SPIKE CALIBRATION: BRIDGING THE GAP BETWEEN ANNS AND SNNS IN ANN-SNN CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) have attracted great attention due to the distinctive characteristics of low power consumption and temporal information processing. ANN-SNN conversion, as the most commonly used method, can make converted SNNs achieve comparable performance as ANNs on large-scale datasets. However, the performance degrades severely under low time-steps, which hampers the practical applications of SNNs on neuromorphic chips. In this paper, instead of evaluating different conversion errors and then eliminating these errors, we define offset spike to measure the deviation degree of actual and desired firing rates of SNNs. We make a detailed analysis of offset spike and point out that the case of firing one more (or less) spike is the main reason for conversion error. Based on this, we propose an optimization strategy based on shifting initial membrane potential and theoretically prove the corresponding optimal shifting distance to calibrate the spike. In addition, we also note that our method has a unique iterative property to further reduce conversion error. The experimental results show that our proposed method achieves state-of-the-art performance on CIFAR-10, CIFAR-100, and ImageNet datasets. For example, we reach top-1 accuracy of 67.12% on ImageNet with 6 time-steps. To the best of our knowledge, this is the first time ANN-SNN conversion can simultaneously achieve high accuracy and ultra-low latency on the complex dataset.

1 INTRODUCTION

Acclaimed as the third generation of artificial neural network (Maass, 1997), Spiking Neural Networks (SNNs) have brought brand-new inspiration in computational neuroscience. As the corresponding neuron fires spikes only when the current membrane potential exceeds the firing threshold, SNNs have the distinctive characteristics of binary output, high sparsity, and biological plausibility. Therefore, compared with traditional ANN models, SNNs can further improve computational efficiency and reduce power consumption, which facilitates its remarkable superiority in the application of neuromorphic chips (Merolla et al., 2014; Davies et al., 2018; DeBole et al., 2019). Considering that SNNs have not found an effective learning algorithm yet, ANN-SNN conversion and back-propagation through time (BPTT) are still the two most concerning training methods at present. Compared with BPTT, ANN-SNN conversion avoids the non-differentiable problem in the direct training procedure of SNNs and thus reduces the overall training complexity.

ANN-SNN conversion aims to establish the mapping relationship between the activation output and average firing rate. Traditional conversion methods exploit larger time-steps to overcome the conversion error and thus achieve high performance (Diehl et al., 2015). Many of the following works have attempted to optimize the performance from multiple perspectives, including using the soft-reset mechanism (Han et al., 2020; Rueckauer et al., 2017), proposing more adaptive activation functions (Cao et al., 2015; Ho & Chang, 2021; Bu et al., 2022b), adopting a trainable threshold (Sengupta et al., 2019; Ding et al., 2021; Bu et al., 2022a), etc. But these strategies cannot effectively eliminate the error caused by the deviation between the actual and desired firing rate, especially when the number of time-step is small. Some recent works consider to compensate the errors by introducing burst spikes (Li & Zeng, 2022) and signed spiking neurons (Li et al., 2022). Unlike these works, our paper attempts to eliminate the error with vanilla spiking neurons and answer the question of where the extreme performance of ANN-SNN conversion is.

In this paper, we observe and identify the source of the conversion error and propose an iterative optimization method based on shifting initial membrane potential, which can fulfill accurate mapping between ANNs and SNNs under ideal situation. Our main contributions are summarized as follows:

1. We introduce the concept of offset spike to infer the deviation between the actual SNN firing rate and desired SNN firing rate. We point out that the case of firing one more (or less) spike is the main reason for conversion error.
2. We propose a method to judge the offset spike based on the residual membrane potential and an optimization method to eliminate the conversion error through shifting up or down the initial membrane potential. We derive the optimal shifting distance and prove that one spike can be increased or decreased under this condition.
3. We evaluate our methods on CIFAR-10/100 and ImageNet datasets. The proposed method outperforms the state-of-the-art ANN-SNN conversion methods using fewer time-steps. For example, we achieve 67.12% top-1 accuracy on ImageNet with only 6 time-steps (4 time-steps calibration and 2 time-steps inference). Moreover, it is worth noting that we have reached the same level of performance as BPTT under the condition of reducing memory and computing resources significantly.
4. We discover that our proposed method has an iterative property. Under an ideal situation, the deviation within the range of k spikes will be eliminated entirely after adopting our approach for k times. After 4 iterations, the mean-square error between actual and desired firing rate of the output layer can reach 0.001 for the ResNet-20 model on CIFAR-100.

2 Related Works

The principle of ANN-SNN conversion is to map the parameters from pre-trained ANN models to SNNs, which avoids training SNNs directly and reduces energy consumption significantly. The primary target is to match the activation value of ANNs and average firing rate of SNNs. Cao et al. (2015) were the pioneers to do research in this field, and they replaced ReLU activation layers in ANNs with spiking neurons to fulfill the conversion procedure. Ho & Chang (2021); Bu et al. (2022b) proposed new activation functions, which better fit the finiteness and discreteness of the spike firing rate. Rueckauer et al. (2017); Han et al. (2020) adopted "reset-by-subtraction" mechanism, which alleviated the problem of information loss and improved the precision of conversion process effectively. For the setting of firing threshold, various strategies have been proposed, including RobustNorm (Rueckauer et al., 2017), SpikeNorm (Sengupta et al., 2019), adjustable threshold (Han et al., 2020; Ding et al., 2021; Ho & Chang, 2021; Bu et al., 2022a;b), etc. Recently, spiking neural networks with high accuracy and low latency have become the focus and target of academic research. To reduce the time latency of the network, one must carefully deal with the exact spiking time of neurons. Deng & Gu (2021); Li et al. (2021) fine-tuned the bias in each layer under the uniform current assumption. Nevertheless, the actual current would never be distributed uniformly. In terms of expectation, Bu et al. (2022b) proved that one-half of the threshold is the optimal value for initial membrane potential, and charging this value can prompt neurons to spike more uniformly. However, as the authors pointed out: there is still a mismatch between ANN and SNN due to the so called "unevenness error". In addition, other methods like burst spikes (Li & Zeng, 2022) and signed spiking neurons (Wang et al., 2022a; Li et al., 2022), have also been introduced to further improve the performance. These works aim to alleviate the conversion loss. However, they damaged the biological plausibility and binary property of spiking neurons.

In addition to ANN-SNN conversion, back-propagation with exact spike time is another common way to train SNNs. Surrogate gradient (O'Connor et al., 2018; Zenke & Ganguli, 2018; Bellec et al., 2018; Wu et al., 2018; 2019; Kim & Panda, 2020; Zenke & Vogels, 2021) has been widely used to tackle the non-differentiable problem in the training process, which substitutes the Heaviside function with a derivable function. With the help of the surrogate gradient, back-propagation through time enables the network to adjust weights by focusing every exact time-step. On this basis, Fang et al. (2021); Rathi & Roy (2021); Guo et al. (2022) further attempted the optimization of hyper-parameters and gradient. Bohte et al. (2002); Kheradpisheh & Masquelier (2020); Zhang & Li (2020) proposed a timing-based learning method, which viewed the specific spike firing time as significant temporal information to transmit between layers. Still, this type of method only applies to shallow networks at present. In addition, hybrid training methods have attracted extensive attention

recently. Wang et al. (2022b); Rathi & Roy (2021) combined ANN-SNN conversion with BPTT to obtain higher performance under low latency. Kim et al. (2020) adopted rate-coding and time-coding simultaneously to train SNNs with fewer spikes. Mostafa (2017); Zhou et al. (2021); Zhang & Li (2020) established a linear transformation about the spike firing time from adjacent layers, which obtained SNNs under the training mode of ANNs. Besides, BPTT can calibrate the spike time in the training phase (Rathi et al., 2020). Inspired by these works, we incorporate the concept of calibration spike time into the conversion pipeline to bridge the gap between ANNs and SNNs.

