MSAT: Biologically Inspired Multi-Stage Adaptive Threshold for Conversion of Spiking Neural Networks[⋆]

Xiang He[a,b,c], Yang Li[a,b,c], Dongcheng Zhao[a,c], Qingqun Kong[a,b,c,*], Yi Zeng[a,b,c,d,*]

[a]*Institute of Automation, Chinese Academy of Sciences (CAS), BeiJing, China*
[b]*School of Artificial Intelligence, University of Chinese Academy of Sciences, BeiJing, China*
[c]*Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, BeiJing, China*
[d]*Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China*

Abstract

Spiking Neural Networks (SNNs) can do inference with low power consumption due to their spike sparsity. ANN-SNN conversion is an efficient way to achieve deep SNNs by converting well-trained Artificial Neural Networks (ANNs). However, the existing methods commonly use constant threshold for conversion, which prevents neurons from rapidly delivering spikes to deeper layers and causes high time delay. In addition, the same response for different inputs may result in information loss during the information transmission. Inspired by the biological model mechanism, we propose a multi-stage adaptive threshold (MSAT). Specifically, for each neuron, the dynamic threshold varies with firing history and input properties and is positively correlated with the average membrane potential and negatively correlated with the rate of depolarization. The self-adaptation to membrane potential and input allows a timely adjustment of the threshold to fire spike faster and transmit more information. Moreover, we analyze the Spikes of Inactivated Neurons error which is pervasive in early time steps and propose spike confidence accordingly as a measurement of confidence

---

[⋆]Xiang He and Yang Li contributed equally to this work.
[*]Corresponding author
  Email addresses: `hexiang2021@ia.ac.cn` (Xiang He), `liyang2019@ia.ac.cn` (Yang Li), `zhaodongcheng2016@ia.ac.cn` (Dongcheng Zhao), `qingqun.kong@ia.ac.cn` (Qingqun Kong), `yi.zeng@ia.ac.cn` (Yi Zeng)

about the neurons that correctly deliver spikes. We use such spike confidence in early time steps to determine whether to elicit spike to alleviate this error. Combined with the proposed method, we examine the performance on non-trivial datasets CIFAR-10, CIFAR-100, and ImageNet. We also conduct sentiment classification and speech recognition experiments on the IDBM and Google speech commands datasets respectively. Experiments show near-lossless and lower latency ANN-SNN conversion. To the best of our knowledge, this is the first time to build a biologically inspired multi-stage adaptive threshold for converted SNN, with comparable performance to state-of-the-art methods while improving energy efficiency.

1. Introduction

Artificial Neural Networks (ANNs) have been widely used in speech recognition, image processing, and other fields. However, with network structures becoming more complex, they often require large amounts of computational resources. In addition, the current ANNs computing paradigm differs from the human brain's mechanism, which communicates by transmitting spike trains generated by action potentials. Spiking Neuron networks (SNNs) work similarly to the brains and transmit the spike sequence to the downstream neurons. The information of spikes is enormous, and their distribution is sparse, so SNNs could demonstrate low power consumption characteristics.

SNNs have the potential for efficient inference when combined with neuromorphic hardware. Furthermore, they inherently show efficiency in processing temporal and spatial data. Their diverse coding mechanisms and event-driven characteristics are also promising. However, because the transmitted spikes are not differentiable, it is challenging to train SNNs. To solve this problem, some algorithms based on surrogate gradients (SG) (Lee et al., 2016; Wu et al., 2018, 2019; Zhang & Li, 2020; Shen et al., 2022) and spike-timing-dependent plastic-

ity (STDP) (Caporale et al., 2008; Diehl & Cook, 2015; Hao et al., 2020; Zhao et al., 2020) have been proposed. However, it is still difficult to train deeper SNNs with complex network structures from scratch, resulting in a massive gap in performance between SNNs and ANNs in recognition and detection tasks.

To bridge the performance gap between SNNs and ANNs, methods of converting well-trained ANNs to SNNs with the same structure have been proposed. The fundamental principle supporting such conversion is that firing rates of spiking neurons could approximate their counterparts activation (ReLU) in ANNs with sufficient time steps (Diehl et al., 2015).

