases, SNNs can still leverage temporal accumulation to asymptotically approximate activation values of ANNs.

Experiments

We evaluated our proposed method on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), as well as ImageNet (Deng et al. 2009), using three widely used architectures: VGG-16 (Simonyan and Zisserman 2014), ResNet-20 (He et al. 2016), and ViT-S (Dosovitskiy et al. 2020) that represent plain deep CNNs, ANNs with residual connections and large-scale models respectively, thus providing a comprehensive evaluation of the method in terms of performance and generalizability. In experiments on the ViT-S architecture, we drew upon existing research (You et al. 2024) regarding conversion schemes for attention blocks. During the training phase, we replaced the original activation function with the PQA function. In the conversion phase, we substituted the original ST-BIF$^+$ (You et al. 2024) neurons with AIF neurons. All experiments were conducted on a NVIDIA RTX 4090 GPU.

Comparison with the State of the Art

Table 1 and Table 2 present the conversion results of our method on CIFAR-10 and CIFAR-100 using the VGG-16, ResNet-20, and ViT-S architectures, respectively. The results demonstrate that our proposed PMSM achieves competitive performance across convolutional, residual, and attention-based models, validating its effectiveness and generality across various mainstream architectures. Furthermore, as shown in Table 2, PMSM surpasses previous state-of-the-art methods even under extremely low latency, achieving superior performance at the first timestep. Notably, we observe a marginal accuracy drop in ANN-to-SNN conversion on ViT-S. This is mainly because we intentionally retained the baseline's SNN-friendly treatment of attention blocks (You et al. 2024), which requires a large number of timesteps to match the outputs of SNNs with those of their ANNs counterparts. Nevertheless, the resulting SNNs with

Table 1: Performance on CIFAR-10 and CIFAR-100 with VGG-16 and ResNet-20 architectures. We compare our method with existing state-of-the-art approaches. The method marked with * is obtained via direct training.

Method	CIFAR-10						CIFAR-100					
	VGG-16			ResNet-20			VGG-16			ResNet-20		
	ANN	T	Acc	ANN	T	Acc	ANN	T	Acc	ANN	T	Acc
RMP-L* (Guo et al. 2023)	N/A	4	93.33	N/A	4	91.89	N/A	4	72.55	N/A	4	66.65
OPI (Bu et al. 2022a)	94.57	8	90.96	92.74	8	66.24	76.31	8	60.49	70.43	8	23.09
Off-LTL (Yang et al. 2022)	94.05	16	93.04	95.36	16	94.82	74.42	16	74.19	76.36	16	76.12
On-LTL (Yang et al. 2022)	94.05	16	92.85	95.36	16	91.33	74.42	16	71.09	76.36	16	73.22
SRP (Hao et al. 2023a)	95.52	8	95.52	91.77	8	91.37	76.28	8	76.25	69.94	8	62.94
QCFS (Bu et al. 2022b)	95.52	1	88.41	91.77	1	62.43	76.28	8	73.96	69.94	8	55.37
SlipReLU (Jiang et al. 2023)	93.82	1	88.17	82.07	1	80.99	68.46	1	64.21	50.79	1	48.12
COS (Hao et al. 2023b)	95.51	1	94.90	91.77	1	89.88	76.28	1	74.24	69.97	1	59.22
EMORE (Huang et al. 2024)	95.21	1	88.46	85.18	1	65.99	74.86	1	62.27	62.34	1	21.78
TSO (Wang et al. 2025)	95.73	8	95.26	96.66	8	95.46	77.22	8	75.06	-	-	-
Ours	**95.67**	**1** **8**	**95.67** **95.62**	**93.78**	**1** **8**	**93.78** **93.75**	**76.71**	**1** **8**	**76.71** **76.82**	**69.91**	**1** **8**	**69.91** **70.24**

Table 2: Performance comparison across ViT-S architecture and different datasets with existing mapping approaches (including SpikedAttention (Hwang et al. 2024), MST (Wang et al. 2023b), SpikeZIP-TF (You et al. 2024)) and direct training methods (including SpikFormer (Zhou et al. 2023) and Sdformer (Yao et al. 2023)).

Method	**Dataset**	**ANN**	**T**	**Acc**
SpikFormer	CIFAR-10	-	4	95.5
Sdformer		-	4	95.6
SpikedAttention		97.5	24	97.3
MST		98.1	256	97.3
SpikeZIP-TF		99.2	16	97.7
Ours		**98.8**	**1** **16**	**98.5** **98.6**
SpikFormer	CIFAR-100	-	4	78.2
Sdformer		-	4	78.2
SpikedAttention		87.7	24	86.3
MST		88.7	256	86.9
SpikeZIP-TF		91.9	16	87.3
Ours		**90.4**	**1** **16**	**89.3** **89.4**
SpikFormer	ImageNet	-	4	74.8
Sdformer		-	4	77.1
SpikedAttention		79.3	48	77.2
MST		80.5	512	78.5
SpikeZIP-TF		82.3	64	81.4
Ours		**82.3**	**1** **16**	**81.6** **81.7**

our method still achieve the highest accuracy and the lowest number of time steps.

Hyperparameter Validation on PQA Information Entropy

To systematically evaluate the effectiveness of our entropy-based strategy, we conducted ablation studies on the VGG-16 (Simonyan and Zisserman 2014) and ResNet-20 (He et al. 2016) architectures using the CIFAR-10 and CIFAR-100 datasets, focusing on hyperparameters α and β. With $L = \vartheta = 8$ fixed, a grid search revealed that α and β modulate the entropy ratio $\mathcal{R} = \frac{\mathcal{H}_{\text{PQA}}}{\mathcal{H}_{\text{BN}}}$. As shown in Fig. 2, when $\beta = 1.0$ and α decreases from -0.125 to -1, the entropy ratio $\mathcal{R}$ exhibits three distinct phases: transmission with information loss, lossless transmission over a certain range of α, and transmission with the introduction of noise as α decreases further. This progression aligns precisely with the accuracy trends across diverse models and tasks, as shown in Fig. 3(a), where model accuracy first increases and then drops. A similar pattern emerges when fixing α and increasing β from 0.125 to 1, as depicted in Fig. 3(b). These results demonstrate that the $\mathcal{R}$-guided hyperparameter search strategy effectively generalizes across model architectures and tasks.