3 PRELIMINARIES

3.1 NEURON MODELS

For ANNs, the input $\boldsymbol{a}^{l-1}$ to layer l is mapped to the output $\boldsymbol{a}_l$ by a linear transformation matrix $\boldsymbol{W}^l$ and a nonlinear activation function $f(\cdot)$, that is ($l = 1, 2, 3, \cdots, L$):

$$\boldsymbol{a}^l = f(\boldsymbol{W}^l \boldsymbol{a}^{l-1}), \tag{1}$$

where $f(\cdot)$ is often set as the ReLU activation function.

For SNNs, we adopt Integrate-and-Fire (IF) Neuron model (Gerstner & Kistler, 2002), which is similar to the previous works (Cao et al., 2015; Diehl et al., 2015). To minimizes information loss during inference, our neurons perform "reset-by-subtraction" mechanism (Han et al., 2020), which means that the membrane potential is subtracted by the firing threshold θ^l after firing. The overall kinetic equations of IF Neuron can be expressed as follows:

$$\boldsymbol{v}^l(t) = \boldsymbol{v}^l(t-1) + \boldsymbol{I}^l(t) - \boldsymbol{s}^l(t)\theta^l, \tag{2}$$

$$\boldsymbol{I}^l(t) = \boldsymbol{W}^l \boldsymbol{s}^{l-1}(t)\theta^{l-1}. \tag{3}$$

Here $\boldsymbol{v}^l(t)$ and $\boldsymbol{I}^l(t)$ denote the membrane potential and input current of layer l at t-th time-step, respectively. $\boldsymbol{W}$ is the synaptic weights between layer $l-1$ and layer l, and θ^l is the spike firing threshold in l-th layer. $\boldsymbol{s}^l(t)$ represents whether the spike fires at time-step t. For i-th neuron, if the current potential exceed the firing threshold θ^l, the neuron will emit a spike. This firing rule can be described by the equation below.

$$s_i^l(t) = \begin{cases} 1, & v_i^l(t-1) + I_i^l(t) \geqslant \theta^l \\ 0, & v_i^l(t-1) + I_i^l(t) < \theta^l \end{cases}. \tag{4}$$

If not otherwise specified, the subscript x_i denotes the i-th element of $\boldsymbol{x}$, which corresponds to the i-th neuron.

3.2 ANN-SNN CONVERSION

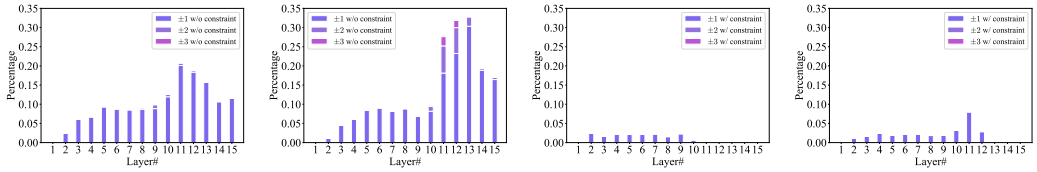
The main principle of ANN-SNN conversion is to map the firing rates (or postsynaptic potential) of spiking neurons to the ReLU activation output of artificial neurons. Specifically, by summing summing equation 2 from $t = 1$ to $t = T$, and then substituting variable $\boldsymbol{I}^l(t)$ with $\boldsymbol{W}^l \boldsymbol{s}^{l-1}(t)\theta^{l-1}$ using equation 3, and finally dividing T on both sides, we get the following equation:

$$\frac{\sum_{t=1}^{T} \boldsymbol{s}^l(t)\theta^l}{T} = \boldsymbol{W}^l \frac{\sum_{t=1}^{T} \boldsymbol{s}^{l-1}(t)\theta^{l-1}}{T} + \left(-\frac{\boldsymbol{v}^l(T) - \boldsymbol{v}^l(0)}{T}\right), \tag{5}$$

where T denotes the total simulation cycle. For simplicity, we use the average postsynaptic potential $\boldsymbol{\phi}^l(T)$ to substitute the term $\sum_{t=1}^{T} \boldsymbol{s}^l(t)\theta^l/T$ in equation 5, then we get

$$\boldsymbol{\phi}^l(T) = \boldsymbol{W}^l \boldsymbol{\phi}^{l-1}(T) + \left(-\frac{\boldsymbol{v}^l(T) - \boldsymbol{v}^l(0)}{T}\right). \tag{6}$$

Equation 6 can be approximated by a linear transformation between $\boldsymbol{\phi}^l(T)$ and $\boldsymbol{\phi}^{l-1}(T)$ as T tends to infinity, which is exactly the same as the forward propagation (equation 1) in ANNs due to $\boldsymbol{\phi}^l(T) \geqslant 0$. This result implies that we can achieve lossless ANN-SNN conversion when T tends to infinity. However, the performance of converted SNNs degrades seriously under the condition of



(a) (b) (c) (d)

Figure 1: The distribution of offset spike in each layer. (a) and (c): VGG-16 on CIFAR-10, (b) and (d): VGG-16 on CIFAR-100. w/ constraint denotes the constraint of $\boldsymbol{\psi}^{l-1} = 0$ through the rectification of the spikes.

short time-steps T (Rueckauer et al., 2017; Han et al., 2020). To achieve high-performance SNNs under low latency, Bu et al. (2022b) proposes to replace the commonly used ReLU activation function of source ANNs with the quantization clip-floor-shift (QCFS) function:

$$\boldsymbol{a}^l = f(\boldsymbol{a}^{l-1}) = \frac{\lambda^l}{L}\text{clip}\left(\left\lfloor \frac{\boldsymbol{W}^l \boldsymbol{a}^{l-1} L}{\lambda^l} + \frac{1}{2} \right\rfloor, 0, L\right), \tag{7}$$

where L denotes the quantization step of ANNs, λ^l is the trainable threshold of the outputs in layer l of ANNs, which is mapped to the threshold θ^l in layer l of SNNs. This paper follows the conversion framework of Bu et al. (2022b) with QCFS function.

4 METHODS

In this section, we first compare the outputs of ANNs and converted SNNs in each layer. We introduce the offset spike to measure the deviation degree between the actual firing rate and the desired firing rate of SNNs. Then, we point out that the offset spike of being one accounts for the main part of each layer. Based on this, we propose sufficient conditions to determine if the offset spike exists and the sign of the offset spike, and the spike calibration strategy to eliminate conversion error through shifting initial membrane potential.

4.1 OFFSET SPIKE AND ITS DISTRIBUTION

ANN-SNN conversion errors can be divided into different categories: clipping error, quantization error (flooring error), and unevenness error (deviation error). In previous works (Han et al., 2020; Li et al., 2021; Meng et al., 2022b), those errors are eliminated (or reduced) separately, and the unevenness error (deviation error) cannot be eliminated so far. As we find that the essential cause of most conversion errors comes from the remaining term $-\frac{\boldsymbol{v}^l(T)-\boldsymbol{v}^l(0)}{T}$ in equation 5, we consider to reduce conversion errors directly based on the prior knowledge of the remaining term. In order to measure the deviation degree between the actual firing rate and the desired firing rate, we first introduce the definition of the offset spike.

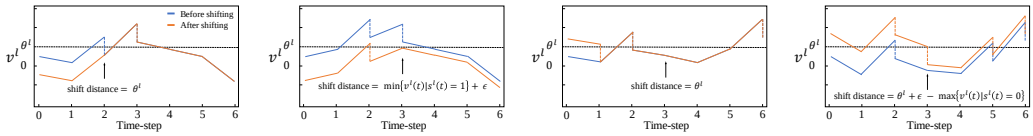
Definition 1. We define **OFFSET SPIKE** as the difference between the desired total spike count and actual spike count, that is

$$\boldsymbol{\psi}^l = \frac{\boldsymbol{a}^l T}{\theta^l} - \sum_{t=1}^{T} \boldsymbol{s}^l(t), \tag{8}$$

where $\boldsymbol{\psi}^l$ stands for the offset spike of layer l. For example, $\psi_i^l = \pm k$ represents that the gap between the actual and desired firing rate of the i-th neuron in layer l of the SNN is k spikes.

