

Generalized Threshold Optimization with Harmony Multi-Threshold Neurons for Accurate ANN-to-SNN Conversion

Wenhan Zhang¹, Zihan Huang¹, Tong Bu^{1,2}, Tiejun Huang¹, Zhaofei Yu^{1,2*}

¹School of Computer Science, Peking University

²Institute for Artificial Intelligence, Peking University

faizwh@stu.pku.edu.cn, hzh@stu.pku.edu.cn, putong30@pku.edu.cn, yuzf12@pku.edu.cn, tjhuang@pku.edu.cn

Abstract

Spiking Neural Networks (SNNs) are a promising paradigm designed to emulate the brain's energy efficient by incorporating the timing of spikes. Conversion is an efficient way to obtain high-performance SNNs from Artificial Neural Networks (ANNs). Existing conversion methods often face a trade-off between accuracy and time steps, which is largely caused by the incomplete release of residual membrane potentials. To minimize the conversion error, this paper proposed a harmonious mathematical property-based neuron, called Harmony Multi-Threshold Neurons (H-MT Neuron), which utilizes multiple spikes to minimize residual membrane potentials. The proposed neuron is further enhanced with an optional effective communication mechanism to achieve more accurate conversion. In addition, we propose a threshold optimization method applicable to a broader range cases of spiking neurons to find the optimal neuron thresholds. Experiment results demonstrate that our method achieve superior accuracy on ImageNet benchmark datasets while significantly reducing the required time steps and energy consumption.

Code and Technical Appendix —

<https://github.com/faizwh/H-MT.git>

1 Introduction

Spiking Neural Networks (SNNs), regarded as the third generation of neural network models (Maass 1997), are distinguished by their biological plausibility and dynamic neuron behaviors (Gerstner et al. 2014), exhibiting significant potential to rival Artificial Neural Networks (ANNs). The two architectures differ in computational mechanisms: ANNs rely on continuous floating-point value transmission, whereas SNNs propagate sparse, binary spikes across layers. Such spike-based communication better emulates biological neuron dynamics and enhances energy efficiency (Merolla et al. 2014; Davies et al. 2018; DeBole et al. 2019; Pei et al. 2019).

However, the discrete nature of spikes introduces non-differentiability, posing challenges for training. Recent advancements in direct training have made progress in addressing this challenge via surrogate gradient methods (Neftci,

Mostafa, and Zenke 2019). These methods approximate gradients of non-differentiable spike processes using smooth functions, enabling practical backward propagation and gradient descent, thereby improving SNN performance (Fang et al. 2021; Duan et al. 2022; Shi, Hao, and Yu 2024; Ding et al. 2025). Nevertheless, surrogate gradients remain approximations and may mislead the training (Gygax and Zenke 2025). Moreover, the temporal nature of SNNs leads to high computational costs due to the need for backpropagation through time (BPTT). Although several online training methods have been proposed to estimate gradients more efficiently (Xiao et al. 2022; Bohnstingl et al. 2022; Meng et al. 2023; Zhu et al. 2024), they often result in sub-optimal accuracy. Consequently, direct training of SNNs remains a persistent challenge.

Meanwhile, another prominent spiking deep learning methodology is ANN-to-SNN conversion (Cao, Chen, and Khosla 2015; Han, Srinivasan, and Roy 2020; Li et al. 2021; Deng and Gu 2021; Bu et al. 2022a; Bu, Li, and Yu 2025; Zhao et al. 2025; Ding et al. 2021). This method converts pre-trained ANNs to SNNs by replacing nonlinear units with spiking neurons, such as traditional Integrate-and-Fire (IF) neuron with single threshold or its multi-threshold variants designed for higher precision within shorter inference time-steps. Although ANN-to-SNN conversion avoids complexities of direct training and produces high-performance SNNs with accuracy comparable to original ANNs, it typically requires more time steps for precision, resulting in higher latency and energy consumption compared to surrogate gradient based direct training.

In this article, we identify key inherent defect in many designs of multi-threshold neuron, due to which the output of these neurons can't be well-represented by specific mathematical expressions as IF neuron. We then further propose a novel neuron structure called H-MT neuron, with harmonious mathematical properties and without such defect, designed to minimize residual membrane potential and thereby reduce conversion error. The proposed neuron integrate two symmetric components to approximate the linear function $y = x$, along with a communication mechanism that helps alleviate unevenness errors (Bu et al. 2022b). By simulating $y = x$, we can keep everything in the form of spike, thus maintaining the energy efficiency of SNNs. Furthermore, we generalize the threshold optimization method from Huang

*Corresponding author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

et al. to obtain the optimal threshold, broadening its applicability to a wider variety multi-threshold neuron architectures for simulating either ReLU or $y = x$ function.

Our main contributions are summarized as follows:

- We identify the key inherent defect in many designs of multi-threshold neuron, and further propose a novel multi-threshold neuron with nice mathematical properties and without the defect, capable of simulating both ReLU and $y = x$ functions. The design is further enhanced with a simple yet effective communication mechanism to reduce unevenness errors.
- We propose a general threshold iteration optimization method to determine the optimal threshold, enabling its application across a wide range of spiking neuron structures. This approach facilitates accurate threshold selection for converting both ReLU and $y = x$ functions.
- We evaluate our method on the ImageNet dataset. Compared with previous CNN-SNN conversion methods, especially with those solely based on modification of neurons, the proposed method achieves higher accuracy and decreased energy consumption.

2 Related Works

2.1 Neuron Design in SNNs

Spiking neurons serve as the fundamental computational units in SNNs. Researchers focus on both direct training and ANN-to-SNN conversion consistently investigate existing spiking neuron models and propose novel neuron structures to improve networks' performance and efficiency. Bu et al. conducted an in-depth analysis of the classical IF model as well as conversion errors in ANN-to-SNN, and put forward corresponding adjustment on neuron settings, such as initial membrane potential, inspiring following studies. Some previous works (Lv et al. 2024; Hao et al. 2024; Wang and Zhang 2023; Luo et al. 2024; Huang et al. 2024, 2025; Wang et al. 2025) modified traditional single-threshold IF neuron, for the sake of higher precision, to n -threshold neurons through essentially almost identical designs, which partition single threshold into n equal segments. With 0 taken into account, such neurons can output $n + 1$ distinct values per time-step.