In most existing conversion schemes, the thresholds of neurons in the same layer are identical and equal to the maximum activation value of that layer in the ANN. For this mechanism, spatially, the neuron needs to accumulate more membrane potential to exceed a pre-set threshold to deliver the spike, which makes the deeper neuron need to wait longer to receive the spike from the prior layer, and temporally, the neuron delivers the same information when firing the spike at different time steps, regardless of how much the membrane potential exceeds the threshold, causing an information loss. While in the biological counterpart of the spiking neuron, the spiking thresholds exhibit large variability, which has a significant impact on the encoding of inputs to spiking neurons (Fontaine et al., 2014). The threshold value is not fixed even for the same neuron and threshold variability can be considered as an inherent characteristic of biological neurons, which exhibits self-adaptation to the membrane potential over a short time scale (Azouz & Gray, 2000; Henze & Buzsáki, 2001; Azouz & Gray, 2003; Pena & Konishi, 2002; Wilent & Contreras, 2005). Threshold variability also enhances coincidence detection properties of cortical neurons (Azouz & Gray, 2000; Wilent & Contreras, 2005) as mentioned in (Fontaine et al., 2014). Therefore, from the biological plausibility perspective, the different threshold across neurons and time steps is genuine feature of neurons and has a profound impact on the input coding.

Motivated by the above biological model mechanisms, we propose a multistage adaptive threshold (MSAT) for ANN-SNN conversion to effectively take

advantage of the spatial information brought by different neurons in the same layer and the temporal information brought by the accumulation of membrane potentials over time steps. For each neuron, the dynamic threshold is related to the firing history, i.e., positively correlated with the preceding membrane potential, and to the input characteristics, i.e., negatively correlated with the rate of depolarization. In addition, we analyze the conversion error layer-to-layer and show that the Spikes of Inactivated Neurons (SIN), i.e., the misfired spikes which are due to the transient characteristics of the spikes, are pervasive in the early time steps and cause degradation of accuracy. According to statistical analysis of the error introduced by SIN in each layer, we propose spike confidence to measure the confidence about the correct neurons rather than misfired neurons and use spike confidence in early time steps to determine whether to elicit spike or not. We conduct experiments on standard object classification benchmarks and non-visual domains such as natural language processing and speech to validate the effectiveness and universality of the proposed method.

Our major contributions can be summarized as follows:

- We use the widely existing mechanism in the nervous system for reference and propose multi-stage adaptive threshold (MSAT) for converted SNNs for the first time. Different from constant threshold, self-adaptation to input value and membrane potential reduces latency and information loss.

- We formulate the layer-wise conversion error and derive the Spikes of Inactivated Neurons error. Accordingly, we propose spike confidence to alleviate SIN error. Each spike is given a confidence according to the statistical SIN error, and the spike confidence determines whether to fire a spike rather than fire directly.

- We conduct comprehensive experiments and ablation studies to show that the multi-stage adaptive threshold and spike confidence measurement facilitate faster transmission of information and reduce information loss in spike trains. Experimental results show that our proposed method has a comparable performance with SOTA methods but consumes less energy.

4

2. Related work

The conversion methods use the well-trained ANN and map it to an equivalent SNN. The studies begin with (Pérez-Carrasco et al., 2013) and the main idea that the mean firing rate in Integrate-and-Fire (IF) neuron (Burkitt, 2006) can approximate ReLU activation value is proposed by (Diehl et al., 2015). A mathematical statement of the feasibility of conversion is expressed firstly in (Rueckauer et al., 2017). Since (Sengupta et al., 2019), the converted SNNs begin going deeper and doing classification tasks in larger datasets. However, performance degradation is still the main problem. (Han et al., 2020) propose a more efficient reset method named soft-reset, which uses a reset by subtraction mechanism. (Rathi & Roy, 2020) first use threshold optimization in deep SNN. Although the performance of the converted SNN is improved, the cost of large time delays also comes. (Deng & Gu, 2021) divide conversion into floor and clip error from a new quantization perspective, (Li et al., 2021a) further optimize the conversion error. (Yu et al., 2021) construct the deep SNN with double-threshold, (Liu et al., 2022) propose temporal separation to further zip gap between ANN and SNN. (Li & Zeng, 2022) use burst mechanism and propose LIPooling to solve the conversion error caused by the MaxPooling layer. (Wang et al., 2022) present a dual-phase converting algorithm to relieve clip and floor error. (Bu et al., 2021, 2022) consider membrane potential initialization and quantization of the activation function to obtain faster SNN. Nonetheless, most previous works have been limited to the fixed threshold and ignored utilizing the individual neurons to transmit information effectively. Some studies (Kim et al., 2020; Li et al., 2021a) take threshold variation into consideration while they still use the two-stage or heuristic method for SNNs, which require careful design or extensive search. (Ding et al., 2022) use the biologically dynamic threshold mechanism to do robot obstacle avoidance and continuous control tasks and enhance the host SNN generalization. In all, a bio-plausible multi-stage threshold for deep spiking neural networks has not been explored yet.