We further investigate the detailed outputs of ANN and SNN in each layer. We train the source ANN with the QCFS activation function (equation 7) and then convert it to an SNN. Fig. 1a-b illustrate the distribution of offset spike for the converted SNNs with VGG-16 structure on CIFAR-10 and CIFAR-100, respectively. We have the following observation.

Observation 1. $\boldsymbol{\psi}^l = \pm 1$ accounts for the main part in each layer and $\boldsymbol{\psi}^l = \pm 3$ rarely happens.



(a) (b) (c) (d)

Figure 2: Shift up (down) initial membrane potential can increase (decrease) one output spike.

Considering the cumulative effect of conversion error in the deep layer, the offset spike $\boldsymbol{\psi}^l$ in layer l can be considered as the joint effects of the offset spike $\boldsymbol{\psi}^{l-1}$ in layer $l-1$ and the conversion error in layer l, and tends to get bigger and bigger with the increase of the layer.

For a deeper analysis of the offset spike in each layer, we rectify the firing rates of neurons in layer $l-1$ of SNNs (with constraint) to make $\boldsymbol{\psi}^{l-1}=0$, and then compute the offset spike $\boldsymbol{\psi}^l$ in layer l. After the rectification for each layer, the distribution of offset spike for the converted SNNs with VGG-16 structure on CIFAR-10 and CIFAR-100 are shown in Fig. 1c-d, respectively. We have the following observation.

Observation 2. With constraint, $\boldsymbol{\psi}^l=\pm 1$ accounts for the main part in each layer and $|\boldsymbol{\psi}^l|>1$ rarely happens.

Observations 1 and 2 show that the situation of firing one more (or less) spike is the main reason for the conversion error, which implies that we can eliminate the conversion error after adjusting $\sum_{t=1}^{T}\boldsymbol{s}^l(t)$ with ± 1.

4.2 Judge conversion error through residual membrane potential

Before we propose the optimal strategy to adjust the output spikes to eliminate the offset spike $\boldsymbol{\psi}^l$, we need to determine if the offset spike exists and the sign of the offset spike. If the sign of offset spike is positive, which corresponds to the situation ANN output is larger than SNN output, the spiking neurons should fire more spikes to eliminate the offset spike, and otherwise fire fewer spikes.

In the practical application of SNNs, we cannot directly obtain the specific value of the offset spike $\boldsymbol{\psi}^l$. Fortunately, we find we can determine the sign of $\boldsymbol{\psi}^l$ according to the value of the residual membrane potential $\boldsymbol{v}^l(T)$. We have the following theorem:

Theorem 1. *Supposing that an ANN with QCFS activation function (equation 7) is converted to an SNN with $L=T, \lambda^l=\theta^l, \boldsymbol{v}^l(0)=\theta^l/2$, and the inputs to the l-th layer of ANN and SNN are the same, that is, $\boldsymbol{a}^{l-1}=\boldsymbol{\phi}^{l-1}(T)$. Then for any i-th element of the l-th layer, we will have the following conclusions:*
(i) If $\phi_i^l(T)>0$ and $v_i^l(T)<0$, we will have $\phi_i^l(T)>a_i^l$ and $\psi_i^l<0$.
(ii) If $\phi_i^l(T)<\theta^l$ and $v_i^l(T)\geqslant\theta^l$, we will have $\phi_i^l(T)<a_i^l$ and $\psi_i^l>0$.

The detailed proof is provided in the Appendix. (i). Theorem 1 implies that if the postsynaptic potential is larger than 0 and the residual membrane potential is smaller than 0, we can conclude that the spiking neuron fires more spikes as expected and the sign of offset spike is negative. (ii) implies that if the postsynaptic potential is smaller than 0 and the residual membrane potential is larger than θ, we can conclude that the spiking neuron fires fewer spikes as expected and the sign of offset spike is positive.

4.3 Eliminate conversion error through shifting initial membrane potential

As the situation of firing one more (or less) spike ($\boldsymbol{\psi}^l=\pm 1$) is the main reason for the conversion error, here we propose an optimization strategy to rectify the output spike $\sum_{t=1}^{T}\boldsymbol{s}^l(t)$ by adding or subtracting one spike, thereby eliminating conversion error. Specifically, we consider to adjust the value of $\sum_{t=1}^{T}\boldsymbol{s}^l(t)$ by shifting up or down the corresponding initial membrane potential $\boldsymbol{v}^l(0)$. One intuitive explanation is that higher initial membrane potential will make the spiking neurons fire earlier and increase the firing rates during the period $[0,\ T]$, while lower initial membrane potential

will make the spiking neurons fire later and decrease the firing rates. The following theorem gives the optimal shifting distance when we attempt to move $\sum_{t=1}^{T} \boldsymbol{s}^l(t)$ (and $\boldsymbol{\psi}^l$) by ± 1 distance.

Theorem 2. *If we use $s_i^l(t)$ and $\widetilde{s}_i^l(t)$ to denote the binary spike of the i-th neuron in layer l at time-step t before and after optimization, $v_i^l(0)$ and $\widetilde{v}_i^l(0)$ to represent the initial membrane potential before and after optimization, then $\forall \epsilon \in (0, \theta^l)$, we will have the following conclusions:*

(i) If we set $\widetilde{v}_i^l(0) = v_i^l(0) - \max\left(\theta^l, \min\left\{v_i^l(t) | s_i^l(t) = 1\right\} + \epsilon\right)$, then $\sum\limits_{t=1}^{T} \widetilde{s}_i^l(t) = \sum\limits_{t=1}^{T} s_i^l(t) - 1$.

(ii) If we set $\widetilde{v}_i^l(0) = v_i^l(0) + \max\left(\theta^l, \theta^l + \epsilon - \max\left\{v_i^l(t) | s_i^l(t) = 0\right\}\right)$, then $\sum\limits_{t=1}^{T} \widetilde{s}_i^l(t) = \sum\limits_{t=1}^{T} s_i^l(t) + 1$.

The detailed proof is provided in the Appendix. Note that the variable ϵ illustrates that as long as the initial membrane potential is in a certain range, the number of output spikes can be guaranteed to increase or decrease by 1. Fig.2 shows four different situations before and after shifting $\boldsymbol{v}^l(0)$, which verifies the effectiveness of our theorem.

By combining Theorem 1 and Theorem 2, we propose the complete spike calibration algorithm. Our method can be divided into two stages. Firstly, for l-th layer, we spend ρ time-steps to detect the specific spike firing situation. According to Theorem 1, if $v_i^l(\rho) < 0$ (or $v_i^l(\rho) \geqslant \theta^l$), by combining $\phi_i^l(\rho)$, we can infer that $\phi_i^l(\rho)$ is actually larger (or smaller) than the expected average postsynaptic potential a_i^l, and $\psi_i^l < 0$ (or $\psi_i^l > 0$). In addition, we will preserve the membrane potential after each time-step, which will be used to calculate the subsequent optimal shifting distance.

In the second stage, based on Theorem 2, we will calculate the optimal shifting distance of the initial membrane potential for specific neurons with conversion error. Generally speaking, if $\psi_i^l < 0$, we will shift down its initial membrane potential by calculating (i) in Theorem 2, if $\psi_i^l > 0$, we will shift up its initial membrane potential through adopting (ii) in Theorem 2. After optimizing the initial membrane potential, we will spend T time-steps implementing the test on corresponding datasets and deliver the output to $l+1$-th layer.

4.4 Iterative property of our optimization method

In the previous section, we show that shifting up (or down) the initial membrane potential can move the situation of $\boldsymbol{\psi}^l = \pm 1$ to $\boldsymbol{\psi}^l = 0$. In fact, our method also converts the situation of $\boldsymbol{\psi}^l = \pm k$ to $\boldsymbol{\psi}^l = \pm(k-1)$. As long as the offset spike $\boldsymbol{\psi}^l$ is not zero, the performance of converted SNNs will degrade. One important problem is whether we can further eliminate the offset spike of the situation $\boldsymbol{\psi}^l \geqslant 2$?