Meanwhile, other studies (Li, Zhao, and Zeng 2022; Kim et al. 2018) introduced periodically varying thresholds during inference, but can cause unexpectedly long inference time for high accuracy. Moreover, Notably, distinct from the aforementioned efforts aiming at simulating ReLU or $y = x$, researchers (Jiang et al. 2024) proposed heuristic designs for composite neurons composite of smaller learnable-threshold sub-neurons, to emulate complex nonlinear modules widely used in Transformer architectures, e.g., GELU, e^x , Layer-Norm and so on.

2.2 Neuron Threshold Policy for ANN-to-SNN conversion

Beyond neuron structure modifications, determining spike neuron thresholds is critical. Learning thresholds as trainable parameters is one represent that is popular in direct

training. As for ANN-to-SNN, thresholds are traditionally set to the 99.9% quantile of the activation value of the replaced ANN nonlinear units, as recommended by Rueckauer et al. which does show some excellence in practice despite being empirical. Researchers have also been continuously exploring threshold optimization policies grounded on more reliable methods and concrete theories. For instance, Li et al. attempted to find better thresholds via grid search, while Bu, Li, and Yu introduced a local threshold balancing algorithm that efficiently finds the optimal thresholds through training and performs fine-grained adjustment of the threshold value. Additionally, Huang et al. developed a threshold iteration optimization method to find the threshold minimizing the conversion error of IF neuron simulating ReLU, under the assumption that inputs obey normal distribution. However, the theorems have some limitations, such that they can only support a rough optimization on the multi-threshold neurons employed in this work. In this article, we extensively generalize these theorems by Huang et al. with rigorous proofs, extending their applicability to a broader range of spiking neuron designs for simulating either ReLU or $y = x$.

3 Preliminaries

3.1 The Integrate-and-Fire (IF) Neuron and Designs of Multi-Threshold Spiking Neuron

Similar to many previous works. We begin by focusing on the Integrate-and-Fire (IF) Neuron, which is a classical spiking neuron model with one threshold and is used to replace ReLU neuron in ANN-to-SNN conversion. We use the "reset-by-subtraction" mechanism, and its dynamics can be expressed as follows:

$$\mathbf{m}^l[t] = \mathbf{v}^l[t-1] + \mathbf{I}^l[t] = \mathbf{v}^l[t-1] + f(\mathbf{x}^{l-1}[t]), \quad (1)$$

$$\mathbf{s}^l[t] = H(\mathbf{m}^l[t] - \boldsymbol{\theta}^l), \quad (2)$$

$$\mathbf{v}^l[t] = \mathbf{m}^l[t] - \mathbf{s}^l[t]\boldsymbol{\theta}^l, \quad (3)$$

$$\mathbf{x}^l[t] = \mathbf{s}^l[t]\boldsymbol{\theta}^l, \quad (4)$$

where $\mathbf{x}^{l-1}[t]$ is the initial input received by the l -th layer as well as the output of the $l-1$ -th layer at time-step t , $\mathbf{I}^l[t]$ is the input ultimately fed into the neuron in the by l -th layer after some operation f , which can be affine transformation, while $\mathbf{m}^l[t]$ and $\mathbf{v}^l[t]$ are the membrane potential before and after firing at time-step t . Here $\boldsymbol{\theta}^l$ is the threshold of the l -th layer and $H(\cdot)$ is the Heaviside step function, therefore the element of $\mathbf{s}^l[t]$, which is the spike vector of l -th layer, is either 1 if there is a spike or 0.

When using $\phi^l(T) = \frac{\sum_{t=1}^T \mathbf{x}^l[t]}{T}$ as the average output of the l -th layer and $\mathbf{z}^l = \frac{\sum_{t=1}^T \mathbf{I}^l[t]}{T}$ as the average input fed into the spiking neuron at the l -th layer, we will have:

$$\phi^l(T) = \boldsymbol{\theta}^l \frac{\sum_{t=1}^T \mathbf{s}^l[t]}{T} = \mathbf{z}^l - \frac{\mathbf{v}^l[T] - \mathbf{v}^l[0]}{T}. \quad (5)$$

To estimate $\phi^l(T)$, Bu et al. put forward a strong assumption as shown in Equation (6) (here $\mathbf{u}^l[t]$ is identical to $\mathbf{I}^l[t]$).

$$\begin{cases} \begin{cases} \mathbf{u}^l[t] \leq 0, \forall t = 1, \dots, T & \text{if } \mathbf{z}^l \leq 0 \\ \mathbf{u}^l[t] \in (0, \boldsymbol{\theta}^l), \forall t = 1, \dots, T & \text{if } \mathbf{z}^l \in (0, \boldsymbol{\theta}^l) \\ \mathbf{u}^l[t] \geq \boldsymbol{\theta}^l, \forall t = 1, \dots, T & \text{if } \mathbf{z}^l \geq \boldsymbol{\theta}^l \end{cases} \\ \mathbf{v}^l[0] \in [0, \boldsymbol{\theta}^l] \end{cases} \quad (6)$$

With this assumption, the range of $v^l[t]$ can be limited to $[0, \theta^l)$ for $\forall t = 1, \dots, T$, if $z^l \in (0, \theta^l)$, and a precise mathematical expression of the IF neuron's output can be derived to help the ANN-to-SNN conversion:

$$\phi^l(T) = \frac{\theta}{T} \text{clip} \left(\lfloor \frac{z^l T + v^l[0]}{\theta} \rfloor, 0, T \right). \quad (7)$$

To enhance computational accuracy, many previous works have made effort to design novel structures of Multi-Threshold Spiking Neuron to replace ReLU (Lv et al. 2024; Wang and Zhang 2023; Hao et al. 2024) as well as $y = x$ (Huang et al. 2024, 2025), and have achieved some wonderful performance in comparison to the IF neuron.