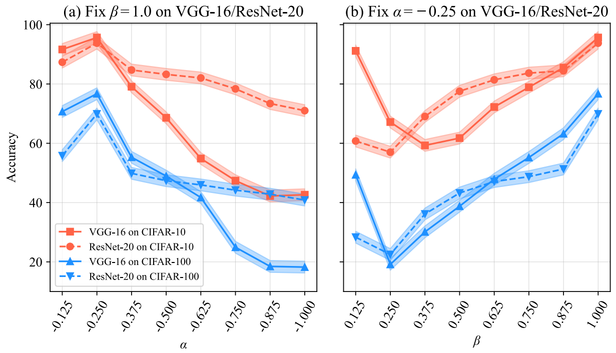

Figure 3: Accuracy variation with different α and β values on VGG-16 and ResNet-20 architectures for CIFAR-10 and CIFAR-100 datasets. (a) Performance when β is fixed at 1.0, (b) Performance when α is fixed at -0.25.

Energy Analysis

We evaluate the energy efficiency of our proposed method using a spike-count-based strategy (Cao, Chen, and Khosla

2015), which has been widely adopted in recent studies (Ding et al. 2021; Wang et al. 2023b; You et al. 2024). An AIF neuron output of $\pm n$ is counted as n spikes within that timestep. Based on the performance characteristics of 45nm hardware (Horowitz 2014), each spike consumes 0.9 pJ of energy, and each timestep lasts 1 millisecond. Under these assumptions, the average power consumption P is computed as:

$$P = \frac{N}{T \times \eta} \times \xi \tag{11}$$

Here N denotes the total number of spikes (in 10^8). T is the total number of timesteps. η is the duration of a single timestep (set to 10^{-3} seconds), and ξ represents the energy per spike (0.9 pJ). As shown in Table 3, under identical timestep settings, our method achieves a more than 5× reduction in energy consumption compared to QCFS (Bu et al. 2022b), while simultaneously yielding higher classification accuracy. Furthermore, Figure 4 illustrates the layer-wise spike activity, indicating that our model exhibits fewer spikes per layer, which leads to reduced energy consumption.

Table 3: VGG-16 power consumption comparison between our method and reproduced QCFS (Bu et al. 2022b). Ratio is computed as $\frac{P_{\text{QCFS}}}{P_{\text{Ours}}}$ under the same setting.

Method	**ANN**	**Acc**	T	N	P	**Ratio**
			CIFAR-10			
QCFS	95.52	72.26	1	3.08	0.278	(5.1×)
Ours	95.56	95.56	1	0.61	0.055	
QCFS	95.52	86.26	2	6.44	0.290	(5.3×)
Ours	95.56	95.63	2	1.23	0.055	
			CIFAR-100			
QCFS	76.28	47.05	1	3.38	0.305	(5.9×)
Ours	76.53	76.54	1	0.58	0.052	
QCFS	76.28	57.56	2	7.36	0.331	(6.4×)
Ours	76.53	76.64	2	1.16	0.052	

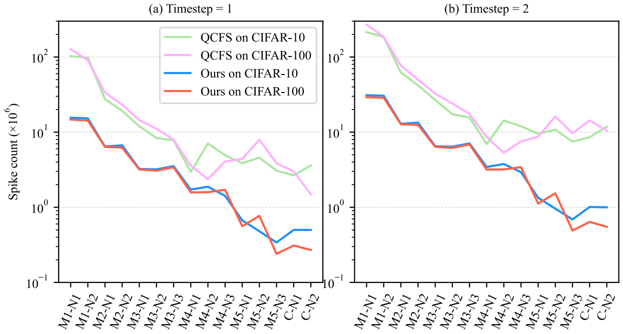

Figure 4: Comparison of layer-wise spike counts between our method and QCFS (Bu et al. 2022b) on VGG-16 with different timestep.The x-axis shows neuron labels, where Mx-Ny and C-Ny denote the y-th neuron layer in Module-x and the Classifier module of VGG-16, respectively.

Discussion on Hardware Compatibility

While our proposed conversion framework achieves accurate and low-latency ANN-to-SNN mapping through the use of both multiple spikes and polarity, it is also designed with practical deployment in mind. Specifically, our method is hardware-friendly and can be adapted to both existing and emerging neuromorphic platforms with minimal modification.

IBM TrueNorth (Merolla et al. 2014) is a digital neuromorphic chip, where each spike is transmitted as a 1-bit event. In such systems, representing negative spikes would require separate pathways for positive and negative signals. In contrast, SpiNNaker (Furber et al. 2014) supports 32-bit event packets, enabling each spike to carry additional information such as polarity and count. This makes it possible to realize our multi-spike and polarity-based mapping through software-defined encoding. Similarly, Intel Loihi 2 (Orchard et al. 2021) supports custom neuron models, signed synaptic weights, and multi-bit spike signaling, making it well-suited for the direct and efficient implementation of our method.

Importantly, the theoretical framework developed in this work reveals properties that are not only compatible with current hardware but also valuable for guiding the design of future neuromorphic chips. The ability to achieve nearly lossless ANN-to-SNN conversion within a single timestep, while maintaining stable performance across time steps, presents a compelling target for hardware–software co-design.