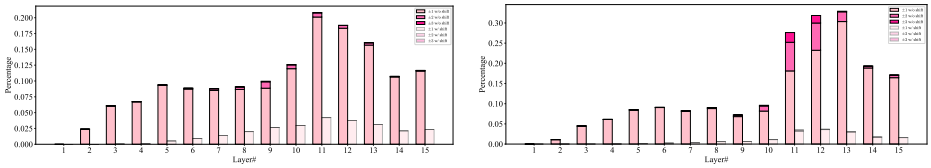
Fortunately, we find that our optimization method has an iterative property. One can reuse Theorem 2 to increase (or decrease) one output spike each time. Of course, it comes at a significant computational cost. In the Experiments Section, we will show that the performance of the converted SNN increases with the iteration. Typically, we can achieve high-performance and low-latency SNNs with only one iteration.

5 Experiments

In this section, we choose image classification datasets to validate the effectiveness and performance of our proposed methods, including CIFAR-10 (LeCun et al., 1998), CIFAR-100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) datasets. The network architecture selected for evaluating includes VGG-16 (Simonyan & Zisserman, 2014), ResNet-18, ResNet-20 and ResNet-34 (He et al., 2016). As for the setting of the hyper-parameter ρ, we set $\rho = 4$ for CIFAR-10/100 and $\rho = 8$ for ImageNet if there are no special instructions. More details of the experiments settings are provides in the Appendix.

5.1 Effectiveness of the proposed method

To illustrate the effectiveness of our proposed shifting operations of initial membrane potential, we compare bar charts of offset spike $\boldsymbol{\psi}^l$ in each layer of SNNs before and after the shift. Fig. 3 il-



(a) CIFAR-10,VGG-16 (b) CIFAR-100,VGG-16

Figure 3: The distribution of offset spike before and after optimization

Table 1: Comparison with the state-of-the-art ANN-SNN conversion methods

Method	ANN	Architecture	T=1	T=2	T=4	T=8	T=16	T=32
CIFAR-100 Dataset								
SNM	74.13%	VGG-16	-	-	-	-	-	71.80%
SNNC-AP	77.89%		-	-	-	-	-	73.55%
OPI	76.31%		-	-	-	60.49%	70.72%	74.82%
QCFS	76.28%		-	63.79%	69.62%	73.96%	76.24%	77.01%
Ours	76.28%		74.24%	76.03%	76.26%	76.52%	76.77%	76.96%
RMP	68.72%	ResNet-20	-	-	-	-	-	27.64%
OPI	70.43%		-	-	-	23.09%	52.34%	67.18%
QCFS	69.94%		-	19.96%	34.14%	55.37%	67.33%	69.82%
Ours	69.97%		59.22%	64.21%	65.18%	67.17%	69.44%	70.29%
ImageNet Dataset								
SNNC-AP	75.36%	VGG-16	-	-	-	-	-	63.64%
SNM	73.18%		-	-	-	-	-	64.78%
OPI	74.85%		-	-	-	6.25%	36.02%	64.70%
QCFS	74.29%		-	-	-	19.12%	50.97%	68.47%
Ours	74.19%		63.84%	70.59%	72.94%	73.82%	74.09%	74.33%
SNNC-AP	75.66%	ResNet-34	-	-	-	-	-	64.54%
QCFS	74.32%		-	-	-	35.06%	59.35%	69.37%
Ours	74.22%		69.11%	72.66%	73.81%	74.17%	74.14%	73.93%

lustrated the results of VGG-16 networks on the CIFAR-10 and CIFAR-100 datasets. It can be observed that the shifting operations significantly reduce offset spike, that is, the deviation between $\boldsymbol{\phi}^l(T)$ and $\boldsymbol{a}^l$, for each layer. For vanilla without shift operation (denoted as "w/o shift" in Fig. 3), one can discover a magnification effect of spike count error from the 1-st to the 11-th layer. In contrast, the apparent magnification is alleviated with the proposed methods. From Fig. 3(b), we notice that the ± 2 and ± 3 offset spike in the VGG-16 model for CIFAR-100 have increased compared to those for CIFAR-10 (Fig. 3(a)). Using our method can clearly decrease these deviations and achieve a comparable error-free conversion.

5.2 Comparison with the state-of-the-art

We compare our methods with previous state-of-the-art ANN-SNN conversion works, including RMP (Han et al., 2020), SNM (Bu et al., 2022b), SNNC-AP (Li et al., 2021), QCFS (Bu et al., 2022b), on CIFAR-10 CIFAR-100 and ImageNet datasets. As we spend ρ time-steps in the first stage to acquire relevant temporal information about membrane potential, considering the fairness of comparison, we will compare the performance of other works at time-step $T + \rho$ with our performance at time-step T.

Tab. 1 reports the results on the CIFAR-100 dataset. For VGG-16, the performance of our method at time-step 1 ($\rho = 4$) outperforms that of SNM and SNNC-AP at time-step 32. Moreover, we achieve

Table 2: Comparison with other types of SNN training methods

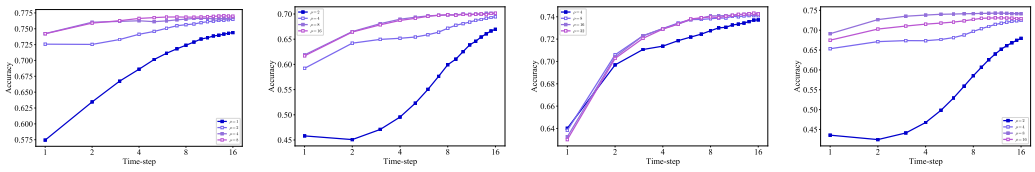
Dataset	Method	Type	Architecture	Accuracy	Time-step
CIFAR-100	Dual-Phase	Hybrid Training	VGG-16	70.08%	4
	Diet-SNN	Hybrid Training	VGG-16	69.67%	5
	RecDis-SNN	BPTT	VGG-16	69.88%	5
	Ours($\rho = 4$)	ANN-SNN conversion	VGG-16	74.24%	1
ImageNet	HC-STDB	Hybrid Training	ResNet-34	61.48%	250
	DSR	Supervised learning	PreAct-ResNet-18	67.74%	50
	STBP-tdBN	BPTT	ResNet-34	63.72%	6
	PLIF	BPTT	ResNet-34	67.04%	7
	TET	BPTT	ResNet-34	64.79%	6
	Ours($\rho = 4$)	ANN-SNN conversion	ResNet-34	67.12%	2

Table 3: The ratio and MSE after multiple iterations

Dataset	Architecture	Baseline		**Ours**		**Ours** $\times 2$		**Ours** $\times 4$	
		Ratio	MSE	Ratio	MSE	Ratio	MSE	Ratio	MSE
CIFAR-10	VGG-16	88.33%	0.120	97.65%	0.024	99.83%	0.002	99.84%	0.002
	ResNet-20	62.38%	0.512	82.56%	0.179	99.73%	0.003	99.76%	0.002
CIFAR-100	VGG-16	82.90%	0.192	98.42%	0.016	99.86%	0.001	99.86%	0.001
	ResNet-20	41.59%	1.641	67.08%	0.453	86.03%	0.165	91.29%	0.101

76.26% top-1 accuracy with 4 time-steps ($\rho = 4$), which is 2.30% higher than QCFS (73.96%, T=8) and 15.77% higher than OPI (60.49%, T=8). For ResNet-20, our method at time-step 1 ($\rho = 4$) surpasses the performance of RMP at time-step 32 (59.22% vs. 27.64%). The accuracy of our method is 65.18% at time-step 4 ($\rho = 4$), whereas the performance of QCFS and OPI are 55.37% and 23.09% at time-step 8, respectively. More results on CIFAR-10 are listed in the Appendix.