Assuming there are n channels with different thresholds to fire spikes inside a neuron, just as IF neuron, these neuron can also derive an equation similar to Equation (5):

$$T\phi^l(T) = \sum_{t=1}^T \sum_{i=1}^n \lambda_i^l s_i^l[t] = z^l T + v^l[0] - v^l[T]. \quad (8)$$

However, due to intrinsic defects of these neuron's structures, it's hard either to extend the strong assumption from (Bu et al. 2022b) or further derive similar mathematical expressions for these neurons, bringing trouble for more precise optimization. In short, the main inherent defect of these neuron designs is that they can't properly limit the range of neuron's membrane potential after firing spikes at per time-step as IF neuron does, even when under some strong assumption. A detailed analysis can be found at Appendix A.

3.2 Optimization Method for Threshold of Spiking Neuron

Besides the learning method in direct training as well as grid search (Li et al. 2021) and the local threshold balancing algorithm (Bu, Li, and Yu 2025) in ANN-to-SNN conversion, Huang et al. present a novel optimization method for the threshold of the IF neuron to lower the conversion error of simulating ReLU. Assume the input x fed into the neuron (here the x corresponds to z^l mentioned ahead) obey a normal distribution with a mean of μ and a variance of σ^2 (de G. Matthews et al. 2018). The error of conversion from ReLU to IF neuron can be expressed as a function of the threshold θ :

$$QE(\theta) = \int_{-\infty}^{\infty} (f(x, \theta) - \max(x, 0))^2 e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \quad (9)$$

$$f(x, \theta) = \frac{\theta}{N} \text{clip} \left(\lfloor \frac{Nx + \theta}{\theta} \rfloor, 0, N \right). \quad (10)$$

Then it can be proved that through the iteration $\theta_{new} = k_{\mu, \sigma, n}(\theta) \times \theta_{ori}$ (here θ_{new} and θ_{ori} means the threshold after and before being updated), the threshold will converge to the unique optimal solution of the error $QE(\theta)$. $k_{\mu, \sigma, n}(\theta)$ is as follows:

$$k_{\mu, \sigma, n}(\theta) = \frac{\mu}{\theta} \frac{1 - \sum_{i=1}^n \frac{1}{n} \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right)}{1 - \sum_{i=1}^n \frac{2i-1}{n^2} \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right)} + \frac{\sigma}{\sqrt{2}\theta} \frac{1 - \sum_{i=1}^n \frac{1}{n} e^{-\frac{(\frac{2i-1}{2n}\theta - \mu)^2}{2\sigma^2}}}{1 - \sum_{i=1}^n \frac{2i-1}{n^2} \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right)}. \quad (11)$$

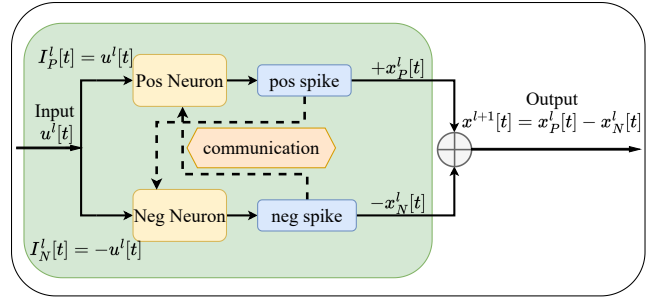


Figure 1: The structure of Harmony Multi-Threshold Spiking Neuron

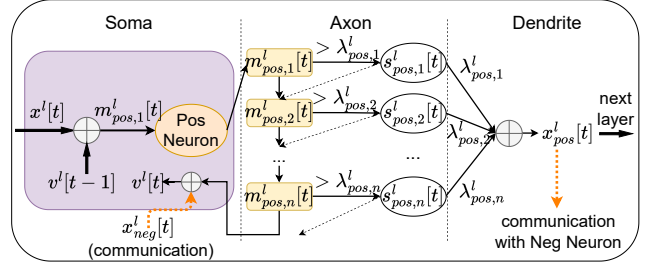


Figure 2: How spikes are emitted inside the Pos Neuron of H-MT. The black dot lines here mean if a spike is fired, the membrane potential should minus the corresponding threshold. Since the communication mechanism is optional, we mark with tangerine dashed lines.

4 Method

In this section, we propose a novel structure of multi-threshold neuron with harmonious mathematical properties as well as a generalized threshold optimization method, to achieve conversion of high accuracy with low latency.

4.1 Harmony Multi-Threshold Spiking Neuron

To better simulate $y = x$ (and ReLU) and to further optimize neuron's performance, it's necessary to have a good design of multi-threshold neuron able to overcome the defects mentioned in Section 3.1 and Appendix A. To properly limit the range of its membrane potential under some strong assumption, the neuron should have a better mechanism for emitting spikes as well as a proper structure. Here we present a new design called **Harmony Multi-Threshold Spiking Neuron** (in short **H-MT**), as it enjoys some good properties and is harmonious.

As shown in Figure 1, inspired by (Jiang et al. 2024), H-MT is composed of two symmetric neurons, respectively called **Pos neuron** and **Neg neuron**. Both are based on the same structure **Basal Neuron** which can simulate ReLU, but receive opposite inputs. When received a transformed input $u^l[t]$, **H-MT** will feed $I_P^l[t] = u^l[t]$ to the Pos Neuron and feed $I_N^l[t] = -u^l[t]$ to the Neg Neuron. The spikes emitted by the two neuron are called **pos spikes** (defined as $x_P^l[t]$) and **neg spike** (defined as $x_N^l[t]$), and the output of H-MT is $x^l[t] = x_P^l[t] - x_N^l[t]$.