Conclusion

This paper presents an ANN-to-SNN conversion framework targeting high accuracy and low latency, incorporating a Polarity Quantized Activation (PQA) function and Augmented Integrate-and-Fire (AIF) neurons to faithfully approximate the activation values of ANNs at the first timestep. The PQA function adopts an entropy-guided quantization strategy that preserves activation distribution under bit-width representation, thereby ensuring lossless information transfer in the quantization domain and serving as a solid foundation for accurate and efficient conversion. AIF neurons emit multiple spikes with polarity by leveraging both positive and negative thresholds, allowing SNNs to approximate quantized activations at the first timestep, thus reducing inference latency while maintaining accuracy. We further provide a theoretical analysis of the activation shift, revealing a membrane potential modulation effect induced by the temporal dynamics of SNNs. Specifically, we prove that with extremely few timesteps ($T = 1$), the conversion error is approximately zero. Experimental results on CIFAR-10, CIFAR-100, and ImageNet demonstrate that PMSM surpasses existing ANN-to-SNN methods in accuracy, without requiring fine-tuning or additional training, and generalizes well across diverse network architectures. Although PMSM shows competitive performance on the image classification benchmarks, the applicability to more complex tasks such as object detection and semantic segmentation remains to be explored. Overall, this work represents a promising step toward accurate and efficient ANN-to-SNN conversion with ultra-low latency.

References

Bohte, S. M.; Kok, J. N.; and La Poutré, J. A. 2000. SpikeProp: backpropagation for networks of spiking neurons. In *ESANN*, volume 48, 419–424. Bruges.

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*. arXiv. ArXiv:2303.04347 [cs].

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.

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.

Deng, L.; Wu, Y.; Hu, X.; Liang, L.; Ding, Y.; Li, G.; Zhao, G.; Li, P.; and Xie, Y. 2020. Rethinking the performance comparison between SNNS and ANNS. *Neural Networks*, 121: 294–307.

Deng, S.; and Gu, S. 2021. Optimal Conversion of Conventional Artificial Neural Networks to Spiking Neural Networks. In *International Conference on Learning Representations*.

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. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2328–2336. Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6.

Ding, Y.; Zuo, L.; Jing, M.; He, P.; and Xiao, Y. 2024. Shrinking Your TimeStep: Towards Low-Latency Neuromorphic Object Recognition with Spiking Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38 of *10*, 11811–11819.

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*.

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

Gao, H.; He, J.; Wang, H.; Wang, T.; Zhong, Z.; Yu, J.; Wang, Y.; Tian, M.; and Shi, C. 2023. High-accuracy deep ANN-to-SNN conversion using quantization-aware training framework and calcium-gated bipolar leaky integrate and fire neuron. *Frontiers in Neuroscience*, Volume 17 - 2023.

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.

Guo, Y.; Liu, X.; Chen, Y.; Zhang, L.; Peng, W.; Zhang, Y.; Huang, X.; and Ma, Z. 2023. RMP-Loss: Regularizing Membrane Potential Distribution for Spiking Neural Networks. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 17345–17355. Paris, France: IEEE. ISBN 9798350307184.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13555–13564. Seattle, WA, USA: IEEE. ISBN 978-1-72817-168-5.

Han, J.; Wang, Z.; Shen, J.; and Tang, H. 2023. Symmetric-threshold ReLU for Fast and Nearly Lossless ANN-SNN Conversion. *Machine Intelligence Research*, 20(3): 435–446.

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*. ArXiv:2302.10685 [cs].

Hao, Z.; Shi, X.; Liu, Y.; Yu, Z.; and Huang, T. 2024. LM-HT SNN: Enhancing the Performance of SNN to ANN Counterpart through Learnable Multi-hierarchical Threshold Model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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.

Horowitz, M. 2014. 1.1 Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14.

Huang, Z.; Ding, J.; Pan, Z.; Li, H.; Fang, Y.; Yu, Z.; and Liu, J. K. 2024. Converting High-Performance and Low-Latency SNNs through Explicit Modelling of Residual Error in ANNs. *arXiv preprint arXiv:2404.17456*.

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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jang, H.; Simeone, O.; Gardner, B.; and Gruning, A. 2019. An introduction to probabilistic spiking neural networks:

Probabilistic models, learning rules, and applications. *IEEE Signal Processing Magazine*, 36(6): 64–77.

Jiang, H.; Anumasa, S.; Masi, G. D.; Xiong, H.; and Gu, B. 2023. A Unified Optimization Framework of ANN-SNN Conversion: Towards Optimal Mapping from Activation Values to Firing Rates. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 14945–14974. PMLR.

Kepecs, A.; Wang, X.-J.; and Lisman, J. 2002. Bursting Neurons Signal Input Slope. *Journal of Neuroscience*, 22(20): 9053–9062.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada.

Lan, Y.; Zhang, Y.; Ma, X.; Qu, Y.; and Fu, Y. 2023. Efficient Converted Spiking Neural Network for 3D and 2D Classification. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9177–9186. Paris, France: IEEE. ISBN 9798350307184.

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

Li, Y.; Deng, S.; Dong, X.; Gong, R.; and Gu, S. 2021. A Free Lunch From ANN: Towards Efficient, Accurate Spiking Neural Networks Calibration. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6316–6325. PMLR.

Li, Y.; Deng, S.; Dong, X.; and Gu, S. 2022. Converting Artificial Neural Networks to Spiking Neural Networks via Parameter Calibration. *arXiv preprint arXiv:2205.10121*.

Liu, C.; Shen, J.; Ran, X.; Xu, M.; Xu, Q.; Xu, Y.; and Pan, G. 2025. Efficient ANN-SNN Conversion with Error Compensation Learning. In *Forty-second International Conference on Machine Learning*.

Liu, F.; Zhao, W.; Chen, Y.; Wang, Z.; and Jiang, L. 2022. SpikeConverter: An Efficient Conversion Framework Zipping the Gap between Artificial Neural Networks and Spiking Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1692–1701.

Ma, C.; and Yu, Q. 2020. AugMapping: Accurate and Efficient Inference with Deep Double-Threshold Spiking Neural Networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2002–2007.