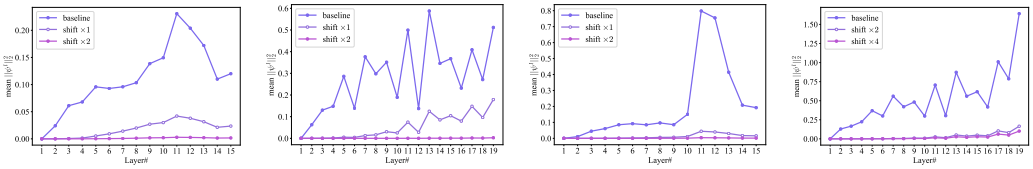
We further test the generalization of our method on the ImageNet (Tab. 1). For VGG-16, we achieve 73.82% top-1 accuracy at time-step 8 ($\rho = 8$), which outperforms QCFS (50.97%, T=16) by 22.85% and OPI (36.02%, T=16) by 37.80%. For ResNet-34, the performance of our method at time-step 1 ($\rho = 8$) outperforms that of SNM and SNNC-AP at time-step 32. Moreover, we achieve 74.17% with 8 time-steps ($\rho = 8$), which is 14.82% higher than QCFS (59.35%, T=16). These results show that our method can achieve better classification accuracy with fewer time-steps.



(a) (b) (c) (d)

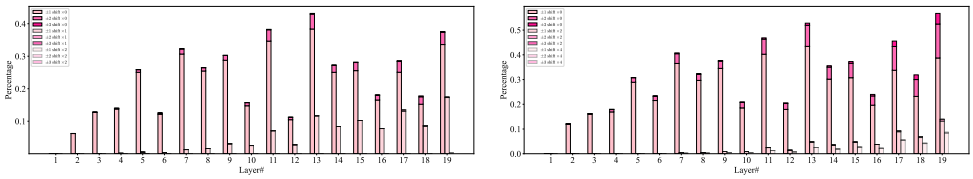
Figure 4: Influence of different ρ. (a) VGG-16 on CIFAR-100, (b) ResNet-20 on CIFAR-100, (c) VGG-16 on ImageNet, (d) ResNet-34 on ImageNet.

In addition, we compare our method with other types of SNN training methods (Hybrid Training & BPTT), including Dual-Phase (Wang et al., 2022b), Diet-SNN (Rathi & Roy, 2021), RecDis-SNN (Guo et al., 2022), HC-STDB (Rathi et al., 2020), STBP-tdBN (Zheng et al., 2021), PLIF (Fang et al., 2021), TET (Deng et al., 2022) and DSR (Meng et al., 2022a). Here we set $\rho = 4$ for the CIFAR-100 and ImageNet datasets. As reported in Tab. 2, our method achieves better accuracy on the CIFAR-100 dataset and comparable accuracy on the ImageNet dataset with the same time-steps. Note that the complexity of back-propagation chain in these methods is much higher than that of



(a) (b) (c) (d)

Figure 5: The MSE of conversion error after using iterative optimization. (a): VGG-16 on CIFAR-10, (b): ResNet-20 on CIFAR-10, (c): VGG-16 on CIFAR-100, (d): ResNet-20 on CIFAR-100.



(a) ResNet-20 on CIFAR-10 (b) ResNet-20 on CIFAR-100

Figure 6: The distribution of offset spike after using iterative optimization

ANN-SNN conversion, which consumes large amounts of memory and computing resources. All these results demonstrate the superiority of our method.

5.3 Effect of the inference time-step ρ

We further explore the influence of the hyper-parameter ρ in the first stage of our method. Fig. 4 shows the accuracy of the network with different values of ρ. For Fig.4(a)-4(d), the value of quantization level L in QCFS function are set to 4, 8, 16 and 8, respectively. We can find that the SNN accuracy tends to converge as ρ gradually approaches L. This phenomenon can be understood as follows. We use the QCFS activation function in source ANN and we have $\boldsymbol{a}^l \in \{k\theta^l/L | k = 0, 1, ..., L\}$ and $\boldsymbol{\phi}^l(T) \in \{k\theta^l/\rho | k = 0, 1, ..., \rho\}$. Thus, the mapping relationship between $\boldsymbol{a}^l$ and $\boldsymbol{\phi}^l(T)$ will become more accurate when ρ approaches L, which makes the temporal information obtained from the first stage more precise so as to improve the performance of the network.

5.4 Effect of the iterative optimization

In section 4.4, we explain that our method has an iterative property that can reduce the offset spike through multiple iterations. To demonstrate it, we define ratio as the percentage of $a_i^l = \phi_i^l(T)$ in the output layer. Despite this, we also consider the indicators of mean-square error (MSE), which is defined as $||\boldsymbol{\psi}^l||_2^2$. Tab. 3 reports the ratio and MSE of the output layer, in which the baseline denotes the performance without using our methods and $\times 2$ represents two iterations. Besides, we set $\rho = L = T$. From top to bottom in Tab. 3, the values of L are set to 4, 4, 4, 8, respectively. From Tab. 3 and Fig. 5, we can conclude that, generally, ratio and MSE in each layer continue to decrease as the number of iterations increases, which is consistent with the result in Fig. 6.

6 Conclusions

In this paper, we first define offset spike to measure the deviation degree of actual and desired firing rates of SNNs. Then we analyze the distribution of offset spike and point out that we can infer the specific value of the deviation according to the corresponding residual membrane potential. Furthermore, we propose an optimization method to eliminate offset spike by shifting up and down the initial membrane potential. Finally, we demonstrate the superiority of our method on CIFAR-10/100 and ImageNet datasets. Our results will further facilitate the relevant research and application of SNNs on neuromorphic chips.

REFERENCES

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, 2018.

Sander M Bohte, Joost N Kok, and Han La Poutre. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1-4):17–37, 2002.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pp. 421–436. Springer, 2012.

Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. In *AAAI Conference on Artificial Intelligence*, 2022a.

Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022b.

Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

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

Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al. TrueNorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52(5):20–29, 2019.

Jia Deng, Richard Socher, Lijia Li, Kai Li, and Feifei Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2015.

Jianhao Ding, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks. In *International Joint Conference on Artificial Intelligence*, pp. 2328–2336, 2021.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671, 2021.

Wulfram Gerstner and Werner M Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge university press, 2002.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. RecDis-SNN: Rectifying membrane potential distribution for directly training spiking neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 326–335, 2022.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13558–13567, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Nguyen-Dong Ho and Ik-Joon Chang. TCL: an ANN-to-SNN conversion with trainable clipping layers. In *ACM/IEEE Design Automation Conference (DAC)*, pp. 793–798. IEEE, 2021.

Saeed Reza Kheradpisheh and Timothée Masquelier. Temporal backpropagation for spiking neural networks with one spike per neuron. *International Journal of Neural Systems*, 30(06):2050027, 2020.

Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation- and timing-based learning rules for spiking neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 19534–19544, 2020.

Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in neuroscience*, pp. 1638, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

Chen Li, Lei Ma, and Steve Furber. Quantization framework for fast spiking neural networks. *Frontiers in Neuroscience*, 16, 2022.

Yang Li and Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes. In *International Joint Conference on Artificial Intelligence*, 2022.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning*, pp. 6316–6325, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*. OpenReview.net, 2017.

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

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12444–12453, 2022a.

Qingyan Meng, Shen Yan, Mingqing Xiao, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training much deeper spiking neural networks with a small number of time-steps. *Neural Networks*, 153: 254–268, 2022b.

Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345 (6197):668–673, 2014.

Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3227–3235, 2017.

Peter O'Connor, Efstratios Gavves, Matthias Reisser, and Max Welling. Temporally efficient deep learning with spikes. In *International Conference on Learning Representations*, 2018.

Nitin Rathi and Kaushik Roy. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–9, 2021. doi: 10.1109/TNNLS.2021.3111897.

Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations*, 2020.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

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

Yuchen Wang, Malu Zhang, Yi Chen, and Hong Qu. Signed neuron with memory: Towards simple, accurate and high-efficient ANN-SNN conversion. In *International Joint Conference on Artificial Intelligence*, 2022a.

Ziming Wang, Shuang Lian, Zhang Yuhao, et al. Towards lossless ANN-SNN conversion under ultra-low latency with dual-phase optimization. *arXiv preprint arXiv:2205.07473*, 2022b.