To be precise, the basic dynamics of Basal Neuron are as Equation (12) to (18). You may also referred to Figure 2, which describes how spikes are emitted inside the Pos Neuron of H-MT (since the Pos and Neg Neuron are symmetric).

$$\text{Thresholds : } \lambda_1^l = \theta^l, \lambda_2^l = \frac{\theta^l}{2}, \dots, \lambda_n^l = \frac{\theta^l}{2^{n-1}} = \lambda_{min}^l, \quad (12)$$

$$v^l[0] = \frac{\lambda_{min}^l}{2}, \quad (13)$$

$$m_1^l[t] = v^l[t-1] + I^l[t], \quad (14)$$

$$s_i^l[t] = \begin{cases} 1, & \text{if } m_i^l[t] \geq \lambda_i^l \\ 0, & \text{else} \end{cases}, \quad (15)$$

$$m_{i+1}^l[t] = m_i^l[t] - \lambda_i^l s_i^l[t], \quad i = 1, 2, \dots, n-1, \quad (16)$$

$$x^l[t] = \sum_{i=1}^n \lambda_i^l s_i^l[t], \quad (17)$$

$$\begin{aligned} v^l[t] &= m_n^l[t] - \lambda_n^l s_n^l[t] = m_1^l[t] - x^l[t] \\ &= v^l[t-1] + I^l[t] - x^l[t]. \end{aligned} \quad (18)$$

At each time-step, there are 2^n possible output values for the H-MT neuron, from $0, \lambda_{min}^l$ to $(2^n - 1)\lambda_{min}^l$. Therefore, we extend the strong assumption from (Bu et al. 2022b) as follows:

$$\forall t = 1, \dots, T,$$

$$\begin{cases} u^l[t] \leq -(2^n - 1)\lambda_{min}^l, & \text{if } z^l \leq -(2^n - 1)\lambda_{min}^l \\ u^l[t] \in (-(2^n - 1)\lambda_{min}^l, 0), & \text{if } z^l \in (-(2^n - 1)\lambda_{min}^l, 0) \\ u^l[t] \in [0, (2^n - 1)\lambda_{min}^l), & \text{if } z^l \in [0, (2^n - 1)\lambda_{min}^l) \\ u^l[t] \geq (2^n - 1)\lambda_{min}^l, & \text{if } z^l \geq (2^n - 1)\lambda_{min}^l \end{cases}. \quad (19)$$

Let $\phi_{P/N}^l(T) = \frac{\sum_{t=1}^T x_{P/N}^l[t]}{T}$. Similar to Equation (7), under this strong assumption, we can further derive the mathematical expression for H-MT's output, as given in Equation (20). This derivation is feasible because the Basal Neuron of H-MT is capable of restricting its membrane potential within either $[0, \lambda_{min}^l)$, depending on the input.

$$\begin{aligned} \phi^l(T) &= \phi_P^l(T) - \phi_N^l(T) \\ &= \frac{\lambda_{min}^l}{T} \text{clip}(\lfloor \frac{z^l T + \frac{\lambda_{min}^l}{2}}{\lambda_{min}^l} \rfloor, -(2^n - 1)T, (2^n - 1)T). \end{aligned} \quad (20)$$

Meanwhile, we are faced with another challenge called Unevenness Error, which is first proposed by Bu et al. and calls for more serious consideration. It describes the conversion error caused by the unevenness of input. A classical example of the unevenness error can be found in Appendix B for a better understanding. It can be even more complicated when it comes to multi-threshold spiking neurons, as the input and output become more complex, making it difficult to define "evenness" at least.

We notice that such errors are largely caused by specific inputs with extreme magnitude and the clustering of same-signed inputs, which can lead to extreme membrane potentials and further result in extra or missing spikes. To address this issue, we introduce a simple yet powerful **communication mechanism** to reduce unevenness error by preventing extreme membrane potentials. This mechanism, as defined in Equation (21), operates on the membrane potentials of the Pos Neuron and Neg Neuron ($v_P^l[t]$ and $v_N^l[t]$) after spike

firing. This mechanism is executed after the neuron dynamic computation at each time-step.

$$v_{P/N}^l[t] \leftarrow v_{P/N}^l[t] + x_{N/P}^l[t], \quad (21)$$

It can be easily proven that under the strong assumption Equation (19), H-MT with communication mechanism can still keep the mathematical expression Equation (20). Moreover, at each time-step, H-MT with communication mechanism will prevent the membrane potential of the two parts from being too extreme, and as following Lemma 4.1 shows:

Lemma 4.1. *H-MT with communication mechanism will have only one of its Pos and Neg neuron to fire spikes at every time-step, as for $\forall t = 1, \dots, T$, we have:*

$$v_P^l[t] + v_N^l[t] = \lambda_{min}^l. \quad (22)$$

Detailed analysis of H-MT's properties can be found at Appendix C. And the test cases demonstrating how communication mechanism suppresses unevenness error can be found at Appendix D.

4.2 Extension of the Optimization Method for Threshold

Since the mathematical expression of H-MT (Equation (20)) differs from that of IF model (Equation (7)), the optimization method of threshold from (Huang et al. 2025) doesn't apply. Therefore, we generalize the Huang et al.'s theorem and make it applicable for more multi-threshold neuron with proper structure that can derive similar mathematical expression.

Assumption 4.2. According to the (de G. Matthews et al. 2018), assume that the input x fed into the neuron (here the x corresponds to z^l mentioned ahead) follows a normal distribution with mean μ and variance σ^2 .

By extending the error of conversion from ReLU to $y = x$, the following definitions and lemmas can be proposed to find the best threshold in such condition.

Definition 4.3. The quantization and clipping errors introduced when approximating the $y = x$ function can be formulated as:

$$QE(\theta, a, b) = \int_{-\infty}^{\infty} (f(x, \theta, a, b) - x)^2 e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \quad (23)$$

$$f(x, \theta, a, b) = \frac{\theta}{n} \text{clip}(\lfloor \frac{nx + \frac{\theta}{2}}{\theta} \rfloor, a, b), a, b \in \mathbb{Z}, a < b, \quad (24)$$

where a, b are arbitrary integers not necessarily related to n , and that the \mathbb{Z} here is the set of integer.

We can extend the error function QE to two auxiliary functions QE_1 and QE_2 , which facilitate the calculation of the optimal threshold. These functions are defined as follows:

$$QE_1(\theta, k, a, b) = \int_{-\infty}^{\infty} (f_1(x, \theta, k, a, b) - x)^2 e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \quad (25)$$

$$f_1(x, \theta, k, a, b) = k \frac{\theta}{n} \text{clip}(\lfloor \frac{nx + \frac{\theta}{2}}{\theta} \rfloor, a, b), a, b \in \mathbb{Z}, a < b, \quad (26)$$

$$QE_2(\theta, k, a, b) = \int_{-\infty}^{\infty} (f_2(x, \theta, k, a, b) - x)^2 e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx, \quad (27)$$

$$f_2(x, \theta, k, a, b) = \frac{\theta}{n} \text{clip}(\lfloor \frac{nx + \frac{k\theta}{2}}{k\theta} \rfloor, a, b), a, b \in \mathbb{Z}, a < b. \quad (28)$$

Then we have following lemmas.