Merolla, P. A.; Arthur, J. V.; Alvarez-Icaza, R.; Cassidy, A. S.; Sawada, J.; Akopyan, F.; Jackson, B. L.; Imam, N.; Guo, C.; Nakamura, Y.; Brezzo, B.; Vo, I.; Esser, S. K.; Appuswamy, R.; Taba, B.; Amir, A.; Flickner, M. D.; Risk, W. P.; Manohar, R.; and Modha, D. S. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197): 668–673.

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.

Orchard, G.; Frady, E. P.; Rubin, D. B. D.; Sanborn, S.; Shrestha, S. B.; Sommer, F. T.; and Davies, M. 2021. Efficient Neuromorphic Signal Processing with Loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 254–259.

Rudnicka, Z.; Szczepanski, J.; and Pregowska, A. 2024. Artificial Intelligence-Based Algorithms in Medical Image Scan Segmentation and Intelligent Visual Content Generation—A Concise Overview. *Electronics*, 13(4).

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.

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

Song, S.; Ma, C.; Sun, W.; Xu, J.; Dang, J.; and Yu, Q. 2021. Efficient learning with augmented spikes: A case study with image classification. *Neural Networks*, 142: 205–212.

Tan, W.; Patel, D.; and Kozma, R. 2021. Strategy and Benchmark for Converting Deep Q-Networks to Event-Driven Spiking Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11): 9816–9824.

Wang, B.; Cao, J.; Chen, J.; Feng, S.; and Wang, Y. 2023a. A New ANN-SNN Conversion Method with High Accuracy, Low Latency and Good Robustness. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 3067–3075. Macau, SAR China: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-03-4.

Wang, Y.; Liu, H.; Zhang, M.; Luo, X.; and Qu, H. 2024a. 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 *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2501–2508. Vienna, Austria: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3.

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

Wang, Z.; Lil, S.; Ma, Z.; and Yao, Q. 2024b. High Accurate, Low Latency Conversion of Spiking Neural Networks with BLIF Neurons. In *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, 432–440.

Wang, Z.; Zhang, Y.; Lian, S.; Cui, X.; Yan, R.; and Tang, H. 2025. Toward High-Accuracy and Low-Latency Spiking Neural Networks With Two-Stage Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2): 3189–3203.

Yang, H.; Yang, S.; Zhang, L.; Dou, H.; Shen, F.; and Zhao, J. 2025. CS-QCFS: Bridging the performance gap in ultra-low latency spiking neural networks. *Neural Networks*, 184: 107076.

Yang, Q.; Wu, J.; Zhang, M.; Chua, Y.; Wang, X.; and Li, H. 2022. Training Spiking Neural Networks with Local Tandem Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 12662–12676. Curran Associates, Inc.

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

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 57367–57383. PMLR.

Yu, Q.; Ma, C.; Song, S.; Zhang, G.; Dang, J.; and Tan, K. C. 2022a. Constructing Accurate and Efficient Deep Spiking Neural Networks With Double-Threshold and Augmented Schemes. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4): 1714–1726.

Yu, Q.; Song, S.; Ma, C.; Pan, L.; and Tan, K. C. 2022b. Synaptic Learning With Augmented Spikes. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3): 1134–1146.

Zhang, G.; Li, B.; Wu, J.; Wang, R.; Lan, Y.; Sun, L.; Lei, S.; Li, H.; and Chen, Y. 2020. A low-cost and high-speed hardware implementation of spiking neural network. *Neurocomputing*, 382: 106–115.

Zhang, H.; Zhang, S.; Mao, W.; and Wang, Z. 2025a. An Efficient Brain-Inspired Accelerator Using a High-Accuracy Conversion Algorithm for Spiking Deformable CNN. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 72(1): 288–292.

Zhang, M.; Wang, S.; Wu, J.; Wei, W.; Zhang, D.; Zhou, Z.; Wang, S.; Zhang, F.; and Yang, Y. 2025b. Toward Energy-Efficient Spike-Based Deep Reinforcement Learning With Temporal Coding. *IEEE Computational Intelligence Magazine*, 20(2): 45–57.

Zhou, F.; Fu, M.; Gao, Y.; Wang, B.; and Yu, Q. 2024. Rethinking spikes in spiking neural networks for performance enhancement. In *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 374–379.

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

Appendix

Information Entropy Analysis

Entropy of Batch Normalization Outputs. Batch Normalization (BN) layers normalize input features to follow a standard normal distribution with zero mean and unit variance. Based on the probability density function of the standard normal distribution, the information entropy of the BN output, denoted as $\mathcal{H}_{\text{BN}}$, is given by:

$$\begin{aligned}\mathcal{H}_{\text{BN}}(x) &= -\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \log(\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}})\, dx \\ &= -\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot \frac{-\log(2\pi) - x^2}{2}\, dx \\ &= \frac{1}{2}\left(\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot \log 2\pi\, dx \right. \\ &\qquad \left. \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot x^2\, dx\right)\end{aligned} \tag{12}$$

According to the definition of variance, $D(x) = E[x^2] - (E[x])^2$, the integral term in the above equation can be evaluated as follows:

$$\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot x^2\, dx = 1 \tag{13}$$

Hence, the entropy of the BN output is:

$$\mathcal{H}_{\text{BN}}(x) = \frac{1}{2}(\log 2\pi e) \tag{14}$$

Entropy Maximization of Traditional Quantized Activation Functions. A traditional quantized activation (QA) function is defined as:

$$Y = f(x) = \vartheta \cdot \text{clip}\left(\frac{1}{L}\left\lfloor\frac{xL}{\vartheta}\right\rfloor, 0, 1\right) \tag{15}$$