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

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

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

Friedemann Zenke and Tim P Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021.

Wenrui Zhang and Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks. In *Advances in Neural Information Processing Systems*, pp. 12022–12033, 2020.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *AAAI Conference on Artificial Intelligence*, pp. 11062–11070, 2021.

Shibo Zhou, Xiaohua Li, Ying Chen, Sanjeev T Chandrasekaran, and Arindam Sanyal. Temporal-coded deep spiking neural network with easy training and robust performance. In *AAAI Conference on Artificial Intelligence*, pp. 11143–11151, 2021.

A APPENDIX

A.1 THE NETWORK CONFIGURATION IN THE TRAINING PROCEDURE

We choose Stochastic Gradient Descent optimizer (Bottou, 2012) and Cosine Annealing scheduler (Loshchilov & Hutter, 2017) to train ANN models for 300 epochs. For CIFAR-10/100, the value of weight decay is set to 5×10^{-4}, and the initial learning rates are 0.1 and 0.02, respectively. For ImageNet, we set the initial learning rate as 0.1 and weight decay as 1×10^{-4}. In addition, we adopt data-augmentation techniques (DeVries & Taylor, 2017; Cubuk et al., 2019; Li et al., 2021) to further improve the performance of the models.

A.2 PROOF OF THEOREM

Theorem 1. *Supposing that an ANN with QCFS activation function (equation 7) is converted to an SNN with $L = T, \lambda^l = \theta^l, \boldsymbol{v}^l(0) = \theta^l/2$, and the inputs to the l-th layer of ANN and SNN are the same, that is, $\boldsymbol{a}^{l-1} = \boldsymbol{\phi}^{l-1}(T)$. Then for any i-th element of the l-th layer, we will have the following conclusions:*
(i) If $\phi_i^l(T) > 0$ and $v_i^l(T) < 0$, we will have $\phi_i^l(T) > a_i^l$ and $\psi_i^l < 0$.
(ii) If $\phi_i^l(T) < \theta^l$ and $v_i^l(T) \geqslant \theta^l$, we will have $\phi_i^l(T) < a_i^l$ and $\psi_i^l > 0$.

Proof. According to the preconditions and equation 5, we have:

$$\phi_i^l(T) = \frac{\sum_{t=1}^{T} I_i^l(t)}{T} - \frac{v_i^l(T) - \theta^l/2}{T}. \tag{S1}$$

If $\sum_{t=1}^{T} I_i^l(t) \in [-\theta^l/2, \theta^l T + \theta^l/2)$, based on the preconditions and equation 7, we get:

$$a_i^l = \frac{\theta^l}{T} \left\lfloor \frac{\sum_{t=1}^{T} I_i^l(t)}{\theta^l} + \frac{1}{2} \right\rfloor. \tag{S2}$$

When $\sum_{t=1}^{T} I_i^l(t) \in [k\theta^l - \theta^l/2, k\theta^l + \theta^l/2), k = 0, 1, ..., T$, from equation S2 we will have $a_i^l = k\theta^l/T$. For (i), by combining $v_i^l(T) < 0$ and equation S1 we will have:

$$\begin{aligned} \phi_i^l(T) &= \frac{\sum_{t=1}^{T} I_i^l(t)}{T} - \frac{v_i^l(T) - \theta^l/2}{T} \\ &> \sum_{t=1}^{T} I_i^l(t)/T + \theta^l/2T \\ &\geqslant k\theta^l/T = a_i^l. \end{aligned} \tag{S3}$$

When $\sum_{t=1}^{T} I_i^l(t) < -\theta^l/2, a_i^l = 0$, according to the precondition $\phi_i^l(T) > 0$, we have $\phi_i^l(T) > a_i^l$. In addition, if $\sum_{t=1}^{T} I_i^l(t) \geqslant \theta^l T + \theta^l/2$, according to $v_i^l(T) < 0$ and equation S1, $\phi_i^l(T) > \theta^l$, which is impossible. Therefore, we can derive that $\phi_i^l(T) > a_i^l, \psi_i^l = (a_i^l - \phi_i^l(T))T/\theta^l < 0$ and we have already proved (i).

For (ii), we also first consider $\sum_{t=1}^{T} I_i^l(t) \in [-\theta^l/2, \theta^l T + \theta^l/2)$. When $\sum_{t=1}^{T} I_i^l(t) \in [k\theta^l - \theta^l/2, k\theta^l + \theta^l/2), k = 0, 1, ..., T$, from equation S2 we will have $a_i^l = k\theta^l/T$, by combining $v_i^l(T) \geqslant \theta^l$ and equation S1, we will have:

$$\begin{aligned} \phi_i^l(T) &= \frac{\sum_{t=1}^{T} I_i^l(t)}{T} - \frac{v_i^l(T) - \theta^l/2}{T} \\ &\leqslant \sum_{t=1}^{T} I_i^l(t)/T - \theta^l/2T. \\ &< k\theta^l/T = a_i^l \end{aligned} \tag{S4}$$

When $\sum_{t=1}^{T} I_i^l(t) \geqslant \theta^l T + \theta^l/2, a_i^l = \theta^l$, according to the precondition $\phi_i^l(T) < \theta^l$, we have $\phi_i^l(T) < a_i^l$. In addition, if $\sum_{t=1}^{T} I_i^l(t) < -\theta^l/2$, according to $v_i^l(T) \geqslant \theta^l$ and equation S1, $\phi_i^l(T) < 0$, which

is impossible. Therefore, we can derive that $\phi_i^l(T) < a_i^l, \psi_i^l = (a_i^l - \phi_i^l(T))T/\theta^l > 0$ and we have proved (ii). □

Theorem 2. *If we use $s_i^l(t)$ and $\widetilde{s}_i^l(t)$ to denote the i-th element in the binary output of the l-th layer at time-step t before and after optimization, $v_i^l(0)$ and $\widetilde{v}_i^l(0)$ to represent the initial membrane potential before and after optimization, then $\forall \epsilon \in (0, \theta^l)$, we will have the following conclusions:*

(i) If we set $\widetilde{v}_i^l(0) = v_i^l(0) - \max\left(\theta^l, \min\left\{v_i^l(t)|s_i^l(t) = 1\right\} + \epsilon\right)$, then $\sum_{t=1}^{T} \widetilde{s}_i^l(t) = \sum_{t=1}^{T} s_i^l(t) - 1$.

(ii) If we set $\widetilde{v}_i^l(0) = v_i^l(0) + \max\left(\theta^l, \theta^l + \epsilon - \max\left\{v_i^l(t)|s_i^l(t) = 0\right\}\right)$, then $\sum_{t=1}^{T} \widetilde{s}_i^l(t) = \sum_{t=1}^{T} s_i^l(t) + 1$.

We use $m_i^l(t), \widetilde{m}_i^l(t)$ to represent the accumulative potential at time-step t before and after using optimization. Before the proof of theorem, we firstly introduce Lemma 1.

Lemma 1. *For situation (i) in Theorem 1, $\exists t \in [1, T]$, $\sum_{k=1}^{t} s_i^l(k) = \sum_{k=1}^{t} \widetilde{s}_i^l(k) + 1$. For situation (ii) in Theorem 1, $\exists t \in [1, T]$, $\sum_{k=1}^{t} s_i^l(k) = \sum_{k=1}^{t} \widetilde{s}_i^l(k) - 1$.*

Proof. For situation (i) in Theorem 1, we use t_o to denote the specific time when $v_i^l(t_o) = \min\left\{v_i^l(t)|s_i^l(t) = 1\right\} \wedge s_i^l(t_o) = 1$. $\forall t$, as we optimize SNNs layer by layer, we have the following equation:

$$m_i^l(t) = v_i^l(0) + \sum_{k=1}^{t} I_i^l(k) - \sum_{k=1}^{t-1} s_i^l(k)\theta^l, \tag{S5}$$

$$\widetilde{m}_i^l(t) = \widetilde{v}_i^l(0) + \sum_{k=1}^{t} I_i^l(k) - \sum_{k=1}^{t-1} \widetilde{s}_i^l(k)\theta^l. \tag{S6}$$

As $v_i^l(0) > \widetilde{v}_i^l(0)$ and we use the same input $\sum_{k=1}^{t} I_i^l(k)$ before and after optimization, when $\sum_{k=1}^{t-1} s_i^l(k)\theta^l = \sum_{k=1}^{t-1} \widetilde{s}_i^l(k)\theta^l$, we will have $m_i^l(t) > \widetilde{m}_i^l(t)$ and further derive $s_i^l(t) \geqslant \widetilde{s}_i^l(t)$, which means that $\forall t$, $\sum_{k=1}^{t} s_i^l(k)\theta^l \geqslant \sum_{k=1}^{t} \widetilde{s}_i^l(k)\theta^l$.