Algorithm 1: Threshold iteration method to find the best threshold for conversion from ReLU/ $y = x$ to spiking neuron

- 1: **Input:** Pre-trained ANN Model F_{ANN} , Dataset \mathcal{D} .
 - 2: **Initialize:** Set $\theta \leftarrow 1$ (any positive initial value)
 - 3: Run the model F_{ANN} on dataset \mathcal{D} to statically compute the mean μ and variance σ^2 of pre-activations of each ReLU / $y = x$ separately.
 - 4: **repeat**
 - 5: Compute k_1 for current θ based on μ and σ^2 according to Eq (29)
 - 6: Update $\theta \leftarrow k_1 \cdot \theta$
 - 7: **until** $1 - \epsilon < k_1 < 1 + \epsilon$, where ϵ tends to 0.
 - 8: **Output:** Threshold θ
-

Lemma 4.4. For any fixed integer a, b, n , any fixed $\theta > 0, \mu, \sigma > 0$, $QE_1(\theta, a, b)$ reaches the minimal value only when:

$$k = k_{\mu, \sigma, n, a, b}(\theta) = \frac{\mu}{\theta} \frac{n \left[a + b - \sum_{i=a+1}^b \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right) \right]}{a^2 + b^2 - \sum_{i=a+1}^b (2i-1) \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right)} + \frac{\sigma}{\sqrt{\frac{\pi}{2}}\theta} \frac{n \sum_{i=a+1}^b e^{-\frac{(\frac{2i-1}{2n}\theta - \mu)^2}{2\sigma^2}}}{a^2 + b^2 - \sum_{i=a+1}^b (2i-1) \text{erf} \left(\frac{\frac{2i-1}{2n}\theta - \mu}{\sqrt{2}\sigma} \right)}. \quad (29)$$

Lemma 4.5. When $a \leq 0 \leq b$, $a < b$, for any $\theta > 0, \mu \in \mathbb{R}, \sigma > 0$, we will always have: $k_{\mu, \sigma, n, a, b}(\theta) > 0$.

Lemma 4.6. For any fixed integer a, b, n , any fixed $\theta > 0, \mu, \sigma > 0$, for all $k > 0$, $QE_2(\theta, a, b)$ reaches the minimal value only when: $k = 1$.

Then for integers $a \leq 0 \leq b$ with $a < b$, the following inequality naturally holds:

$$QE(k_1\theta, a, b) \leq QE_2(k_1\theta, \frac{1}{k_1}, a, b) = QE_1(\theta, k_1, a, b) \leq QE(\theta, a, b). \quad (30)$$

where k_1 denotes $k_{\mu, \sigma, a, b}(\theta)$ for simplicity. Moreover, the equality holds if and only if $k_1 = 1$.

Lemma 4.7. When $a \leq 0 \leq b$, $a < b$, there exists a unique $\theta_0 > 0$ such that $k_1 = 1$. When $0 < \theta < \theta_0$, $k_1 > 1$. When $\theta > \theta_0$, $0 < k_1 < 1$. We can further derive that for fixed a, b $QE(\theta, a, b)$ reaches the minimal when $\theta = \theta_0$.

It's worth noticing that since $a \leq 0 \leq b$, $a < b$, and since a, b are arbitrary integers not necessarily related to n , all the generalized theorem above can be applied not only to optimization of the threshold of both H-MT and classical IF model, but also to the conversion from both $y = x$ and ReLU to spiking neuron.

Therefore, we can still optimize the threshold θ through iteration, as shown in Algorithm 1. The detailed proof of the theorems in this section can be found in Appendix E.

5 Experimental Results

In this section, we first evaluate the performance of our proposed method on ImageNet dataset (Deng et al. 2009) across

different models, including ResNet18, ResNet34, and VGG-16bn, comparing our results with some previous ANN-to-SNN conversion methods. Then, we compute and analyze the energy consumption of the converted SNNs. Finally, we conduct comparative experiments to explore the influence of the number of thresholds in H-MT, check H-MT's performance of simulating identity mapping, and validate the effectiveness of generalized threshold optimization method as well as the communication mechanism.

5.1 Comparison with Previous ANN-to-SNN Conversion Methods

As shown in Table 1, we conducted CNN-to-SNN conversion with our H-MT on ImageNet (Deng et al. 2009) across ResNet18, ResNet34 and VGG-16bn, and compare the results with those of previous works. We denote the converted model as $model - n/M$, meaning that the multi-threshold neuron have n threshold and its threshold are chosen through method M . Here method **Opt** means our generalized threshold optimization method while method **999** means the traditional method of choosing the 99.9% quantile of ANN's activation value (denoted as $Ac_{99.9}$) as the threshold. To be precise, when using the **999** method, we normalize the minimum threshold of H-MT as $\lambda_{min} = Ac_{99.9}/(2^n - 1)$. The conversions are conducted basically through replacing ReLU with the Pos Neuron part of H-MT, since there are not many identity mappings in the 3 models involved. But we conduct experiments as well of H-MT substituting $y = x$ to check the neuron's performance with 16 inference time-steps, corresponding to $16^{y=x}$ in Table 1. H-MT here is equipped with communication mechanism.

From the results of experiments conducted with traditional threshold method, we can see that (Pos Neuron of H-MT is of high precision due to its nicely designed structure with good mathematical property. Performance further improves with our generalized optimization method, highlighting the value of this approach. Moreover, the excellence of our method isn't fully demonstrated, even if DCGS from (Huang et al. 2025) outperforms our method, as they introduce a novel differential coding method with better precision to replace traditional rate coding, while we focus on modification of spiking neuron's structure. Besides, concrete data of experiments conducted on ImageNet (Deng et al. 2009) from other previous ANN-to-SNN conversion works based on design of spiking neuron structure is lacked for meaningful comparison.