Here ϑ is a learnable quantization threshold. L specifies the number of quantization levels, and x represents the input. The clip function restricts the output to the range $[0, 1]$. Thus, the output set of the function is defined as $\mathcal{Y} = \left\{y_k = \vartheta \cdot \frac{k}{L} | k = 0, 1, \ldots, L\right\}$. Then, the information entropy of the QA output can be calculated as follows:

$$\mathcal{H}_{\text{QA}} = -\sum_{k=0}^{L} P(y_k) \log P(y_k) \tag{16}$$

Here, $P(y_k)$ denotes the probability of output y_k. According to the definition of entropy, the entropy attains its maximum value when the output distribution is uniform over all possible outcomes. In this case, $P(y_k) = \frac{1}{L+1}$, and the maximum entropy is given by $\mathcal{H}_{\text{QA}}^{\max} = \log(L + 1)$. Theoretically, as $L \to \infty$, the upper bound of entropy also increases without limit. However, due to the finite quantization threshold ϑ, when the input $x \to \infty$, the output of the QA function becomes saturated at ϑ, which means that $f(x) = \vartheta$. This saturation effect significantly increases $P(y_L)$ while reducing the probabilities of other quantized values, leading to a decline in entropy. In summary, to maintain a uniform output distribution and maximize information entropy, both $L \to \infty$ and $\vartheta \to \infty$ must be satisfied simultaneously. That is, only under the limiting condition of infinite quantization resolution and unbounded threshold range can the representational capacity of traditional QA functions approach its theoretical upper bound. Under these conditions, Equation 15 simplifies to:

$$\lim_{\vartheta\to\infty}\lim_{L\to\infty} f(x) = \max(0,x) = \mathrm{ReLU}(x) \tag{17}$$

That is, the QA function becomes equivalent to the ReLU activation function when its information entropy reaches the maximum. Consequently, the output entropy of the ReLU function can be regarded as the theoretical upper bound of the information representation capacity of the QA function. When the input follows a standard normal distribution with zero mean and unit variance, the probability density function (PDF) of the ReLU output is given by:

$$f_{\mathrm{ReLU}}(x) = \begin{cases} 0, & x < 0 \\ 0.5 \cdot \delta(x), & x = 0 \\ \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, & x > 0 \end{cases} \tag{18}$$

Here, $\delta(\cdot)$ denotes the Dirac delta function, representing a point mass centered at $x = 0$ with a probability weight of 0.5. Therefore, the output entropy of the ReLU function can be expressed as follows:

$$\begin{aligned}\mathcal{H}_{\mathrm{ReLU}}(x) =& -0.5\log 0.5 - \int_0^\infty f_{\mathrm{ReLU}}(x)\log f_{\mathrm{ReLU}}(x)\,dx \\ =&0.5\log 2 - \int_0^\infty f_{\mathrm{ReLU}}(x)\left(-\frac{1}{2}\log(2\pi)\right. \\ &\left.-\frac{x^2}{2}\right)dx \\ =&0.5\log 2 + \frac{1}{2}\log(2\pi)\int_0^\infty f_{\mathrm{ReLU}}(x)\,dx \\ &+\frac{1}{2}\int_0^\infty f_{\mathrm{ReLU}}(x)x^2dx \\ =&0.5\log 2 + \frac{1}{2}\log(2\pi)\cdot 0.5 + \frac{1}{2}\cdot 0.5 \\ \approx&0.69\mathcal{H}_{\mathrm{BN}}(x)\end{aligned} \tag{19}$$

In conclusion, the information entropy of the ReLU function under standard normal input is approximately 69% of the entropy of the BN output. This value can also be regarded as the upper bound of the information representation capacity that the QA function can achieve under ideal conditions, that is, $L \to \infty, \vartheta \to \infty$. Therefore, $\mathcal{H}_{\mathrm{QA}}^{\max} = \mathcal{H}_{\mathrm{ReLU}} \approx 0.69\mathcal{H}_{\mathrm{BN}}$.

Entropy of PQA. Building on the Polarity Quantization Activation (PQA) function, we define an intermediate variable $k = \left\lfloor \frac{xL}{\vartheta} \right\rceil$, where $k \in \mathbb{Z}$. Since $x \sim \mathcal{N}(0,1)$, it follows that k follows a normal distribution with mean 0 and variance $\frac{L^2}{\vartheta^2}$, and takes values in $\mathbb{Z}$. For convenience, we assume that L_α and L_β are integers. After scaling, the minimum and maximum values of k are given by $k_{\min} = L\alpha$ and $k_{\max} = L\beta$. For any integer k, from the inequality $k \le \frac{xL}{\vartheta} < k+1$, we can derive the corresponding interval $\frac{\vartheta}{L}k \le x < \frac{\vartheta}{L}(k+1)$. Therefore, without clipping, the probability of k is:

$$\begin{aligned}P(k=n) &= P\left(n-\tfrac{1}{2}, n+\tfrac{1}{2}\right)) \\ &= \Phi\left(\frac{n+\frac{1}{2}}{L/\vartheta}\right) - \Phi\left(\frac{n-\frac{1}{2}}{L/\vartheta}\right)\end{aligned} \tag{20}$$

Here, $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. Due to the clipping operation, the probability mass function of y consists of three cases, depending on whether the input x falls below $\vartheta\alpha$, within the range $[\vartheta\alpha, \vartheta\beta]$, or above $\vartheta\beta$. Accordingly, the probability mass function of y can be expressed as:

$$\begin{aligned}P_Y(y) =\; & \delta(y-\vartheta\alpha)\sum_{k=-\infty}^{\lfloor L\alpha\rfloor-1} p(k) \\ &+\sum_{k=\lceil L\alpha\rceil}^{\lfloor L\beta\rfloor}\delta\left(y-\vartheta\frac{k}{L}\right)p(k) \\ &+\delta(y-\vartheta\beta)\sum_{k=\lfloor L\beta\rfloor+1}^{+\infty} p(k)\end{aligned} \tag{21}$$