If $\exists t' \in [1, t_o)$, $\sum_{k=1}^{t'} s_i^l(k) = \sum_{k=1}^{t'} \widetilde{s}_i^l(k) + 1$, then we have already found a qualified time t'. If $\sum_{k=1}^{t_o-1} s_i^l(k) = \sum_{k=1}^{t_o-1} \widetilde{s}_i^l(k)$, we will have:

$$\begin{aligned} m_i^l(t_o) - \widetilde{m}_i^l(t_o) &= v_i^l(0) - \widetilde{v}_i^l(0) \\ &= \max\left(\theta^l, \min\left\{v_i^l(t)|s_i^l(t) = 1\right\} + \epsilon\right) \\ &= \max\left(\theta^l, v_i^l(t_o) + \epsilon\right). \end{aligned} \tag{S7}$$

As $m_i^l(t_o) = v_i^l(t_o) + \theta^l$, we will further have:

$$\begin{aligned} \widetilde{m}_i^l(t_o) &= m_i^l(t_o) - \max\left(\theta^l, v_i^l(t_o) + \epsilon\right) \\ &\leqslant m_i^l(t_o) - v_i^l(t_o) - \epsilon \\ &< \theta^l. \end{aligned} \tag{S8}$$

From the above equation, we can derive $\widetilde{m}_i^l(t_o) < \theta^l$ and $s_i^l(t_o) = 1, \widetilde{s}_i^l(t_o) = 0$, then we will have $\sum_{k=1}^{t_o} s_i^l(k) = \sum_{k=1}^{t_o} \widetilde{s}_i^l(k) + 1$, which means that t_o is a qualified time.

For situation (ii) in Theorem 1, we use t_o to denote the specific time when $v_i^l(t_o) = \max\{v_i^l(t)|s_i^l(t) = 0\} \wedge s_i^l(t_o) = 0$. Similarly, we will derive $\forall t, \sum_{k=1}^{t} s_i^l(k)\theta^l \leqslant \sum_{k=1}^{t} \widetilde{s}_i^l(k)\theta^l$ according to equation S5-equation S6.

If $\exists t' \in [1, t_o), \sum_{k=1}^{t'} s_i^l(k) = \sum_{k=1}^{t'} \widetilde{s}_i^l(k) - 1$, then we have already found a qualified time t'. If $\sum_{k=1}^{t_o-1} s_i^l(k) = \sum_{k=1}^{t_o-1} \widetilde{s}_i^l(k)$, we will have:

$$\begin{aligned}\widetilde{m}_i^l(t_o) - m_i^l(t_o) &= \widetilde{v}_i^l(0) - v_i^l(0) \\ &= \max\left(\theta^l, \theta^l + \epsilon - \max\{v_i^l(t)|s_i^l(t) = 0\}\right) \\ &= \max\left(\theta^l, \theta^l + \epsilon - v_i^l(t_o)\right). \end{aligned} \tag{S9}$$

As $m_i^l(t_o) = v_i^l(t_o)$, we will further have:

$$\begin{aligned}\widetilde{m}_i^l(t_o) &= m_i^l(t_o) + \max\left(\theta^l, \theta^l + \epsilon - v_i^l(t_o)\right) \\ &\geqslant m_i^l(t_o) + \theta^l + \epsilon - v_i^l(t_o) \\ &> \theta^l. \end{aligned} \tag{S10}$$

From the above equation, we can derive $\widetilde{m}_i^l(t_o) > \theta^l$ and $s_i^l(t_o) = 0, \widetilde{s}_i^l(t_o) = 1$, then we will have $\sum_{k=1}^{t_o} s_i^l(k) = \sum_{k=1}^{t_o} \widetilde{s}_i^l(k) - 1$, which means that t_o is a qualified time. □

Now we will further prove Theorem 1.

Proof. **Proof of (i).** If $\theta^l > \min\{v_i^l(t)|s_i^l(t) = 1\} + \epsilon$, $\widetilde{v}_i^l(0) = v_i^l(0) - \theta^l$. According to Lemma 1, $\exists t_s, \sum_{k=1}^{t_s} s_i^l(k) = \sum_{k=1}^{t_s} \widetilde{s}_i^l(k) + 1$, then we will have $m_i^l(t_s+1) = \widetilde{m}_i^l(t_s+1)$ by combining equation S5 and equation S6, which means that $\sum_{k=t_s+1}^{T} s_i^l(k) = \sum_{k=t_s+1}^{T} \widetilde{s}_i^l(k)$ in the remaining time cycle. As a result, we can have $\sum_{k=1}^{T} s_i^l(k) = \sum_{k=1}^{T} \widetilde{s}_i^l(k) + 1$.

If $\theta^l < \min\{v_i^l(t)|s_i^l(t) = 1\} + \epsilon$, $\widetilde{v}_i^l(0) = v_i^l(0) - \min\{v_i^l(t)|s_i^l(t) = 1\} - \epsilon$. According to Lemma 1, $\exists t_s, \sum_{k=1}^{t_s} s_i^l(k) = \sum_{k=1}^{t_s} \widetilde{s}_i^l(k) + 1$, then we will have $m_i^l(t_s+1) = \widetilde{m}_i^l(t_s+1) + \min\{v_i^l(t)|s_i^l(t) = 1\} + \epsilon - \theta^l$, which means that $m_i^l(t_s+1) > \widetilde{m}_i^l(t_s+1)$.

For m_i^l, if we set t' as the first spike firing time after t_s, which means that $m_i^l(t') = v_i^l(t') + \theta^l$ and $\sum_{k=1}^{t'-1} s_i^l(k) = \sum_{k=1}^{t'-1} \widetilde{s}_i^l(k) + 1$, then we will have $\widetilde{m}_i^l(t') = m_i^l(t') - \min\{v_i^l(t)|s_i^l(t) = 1\} - \epsilon + \theta^l = v_i^l(t') - \min\{v_i^l(t)|s_i^l(t) = 1\} - \epsilon + 2\theta^l > \theta^l$. which means that $s_i^l(t') = \widetilde{s}_i^l(t') = 1$, $\sum_{k=1}^{t'} s_i^l(k) = \sum_{k=1}^{t'} \widetilde{s}_i^l(k) + 1$. If we continue to use the above derivation process, we can finally have $\sum_{k=1}^{T} s_i^l(k) = \sum_{k=1}^{T} \widetilde{s}_i^l(k) + 1$.

Proof of (ii). If $\theta^l > \theta^l + \epsilon - \max\{v_i^l(t)|s_i^l(t) = 0\}$, $\widetilde{v}_i^l(0) = v_i^l(0) + \theta^l$. According to Lemma 1, $\exists t_s, \sum_{k=1}^{t_s} s_i^l(k) = \sum_{k=1}^{t_s} \widetilde{s}_i^l(k) - 1$, then we will have $m_i^l(t_s+1) = \widetilde{m}_i^l(t_s+1)$ by combining equation S5 and equation S6, which means that $\sum_{k=t_s+1}^{T} s_i^l(k) = \sum_{k=t_s+1}^{T} \widetilde{s}_i^l(k)$ in the remaining time cycle. As a result, we can have $\sum_{k=1}^{T} s_i^l(k) = \sum_{k=1}^{T} \widetilde{s}_i^l(k) - 1$.