5.2 Energy Estimation and Result Analysis

According to (Horowitz 2014), the energy consumption ratio of the converted SNN relative to the ANN can be expressed by the following Equation,

$$\frac{E_{\text{SNN}}}{E_{\text{ANN}}} = \frac{MAC_{\text{SNN}} \times E_{\text{MAC}} + AC_{\text{SNN}} \times E_{\text{AC}}}{MAC_{\text{ANN}} \times E_{\text{MAC}}}, \quad (31)$$

where $E_{\text{MAC}} = 4.6\text{pJ}$ and $E_{\text{AC}} = 0.9\text{pJ}$.

Since SNNs has almost no multiplication operations, we can regard $MAC_{\text{SNN}} \gg MAC_{\text{ANN}}$. Thus, we only need

Method	Type	Arch.	Param.(M)	ANN Acc(%)	T	SNN Acc(%)
TS (Deng and Gu 2021)	CNN-to-SNN	VGG-16	138	72.40	64	70.97
SNM (Wang et al. 2022)	CNN-to-SNN	VGG-16	138	73.18	64	71.50
MMSE (Li et al. 2021)	CNN-to-SNN	ResNet-34	21.8	75.66	64	71.12
		VGG-16	138	75.36	64	70.69
QCFS (Bu et al. 2022b)	CNN-to-SNN	ResNet-34	21.8	74.32	64	72.35
		VGG-16	138	74.29	64	72.85
SRP (Hao et al. 2023)	CNN-to-SNN	ResNet-34	21.8	74.32	4, 64	66.71, 68.61
		VGG-16	138	74.29	4, 64	66.47, 69.43
GN (Lv et al. 2024)	CNN-to-SNN	ResNet-34	21.8	74.35	8, 32	73.57, 73.46
BSNN (Li, Zhao, and Zeng 2022)	CNN-to-SNN	ResNet-34	21.8	73.27	989	72.64
DCGS (Huang et al. 2025)	CNN-to-SNN	ResNet18	11.7	71.49	4, 8	70.07, 71.31
		ResNet34	21.8	76.42	4, 8	73.35, 76.04
		VGG16bn	138	73.25	4, 8	72.72, 73.17
H-MT (Ours)	CNN-to-SNN	ResNet18 -4/999	11.7	71.50	4, 16, $16^{y=x}$	67.66, 68.48, 66.78
		ResNet34 -4/999	21.8	76.42	4, 16, $16^{y=x}$	71.91, 72.85, 71.67
		VGG-16bn -4/999	138	73.25	4, 16, $16^{y=x}$	72.19, 72.15, 71.78
		ResNet18 -4/Opt	11.7	71.50	4, 16, $16^{y=x}$	69.02, 70.95, 67.61
		ResNet34 -4/Opt	21.8	76.42	4, 16, $16^{y=x}$	73.25, 75.80, 74.86
		VGG-16bn -4/Opt	138	73.25	4, 16, $16^{y=x}$	71.94, 72.92, 72.71
		ResNet18 -8/Opt	11.7	71.50	2, 4	71.39, 71.07
		ResNet34 -8/Opt	21.8	76.42	2, 4	76.276, 76.168
VGG-16bn -8/Opt	138	73.25	2, 4	73.242, 72.452		

Table 1: Comparison between the proposed method and previous ANN-to-SNN conversion works on ImageNet dataset.

Model Config	Acc/Energy	Time-step T	
		2	4
ResNet34 - 8/Opt, Param:21.8M, Acc:76.42%	Acc	76.276	76.168
	Energy ratio	0.2986	0.4703
ResNet34 - 8/999, Param:21.8M, Acc:76.42%	Acc	72.908	72.908
	Energy ratio	0.5435	1.0861
ResNet34 - 8/999 ₂ , Param:21.8M, Acc:76.42%	Acc	75.702	75.684
	Energy ratio	0.4622	0.9228

Table 2: Accuracy and energy ratio of H-MT(Ours) with $n = 8$ thresholds, of ResNet34 on ImageNet Dataset. Thresholds here are chosen through the generalized threshold optimization method, traditional policy and empirical modification of the traditional policy.

to estimate $\frac{AC_{S_{SNN}}}{MAC_{S_{SNN}}}$, which in H-MT Neuron can be calculated by

$$\frac{AC_{S_{SNN}}}{MAC_{S_{SNN}}} = \frac{\sum_{l=1}^L \sum_{i=1}^N \sum_{j=1}^n (s_{i,P,j}^l[t] + s_{i,N,j}^l[t]) c_i^l}{\sum_{l=1}^L \sum_{i=1}^N c_i^l}, \quad (32)$$

where c_i^l represents the number of outgoing connections from the i -th neuron in layer l to the neurons in the next layer. Table 2 presents partial results of energy consumption

ratio, and the detailed supplement results can be found in Appendix F.

As shown in Table 2, with $n = 8$ thresholds on ResNet34, the generalized optimization method (Opt) achieve less than 0.15% accuracy loss with low energy ratio within very low latency ($T = 2$), while the traditional policy (999) causes higher energy consumption and still remarkable accuracy loss. We assume this may be because that the threshold chosen by traditional policy isn't large enough, thus firing spikes that are of large number but insufficient to generate desired output value.

Therefore, we attempt another empirical modification of the traditional policy, denoted as 999₂, i.e. using twice the threshold obtained by traditional methods as the actual threshold. As the experiment results show, method 999 achieves higher accuracy and a little lower energy ratio in comparison with traditional policy, but clearly doesn't outperform the generalized optimization method on these two metrics.