Here, $p(k) = \Phi\left(\frac{\vartheta}{L}(k+1)\right) - \Phi\left(\frac{\vartheta}{L}k\right)$, where $\delta(\cdot)$ denotes the Dirac delta function, which captures the discrete probability mass at specific points. Accordingly, the information entropy of y consists of a continuous term $\mathcal{H}_2$ and two discrete terms, $\mathcal{H}_1$ and $\mathcal{H}_3$, and is given by:

$$\begin{aligned}&\mathcal{H}_{\mathrm{PQA}} = \mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3 \\ &\quad\mathcal{H}_1 = -\Phi\left(\frac{k_{\mathrm{neg}}-\frac{1}{2}}{L/\vartheta}\right)\log(\Phi\left(\frac{k_{\mathrm{neg}}-\frac{1}{2}}{L/\vartheta}\right)) \\ &\quad\mathcal{H}_2 = -\sum_{k=k_{\mathrm{neg}}}^{k=k_{\mathrm{pos}}}\left[\left(\Phi\left(\frac{k+\frac{1}{2}}{L/\vartheta}\right) - \Phi\left(\frac{k-\frac{1}{2}}{L/\vartheta}\right)\right)\right. \\ &\qquad\qquad\left.\cdot\log\left(\Phi\left(\frac{k+\frac{1}{2}}{L/\vartheta}\right) - \Phi\left(\frac{k-\frac{1}{2}}{L/\vartheta}\right)\right)\right] \\ &\quad\mathcal{H}_3 = -\left[1-\Phi\left(\frac{k_{\mathrm{pos}}-\frac{1}{2}}{L/\vartheta}\right)\right] \\ &\qquad\qquad\cdot\log\left(1-\Phi\left(\frac{k_{\mathrm{pos}}-\frac{1}{2}}{L/\vartheta}\right)\right)\end{aligned} \tag{22}$$

The terms $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ share a common factor $\frac{k-\frac{1}{2}}{L/\vartheta}$, which can be simplified to:

$$\frac{k-\frac{1}{2}}{L/\vartheta} = (k-\tfrac{1}{2})\cdot\frac{\vartheta}{L} \tag{23}$$

In Equation (23), the ratio $\frac{\vartheta}{L}$ determines the scaling of k, thereby affecting the probability distribution that defines $\mathcal{H}_{\mathrm{PQA}}$. To analyze its influence, we consider three distinct cases based on the value of $\frac{\vartheta}{L}$:

Case 1. When$\vartheta = L + \epsilon, \epsilon \to 0$:

When $\frac{\vartheta}{L} \approx 1$, $\mathcal{H}_{\text{PQA}}$ attain deterministic values governed by α, β, ϑ, and L. For fixed ϑ and L, all possible values of $\mathcal{H}_{\text{PQA}}$ in this regime can be computed via grid search, as illustrated in Fig. 2. Based on the previously computed entropy of BN outputs, we conclude that when $\vartheta = L + \epsilon$ and $\epsilon \to 0$, there exists a configuration of hyperparameters such that $\mathcal{H}_{\text{PQA}} \approx \mathcal{H}_{\text{BN}}$.

Case 2. When $\vartheta \ll L$:

The ratio $\frac{\vartheta}{L}$ is negligible, causing the term in Equation (23) to tend to zero. Consequently:

$$\Phi\left(\left(k - \tfrac{1}{2}\right) \cdot \frac{\vartheta}{L}\right) \approx \Phi(0) = 0.5 \tag{24}$$

It follows from Equation (22) that:

$$\mathcal{H}_1 = -0.5 \log 0.5, \quad \mathcal{H}_2 = 0, \quad \mathcal{H}_3 = -0.5 \log 0.5 \tag{25}$$

From the previously computed entropy of BN outputs, we conclude that when $\vartheta \ll L$, the entropy reduces to $\mathcal{H}_{\text{PQA}} = -\log 0.5$. The resulting entropy ratio is given by: $\frac{\mathcal{H}_{\text{PQA}}}{\mathcal{H}_{\text{BN}}} = \frac{-\log 0.5}{\frac{1}{2}\log(2\pi e)} \approx 0.49$.

As shown in Fig. 5(a), we consider the setting $L = 20$ and $\vartheta = 1$ as an illustrative example, then evaluate the corresponding entropy ratio $\mathcal{R}$ over $\alpha \in [-1, 0]$ and $\beta \in (0, 1]$. The results indicate that this configuration results in severe information loss throughout the entire value ranges of α and β. Therefore, choosing $\vartheta \ll L$ is generally suboptimal when setting hyperparameters.

Case 3. When $\vartheta \gg L$:

The ratio $\frac{\vartheta}{L}$ becomes arbitrarily large, and Equation (23) also approaches infinity. Hence:

$$\Phi\left(\left(k - \tfrac{1}{2}\right) \cdot \frac{\vartheta}{L}\right) \to \Phi(\infty) = 0 \tag{26}$$

For Equation (22), we can obtain:

$$\mathcal{H}_1 \to 0, \quad \mathcal{H}_2 \to 0, \quad \mathcal{H}_3 \to 0 \tag{27}$$

$$\mathcal{H}_{\text{PQA}} = 0 \tag{28}$$

As shown in Fig. 5(b), we consider the configuration $L = 1$ and $\vartheta = 20$ as an illustrative example, then evaluate the resulting entropy ratio $\mathcal{R}$ over $\alpha \in [-1, 0]$ and $\beta \in (0, 1]$. The results indicate that the PQA function suffers from near-complete information loss. Therefore, setting $\vartheta \gg L$ is also inadvisable for hyperparameter selection.

In summary, when the relationship between ϑ and L satisfies $\vartheta = L + \epsilon$ with $\epsilon \to 0$, one or more locally optimal parameter sets can be reliably identified. Conversely, when this condition is not met, no choice of parameter settings can compensate for the information loss of in the PQA function.