If $\theta^l < \theta^l + \epsilon - \max\{v_i^l(t)|s_i^l(t)=0\}$, $\widetilde{v}_i^l(0) = v_i^l(0) + \theta^l + \epsilon - \max\{v_i^l(t)|s_i^l(t)=0\}$. According to Lemma 1, $\exists t_s$, $\sum_{k=1}^{t_s} s_i^l(k) = \sum_{k=1}^{t_s} \widetilde{s}_i^l(k) - 1$, then we will have $m_i^l(t_s+1) = \widetilde{m}_i^l(t_s+1) - \epsilon + \max\{v_i^l(t)|s_i^l(t)=0\}$, which means that $m_i^l(t_s+1) < \widetilde{m}_i^l(t_s+1)$.

For $\widetilde{m}_i^l$, if we set t' as the first spike firing time after t_s, which means that $\widetilde{m}_i^l(t') = \widetilde{v}_i^l(t') + \theta^l$ and $\sum_{k=1}^{t'-1} s_i^l(k) = \sum_{k=1}^{t'-1} \widetilde{s}_i^l(k) - 1$. Similar to situation (i), we need to prove that $s_i^l(t') = \widetilde{s}_i^l(t') = 1$. However, it is not easy to make a direct proof. Therefore, we attempt to prove its inverse and negative thesis : when $t' > t_s \wedge \sum_{k=1}^{t'-1} s_i^l(k) = \sum_{k=1}^{t'-1} \widetilde{s}_i^l(k) - 1$, if $s_i^l(t') = 0$, then we can have $\widetilde{s}_i^l(t') = 0$.

Under this condition, we can derive $m_i^l(t') = v_i^l(t') \wedge m_i^l(t') = \widetilde{m}_i^l(t') - \epsilon + \max\{v_i^l(t)|s_i^l(t)=0\}$, then we will have $\widetilde{m}_i^l(t') = v_i^l(t') + \epsilon - \max\{v_i^l(t)|s_i^l(t)=0\}$. As $\max\{v_i^l(t)|s_i^l(t)=0\} \geqslant v_i^l(t')$, $\widetilde{m}_i^l(t') \leqslant \epsilon < \theta^l$, which means that $\widetilde{s}_i^l(t') = 0$.

Therefore, if we set t' as the first spike firing time for $\widetilde{m}_i^l$ after t_s, we can prove that $s_i^l(t') = \widetilde{s}_i^l(t') = 1$, which means that $\sum_{k=1}^{t'} s_i^l(k) = \sum_{k=1}^{t'} \widetilde{s}_i^l(k) - 1$. If we continue to use the above derivation process, we can finally have $\sum_{k=1}^{T} s_i^l(k) = \sum_{k=1}^{T} \widetilde{s}_i^l(k) - 1$. □

A.3 Experimental results on CIFAR-10 dataset

For VGG-16, the accuracy of our proposed method is 95.46% with 4 time-step ($\rho = 4$), whereas the accuracies of OPI and QCFS are 90.96% and 94.95% with 8 time-step, respectively. For ResNet-18, we achieve 95.46% with 4 time-steps ($\rho = 4$), whereas the corresponding performance of OPI and QCFS are 75.44% and 95.04%. For ResNet-20, our method reaches 91.68% with 4 time-steps ($\rho = 4$), which is 2.13% higher than QCFS (89.55%, T=8) and 25.44% higher than OPI (66.24%, T=8).

Table S1: Comparison with other ANN-SNN conversion methods on CIFAR-10 dataset

Method	ANN	Architecture	T=1	T=2	T=4	T=8	T=16	T=32
SNM	94.09%	VGG-16	-	-	-	-	-	93.43%
SNNC-AP	95.72%		-	-	-	-	-	93.71%
OPI	94.57%		-	-	-	90.96%	93.38%	94.20%
QCFS	95.52%		-	91.18%	93.96%	94.95%	95.40%	95.54%
Ours	95.51%		94.90%	95.36%	95.46%	95.51%	95.57%	95.61%
SNM	95.39%	ResNet-18	-	-	-	-	-	94.03%
SNNC-AP	95.46%		-	-	-	-	-	94.78%
OPI	96.04%		-	-	-	75.44%	90.43%	94.82%
QCFS	95.64%		-	91.75%	93.83%	95.04%	95.56%	95.67%
Ours	95.64%		95.25%	95.45%	95.46%	95.66%	95.68%	95.68%
OPI	92.74%	ResNet-20	-	-	-	66.24%	87.22%	91.88%
QCFS	91.77%		-	73.20%	83.75%	89.55%	91.62%	92.24%
Ours	91.77%		89.88%	91.26%	91.68%	91.86%	92.20%	92.16%

A.4 Eliminating offset spike through iterative optimization

In Sections 4.4 and 5.4, we have pointed out the iterative property of our proposed method. Here we will make a discussion in detail. Firstly, we can infer the specific value of ψ^l based on the residual membrane potential when the corresponding input current belongs to a specific interval, which is illustrated in the following theorem.

Theorem 3. *Supposing that an ANN with QCFS activation function (equation 7) is converted to an SNN with $L = T, \lambda^l = \theta^l, \boldsymbol{v}^l(0) = \theta^l/2$, and the inputs to the l-th layer of ANN and SNN are the same, that is, $\boldsymbol{a}^{l-1} = \boldsymbol{\phi}^{l-1}(T)$. Then for any i-th element of the l-th layer, we will have the following conclusions:*
If $\sum_{t=1}^{T} I_i^l(t) \in [-\theta^l/2, \theta^l T + \theta^l/2)$, when $v_i^l(T)/\theta^l \in [k, k+1)$, we will have $\psi_i^l = a_i^l T/\theta^l - \sum_{t=1}^{T} s_i^l(t) = k$, where $k \in \mathbb{Z}$.

Proof. As the preconditions of Theorem 3 are same as the preconditions of equation S1 and equation S2, by combining equation S1 and equation S2, we will have:

$$\begin{aligned} a_i^l T/\theta^l - \sum_{t=1}^{T} s_i^l(t) &= \left\lfloor \frac{\sum_{t=1}^{T} I_i^l(t)}{\theta^l} + \frac{1}{2} \right\rfloor - \frac{T}{\theta^l}\left(\frac{\sum_{t=1}^{T} I_i^l(t)}{T} - \frac{v_i^l(T) - \theta^l/2}{T}\right) \\ &= v_i^l(T)/\theta^l + \left\lfloor \sum_{t=1}^{T} I_i^l(t)/\theta^l + 1/2 \right\rfloor - \left(\sum_{t=1}^{T} I_i^l(t)/\theta^l + 1/2 \right). \end{aligned} \tag{S11}$$

As $-1 < \left\lfloor \sum_{t=1}^{T} I_i^l(t)/\theta^l + 1/2 \right\rfloor - \left(\sum_{t=1}^{T} I_i^l(t)/\theta^l + 1/2 \right) \leqslant 0$, when $v_i^l(T)/\theta^l \in [k, k+1)$, $k - 1 < a_i^l T/\theta^l - \sum_{t=1}^{T} s_i^l(t) < k + 1$. Considering that $a_i^l T/\theta^l - \sum_{t=1}^{T} s_i^l(t) \in \mathbb{Z}$, we have $\psi_i^l = a_i^l T/\theta^l - \sum_{t=1}^{T} s_i^l(t) = k$. □

In fact, even if the input current does not belong to the specific interval, from equation 7, we can derive that when $\sum_{t=1}^{T} I^l(t) < -\theta^l/2, a^l = 0$ and when $\sum_{t=1}^{T} I^l(t) \geqslant \theta^l T + \theta^l/2, a^l = \theta^l$, then we can directly determine the $\boldsymbol{\psi}^l$ according to the value of $\phi^l(T)$, although this may require more memory resource and energy consumption. After we have already acquired the value of $\boldsymbol{\psi}^l$, we will adopt our optimization method for $|\psi_i^l|$ times to eliminate the offset spike on i-th element neuron of the l-th layer.