5.3 Influence of the Number of Thresholds

As Table 3 shows, the effect of number of threshold is remarkable. There is evident accuracy gap between H-MT with $n = 3$ and $n = 4$ thresholds, which is understandable, since for $n = 3$ and $n = 4$, there are respectively 2^3

Model Config\Acc(%)	Time-step T			
	2	4	8	16
ResNet34 - 4/Opt, Param:21.8M, Acc:76.42%	71.73	73.25	75.02	75.80
ResNet18 - 4/Opt, Param:11.7M, Acc:71.50%	67.78	69.02	70.35	70.95
VGG16bn - 4/Opt, Param:138M, Acc:73.25%	70.78	71.94	73.03	72.92
ResNet34 - 3/Opt, Param:21.8M, Acc:76.42%	64.33	68.33	71.6	73.91
ResNet18 - 3/Opt, Param:11.7M, Acc:71.50%	61.33	65.38	67.95	69.43
VGG16bn - 3/Opt, Param:138M, Acc:73.25%	67.97	70.37	71.65	72.62

Table 3: Accuracy of H-MT(Ours) with $n = 4$ thresholds, and Accuracy of H-MT(Ours) with $n = 3$ thresholds, of different converted models on ImageNet Dataset. Thresholds here are chosen through the generalized threshold optimization method

and 2^4 possible output values at per time-step. It’s also interesting that there is still notable accuracy gap between $n = 3$ with $T = 16$ and $n = 4$ with $T = 8$. Anyway, such results again emphasizes the high demands of ANN-to-SNN conversion on the precision of neurons. Besides, result from Table 2 validates H-MT’s excellence with a bigger number of thresholds.

5.4 Effectiveness of H-MT’s Communication Mechanism

As the results with $16^{y=x}$ in Table 1 show, although there are very few identity modules directly used only at the very beginning of ResNet18/34 and VGG-16bn, replacing them with H-MT across these models can cause some performance decline. To magnify the effect of communication mechanism, we conduct conversion with H-MT on ViT-base framework. In order to prevent great accuracy loss, we will not replace all the identity modules. Instead, we follow the

Vit-base		Time-step $T = 16$	
Method		Accuracy(%)	
H-MT - 4/Opt	w/ communication	74.942	
	w/o communication	69.214	
H-MT - 4/999	w/ communication	73.878	
	w/o communication	63.482	

Table 4: Accuracy of H-MT(Ours) with $n = 4$ thresholds with or without communication mechanism for simplified conversion of ViT-base. Thresholds here are chosen through either the generalized threshold optimization method or the traditional policy for comparison.

policy of (Huang et al. 2025). Before fed into convolution or linear layer, input will first be fed into spiking neuron simulating identity mapping and be turned into spiking form.

As shown in Table 4, the effect of communication is significant. There will be a great drop of accuracy without communication mechanism no matter what threshold policy applied. Besides, H-MT with the generalized optimization method outperforms H-MT with traditional policy a lot when communication mechanism is not in use. Therefore, both communication mechanism and the generalized optimization method are excellent.

5.5 Effectiveness of Generalized Threshold Optimization Method

As shown in Table 1, 3 and 4, H-MT achieves better conversion with the generalized threshold optimization method, especially when simulating ReLU or $y = x$ across ResNet18, ResNet34 and ViT-base. For example, ResNet34 after such conversion can keep an accuracy of 75.80% through 16 inference time-steps. Besides, results from Table 2 further validate the effectiveness of the generalized optimization method in enhancing accuracy and reducing energy ratio.

However, the generalized threshold optimization method doesn’t overall outperform traditional method. When conducting conversion on VGG-16bn, we find out that with less inference time-steps, the traditional method has a little advantage, but is defeated with more inference time-steps.

From our point of view, such results are largely because the assumptions involved (Equation (19) and Assumption 4.2) are too strong and not so universal, but the generalized optimization method for threshold of spiking neuron is indeed of value and does provide a promising direction for future research of ANN-to-SNN conversion.

6 Conclusion

This article firstly introduces a design of a novel multi-threshold spiking neuron structure called Harmony Multi-threshold Spiking Neuron (H-MT) that simulates $y = x$. H-MT is an extend version of classical IF neuron, containing hybrid structure of a Pos Neuron and a Neg neuron, as well as a nova communication mechanism. It is of harmonious mathematical property which many multi-threshold spiking neuron don’t enjoy and its structure is of potential to simulate ReLU with higher precision. To adapt to H-MT, it then includes a generalized optimization method for threshold of spiking neurons simulating $y = x$ or ReLU, which does further enhance the performance.

The methods proposed in this article of course have plenty of room for improvement. H-MT is not so of advantage in efficiency and energy consumption in exchange for its mathematical property, and its performance of simulating $y = x$ is in urgent need of further advancing. Meanwhile, the generalized optimization method may require lots of iterations to get an ideal threshold, and isn’t generalized enough based on the assumptions. We do hope this article can provide guidance and inspiration for future research, thereby better addressing these challenges.

Acknowledgments

We would like to thank Yuanhong Tang for helpful discussion. This work was supported by the National Natural Science Foundation of China (U24B20140, 62422601 and 62302016), Beijing Municipal Science and Technology Program (Z241100004224004), Beijing Nova Program (20230484362, 20240484703), and State Key Laboratory of General Artificial Intelligence.