Conversion Error Analysis

When the initial membrane potential satisfies $\lim_{\epsilon \to 0} v^l(0) = \frac{1}{2}\vartheta^l_{\text{SNN}} + \epsilon$, and the SNNs are driven by a constant weighted input z^l at each timestep, the approximate form of the conversion error reduces to:

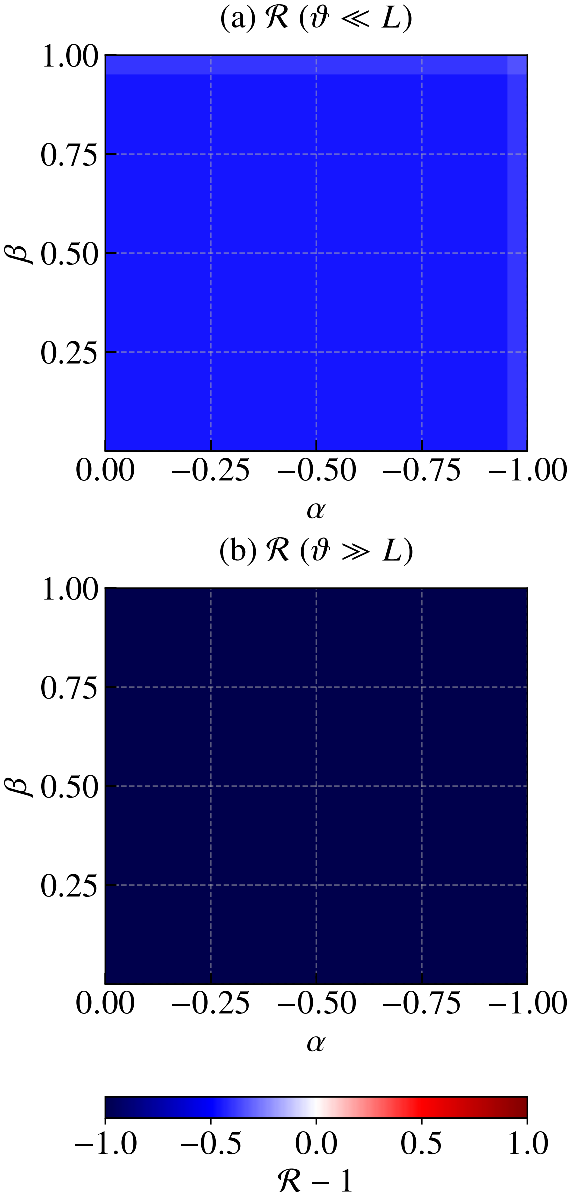

Figure 5: The grid search results of $\mathcal{R} = \frac{\mathcal{H}_{\text{PQA}}}{\mathcal{H}_{\text{BN}}}$ under two conditions: (a) $\vartheta \ll L$ and (b) $\vartheta \gg L$, with $\alpha \in [-1, 0]$ and $\beta \in (0, 1]$.

$$\widetilde{Err}^l = \frac{1}{T}\sum_{i=1}^{T}\left(\left\lfloor \frac{v^l(i-1) + z^l}{\vartheta^l_{\text{SNN}}} \right\rfloor\right) - \left\lfloor \frac{z^l}{\vartheta^l_{\text{SNN}}} \right\rceil \tag{29}$$

In the case where the timestep $T = 1$, the above expression simplifies to:

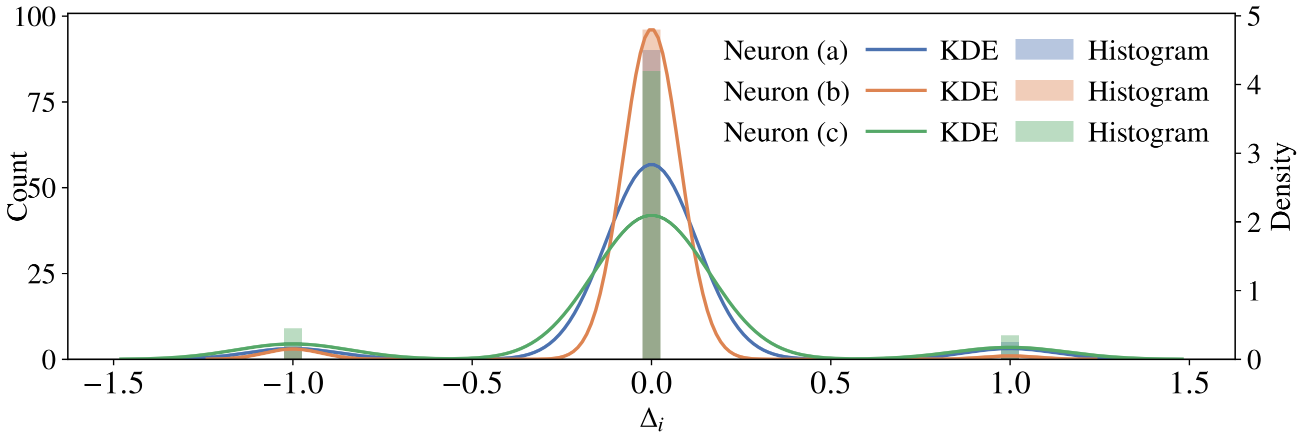

Figure 6: Histograms and kernel density estimates (KDE) of Δ_i across 100 timesteps from three randomly selected neurons from VGG-16.

$$
\begin{aligned}
\widetilde{\mathrm{Err}}^l\Big|_{T=1} &\approx \left\lfloor \frac{v^l(0)+z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor - \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor \\
&\approx \left\lfloor \frac{\frac{1}{2}\vartheta^l_{\mathrm{SNN}}+\epsilon+z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor - \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor \approx 0
\end{aligned} \tag{30}
$$

In the case where the timestep $T > 1$, based on the membrane potential update mechanism of the SNNs, we have:

$$v^l(i) = v^l(i-1) + z^l - \vartheta^l_{\mathrm{SNN}} \cdot s^l(i-1) \tag{31}$$

Here, $s^l(i-1) = \left\lfloor \frac{v^l(i-2)+z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor$ denotes the number of spikes emitted at the $(i-1)$-th timestep. Given that the initial membrane potential satisfies $v^l(0) = \frac{1}{2}\vartheta^l_{\mathrm{SNN}} + \epsilon$ and the input x^l remains constant, the soft-reset mechanism ensures that $v^l(i)$ remains bounded within the range $(-\vartheta^l_{\mathrm{SNN}}, \vartheta^l_{\mathrm{SNN}})$ for all i. The spike function can thus be further expressed as follows:

$$s^l(i) = \lfloor \frac{v^l(i-1)+z^l}{\vartheta_{\mathrm{SNN}}} \rfloor = \lfloor \frac{v^l(0)+z^l}{\vartheta_{\mathrm{SNN}}} + \frac{\delta_i}{\vartheta_{\mathrm{SNN}}} \rfloor \tag{32}$$

Let $\delta_i = v^l(i-1) - v^l(0)$ denote the perturbation introduced by the deviation of the membrane potential at timestep $i-1$ from its initial value. The error term can then be further approximated as:

$$
\begin{aligned}
&\sum_{i=1}^{T} \left\lfloor \frac{v^l(0)+z^l}{\vartheta_{\mathrm{SNN}}} + \frac{\delta_i}{\vartheta_{\mathrm{SNN}}} \right\rfloor \\
=&T \cdot \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} + \frac{1}{2} + o(\epsilon) \right\rfloor + \sum_{i=1}^{T} \Delta_i
\end{aligned} \tag{33}
$$

Where $\Delta_i = \left\lfloor \frac{v^l(0)+z^l}{\vartheta_{\mathrm{SNN}}} + \frac{\delta_i}{\vartheta_{\mathrm{SNN}}} \right\rfloor - \left\lfloor \frac{v^l(0)+z^l}{\vartheta_{\mathrm{SNN}}} \right\rfloor$, $\Delta_i \in \{-1, 0, 1\}$, and by substituting this into Equation (29), we obtain:

$$
\begin{aligned}
\widetilde{\mathrm{Err}}^l_{\mathrm{apx}}\Big|_{T>1} &\approx \frac{1}{T} \cdot T \left(\left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} + \frac{1}{2} + o(\epsilon) \right\rfloor \right) \\
&\quad + \frac{1}{T} \sum_{i=1}^{T} \Delta_i - \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor \\
&= \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} + o(\epsilon) \right\rfloor \\
&\quad - \left\lfloor \frac{z^l}{\vartheta^l_{\mathrm{SNN}}} \right\rfloor + \frac{1}{T} \sum_{i=1}^{T} \Delta_i \\
&= \frac{1}{T} \sum_{i=1}^{T} \Delta_i
\end{aligned} \tag{34}
$$

In conclusion, when the initial membrane potential is properly set and the input remains constant, SNNs can accurately reconstruct the activation values of ANNs within a single timestep, thus leading to the conversion error being effectively zero. When inference is extended to multiple timesteps, the conversion error is caused by the difference in spike count $\Delta_i = s^l(i) - s^l(1)$ at each timestep i, as illustrated in Fig. 6. Since $\Delta_i \in \{-1, 0, +1\}$ and $\mathbb{D} = \{\Delta_i \mid i \in [1, T]\}$, when $T \to \infty$, $E(\mathbb{R}) \to 0$ and $D(\mathbb{D}) \to \sigma^2$. Specifically, the average deviation between the expected and actual spike counts can be expressed as $\frac{1}{T}\sum_{i=1}^{T} \Delta_i$. As $T \to \infty$, $\sum_{i=1}^{T} \Delta_i$ approaches zero. The trend of $\sum_{i=1}^{T} \Delta_i$ suggests that as the number of timesteps increases, the resulting impact on accuracy during ANN-to-SNN conversion becomes negligible.

Supplemental Experimental Results and Training Details

Dataset Description. We conduct experiments on three widely used benchmarks: CIFAR-10, CIFAR-100, and ImageNet. CIFAR-10 comprises 60,000 32×32 color images across 10 classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images. CIFAR-100 has the same number of images but spans 100 classes, with 600 images per class. It follows the same 50,000/10,000 training/test split. ImageNet contains over 1.28 million images for training and 50,000 images in the validation set, with high-resolution natural images collected from the web.

Table 4: Training hyperparameters used for VGG-16 and ResNet-20 on CIFAR-10 and CIFAR-100.

Hyperparameter	**Value**
Batch Size	300
Total Train Epoch	600
Random Seed	42
Initial Learning Rate	1e-1
Weight Decay	5e-4
L	8
ANN Threshold ϑ	8
α	-0.25
β	1

Table 5: Training hyperparameters used for ViT-S experiments on CIFAR-10, CIFAR-100, and ImageNet.

Hyperparameter	**Value**
CIFAR-10/100 Train Epoch	300
CIFAR-10/100 Train Epoch	100
Warmup Epoch	5
Random Seed	0
Initial Learning Rate	1.5e-4
Weight Decay	0.05
Batch Size	64
Patch Size	16
MLP ratio	4
Rate of stochastic depth	0.1
Label Smoothing Factor	0.1
Cutmix	1.0
Mixup	$\alpha = 1$
Mixup probability	0.5
Random Erasing (RE) probability	0.25
RE Max erasing area	0.4
RE Aspect of erasing area	0.3
repeated augmentation(RA)	3
Data AutoAugment	True
LR Scheduler	Cosine LR
augmentation policies for training	False
L	16
α	$-1/2$
β	$7/16$
Locality Self-Attention	False
Shifted Patch Tokenization	False

Training Hyperparameters. To ensure a fair comparison, we evaluate the proposed conversion method on three datasets mentioned above using three architectures: VGG-16, ResNet-20, and ViT-S. These models were selected to represent a diverse range of network structures, including convolutional networks and transformer-based architectures, thereby demonstrating the generalizability of the proposed approach across different model types.

The training hyperparameters for VGG-16, ResNet-20, and ViT-S are provided in Table 4. For ViT-S, which does not include batch normalization (BN) layers, we follow the training settings of SpikeZIP-TF and do not apply the hyperparameter search strategy we proposed. Detailed settings for ViT-S are listed in Table 5.