References

- Bohnstingl, T.; Woźniak, S.; Pantazi, A.; and Eleftheriou, E. 2022. Online Spatio-Temporal Learning in Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*.
- Bu, T.; Ding, J.; Yu, Z.; and Huang, T. 2022a. Optimized Potential Initialization for Low-Latency Spiking Neural Networks. In *AAAI Conference on Artificial Intelligence*.
- Bu, T.; Fang, W.; Ding, J.; DAI, P.; Yu, Z.; and Huang, T. 2022b. Optimal ANN-SNN Conversion for High-Accuracy and Ultra-Low-Latency Spiking Neural Networks. In *International Conference on Learning Representations*.
- Bu, T.; Li, M.; and Yu, Z. 2025. Inference-Scale Complexity in ANN-SNN Conversion for High-Performance and Low-Power Applications. In *Computer Vision and Pattern Recognition Conference*.
- Cao, Y.; Chen, Y.; and Khosla, D. 2015. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition. *International Journal of Computer Vision*.
- Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; Liao, Y.; Lin, C.-K.; Lines, A.; Liu, R.; Mathaikutty, D.; McCoy, S.; Paul, A.; Tse, J.; Venkataramanan, G.; Weng, Y.-H.; Wild, A.; Yang, Y.; and Wang, H. 2018. Loihi: a Neuro-morphic Manycore Processor with On-Chip Learning. *IEEE Micro*.
- de G. Matthews, A. G.; Hron, J.; Rowland, M.; Turner, R. E.; and Ghahramani, Z. 2018. Gaussian Process Behaviour in Wide Deep Neural Networks. In *International Conference on Learning Representations*.
- DeBole, M. V.; Taba, B.; Amir, A.; Akopyan, F.; Andreopoulos, A.; Risk, W. P.; Kusnitz, J.; Ortega Otero, C.; Nayak, T. K.; Appuswamy, R.; Carlson, P. J.; Cassidy, A. S.; Datta, P.; Esser, S. K.; Garreau, G. J.; Holland, K. L.; Lekuch, S.; Mastro, M.; McKinstry, J.; di Nolfo, C.; Paulovicks, B.; Sawada, J.; Schleupen, K.; Shaw, B. G.; Klamo, J. L.; Flickner, M. D.; Arthur, J. V.; and Modha, D. S. 2019. TrueNorth: Accelerating From Zero to 64 Million Neurons in 10 Years. *Computer*.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition*.
- Deng, S.; and Gu, S. 2021. Optimal Conversion of Conventional Artificial Neural Networks to Spiking Neural Networks. In *International Conference on Learning Representations*.
- 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 *International Joint Conference on Artificial Intelligence*.
- Ding, J.; Zhang, J.; Huang, T.; Liu, J. K.; and Yu, Z. 2025. Assisting Training of Deep Spiking Neural Networks With Parameter Initialization. *IEEE Transactions on Neural Networks and Learning Systems*.
- Duan, C.; Ding, J.; Chen, S.; Yu, Z.; and Huang, T. 2022. Temporal Effective Batch Normalization in Spiking Neural Networks. In *Advances in Neural Information Processing Systems*.
- Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep Residual Learning in Spiking Neural Networks. In *Advances in Neural Information Processing Systems*.
- Gerstner, W.; Kistler, W. M.; Naud, R.; and Paninski, L. 2014. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press.
- Gygax, J.; and Zenke, F. 2025. Elucidating the Theoretical Underpinnings of Surrogate Gradient Learning in Spiking Neural Networks. *Neural Computation*.
- 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 *Computer Vision and Pattern Recognition*.
- Hao, Z.; Bu, T.; Ding, J.; Huang, T.; and Yu, Z. 2023. Reducing ANN-SNN Conversion Error through Residual Membrane Potential. In *AAAI Conference on Artificial Intelligence*.
- 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 *Advances in Neural Information Processing Systems*.
- Horowitz, M. 2014. 1.1 Computing's Energy Problem (and What We Can Do About It). In *IEEE International Solid-State Circuits Conference Digest of Technical Papers*.
- Huang, Z.; Fang, W.; Bu, T.; Xue, P.; Hao, Z.; Liu, W.; Tang, Y.; Yu, Z.; and Huang, T. 2025. Differential Coding for Training-Free ANN-to-SNN Conversion. In *International Conference on Machine Learning*.
- 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 *ACM International Conference on Multimedia*.
- 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 *International Conference on Learning Representations*.
- Kim, J.; Kim, H.; Huh, S.; Lee, J.; and Choi, K. 2018. Deep Neural Networks with Weighted Spikes. *Neurocomputing*.
- 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 *International Conference on Machine Learning*.

- Li, Y.; Zhao, D.; and Zeng, Y. 2022. BSNN: Towards Faster and Better Conversion of Artificial Neural Networks to Spiking Neural Networks with Bistable Neurons. *Frontiers in Neuroscience*.
- Luo, X.; Yao, M.; Chou, Y.; Xu, B.; and Li, G. 2024. Integer-Valued Training and Spike-Driven Inference Spiking Neural Network for High-Performance and Energy-Efficient Object Detection. In *European Conference on Computer Vision*.
- Lv, L.; Fang, W.; Yuan, L.; and Tian, Y. 2024. Optimal ANN-SNN Conversion with Group Neurons. In *International Conference on Acoustics, Speech and Signal Processing*.
- Maass, W. 1997. Networks of Spiking Neurons: The Third Generation of Neural Network models. *Neural Networks*.
- Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2023. Towards Memory-and-Time-Efficient Backpropagation for Training Spiking Neural Networks. In *International Conference on Computer Vision*.
- Merolla, P. A.; Arthur, J. V.; Alvarez-Icaza, R.; Cassidy, A. S.; Sawada, J.; Akopyan, F.; Jackson, B. L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. 2014. A Million Spiking-neuron Integrated Circuit with a Scalable Communication Network and Interface. *Science*.
- 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*.
- Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; et al. 2019. Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture. *Nature*.
- Rueckauer, B.; Lungu, I.-A.; Hu, Y.; and Pfeiffer, M. 2016. Theory and Tools for the Conversion of Analog to Spiking Convolutional Neural Networks. *arXiv preprint arXiv:1612.04052*.
- Shi, X.; Hao, Z.; and Yu, Z. 2024. Spikingresformer: Bridging Resnet and Vision Transformer in Spiking Neural Networks. In *Computer Vision and Pattern Recognition*.
- Wang, X.; and Zhang, Y. 2023. MT-SNN: Enhance Spiking Neural Network with Multiple Thresholds. *arXiv preprint arXiv:2303.11127*.
- Wang, Y.; Zhang, M.; Chen, Y.; and Qu, H. 2022. Signed Neuron with Memory: Towards Simple, Accurate and High-Efficient ANN-SNN Conversion. In *International Joint Conference on Artificial Intelligence*.
- Wang, Z.; Fang, Y.; Cao, J.; Ren, H.; and Xu, R. 2025. Adaptive calibration: A unified conversion framework of spiking neural networks. In *AAAI Conference on Artificial Intelligence*.
- Xiao, M.; Meng, Q.; Zhang, Z.; He, D.; and Lin, Z. 2022. Online Training Through Time for Spiking Neural Networks. In *Advances in Neural Information Processing Systems*.
- Zhao, L.; Huang, Z.; Ding, J.; and Yu, Z. 2025. TTFS-Former: A TTFS-Based Lossless Conversion of Spiking Transformer. In *International Conference on Machine Learning*.
- Zhu, Y.; Ding, J.; Huang, T.; Xie, X.; and Yu, Z. 2024. Online Stabilization of Spiking Neural Networks. In *International Conference on Learning Representations*.