Symbol	Definition	Symbol	Definitionp
l	Layer index	M^l	Neuron numbers in layer l
i	Neuron index	W	Weight connection
t	Current time step	a	ANN activation value
T	Total time step	$\mathbf{V}_i^l(t)$	Membrane potential before firing
V_{th}	Constant threshold	$V_i^l(t)$	Membrane potential after firing
$V_{th}(t)$	Dynamic threshold	p	Spike confidence
$\Theta_{t,i}^l$	Step function indicating spike at t	e	conversion error in total time step
c	Spike filter	E	Early time step

Table 1: Summary of notations in this paper.

3. Conversion errors analysis

In this section, we first review the neuron models of ANNs and SNNs. Then we analyze the conversion errors with constant threshold and adaptive threshold separately. The symbols used in this paper are summarized in Table 1.

3.1. Neuron model

In ANN, the activation value a_i^l (after ReLU) of neuron i in layer l can be computed as

$$a_i^l = \max \left(0, \sum_{j=1}^{M^{l-1}} W_{ij}^l a_j^{l-1} + b_i^l \right), \tag{1}$$

where $l \in \{1, \cdots, L\}$ indicates layer l in a network with L layers; W_{ij}^l is the weight connection between neuron i in layer l and neuron j in layer $l-1$; b_i^l means neuron i bias in layer l and is constant all the time; we omit the bias for convenience in the following description. The number of neurons in layer l is M^l, and activation value a_i^l starts from $l=0$ and $a^0 = x$ for the direct input.

The membrane potential before firing at time step t, $\mathbf{V}_i^l(t)$ is a sum of the last time step $t-1$ membrane potential and current input. When $\mathbf{V}_i^l(t)$ exceeds a certain voltage threshold, it emits an output spike and resets the membrane potential. One of the most widely adopted models is Integrate-and-Fire (IF)

neuron, and the membrane potential would then be updated by soft-reset mech-
anism, which subtracts the threshold in $\mathbf{V}_i^l(t)$ rather than reset the membrane
potential to V_{reset}. We use $V_i^l(t)$ to denote membrane potential after firing, then
the mathematical form is as follows

$$V_i^l(t) = V_i^l(t-1) + V_{th}^{l-1} \sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,j}^{l-1} - V_{th}^l \Theta_{t,i}^l, \tag{2}$$

where $\Theta_{t,i}^l$ is a function indicating the neuron i in layer l elicits a spike at time
t. Here we use $\Theta_{t,i}^l$ to replace $\Theta_i^l(t)$ for simplicity.

$$\Theta_{t,i}^l = \mathcal{H}\left(V_i^l(t-1) + z_i^l(t) - V_{th}^l\right),$$

$$\mathcal{H}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{else.} \end{cases} \tag{3}$$

Here $z_i^l(t)$ is the input of neuron i in layer l and time t:

$$z_i^l(t) = V_{th}^{l-1} \sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,j}^{l-1}. \tag{4}$$

3.2. Constant threshold conversion error

The main idea in ANN-SNN conversion is using mean firing rate $r_i^l(t)$, which
indicates the firing rate of neuron i in layer l for a total time t, to approximate
the activation value a_i^l. Here we give an analytical explanation for the approxi-
mation process.

The conversion error comes from two-part: converting ANN to SNN directly,
resulting in quantization error and clip error $e_{i,QC}^l$; the other is transient charac-
teristic of neurons and irregular elicited spike, resulting in Spikes of Inactivated
Neurons (Li et al., 2021b) error $e_{i,SIN}^l$, as shown in Fig. 1. In the subsequent
error analysis, we assume that the firing rate of the preceding layer fully repre-
sents the activation value, which means $r_i^{l-1} = a_i^{l-1}$. Then the conversion error
in layer l could be the following equation:

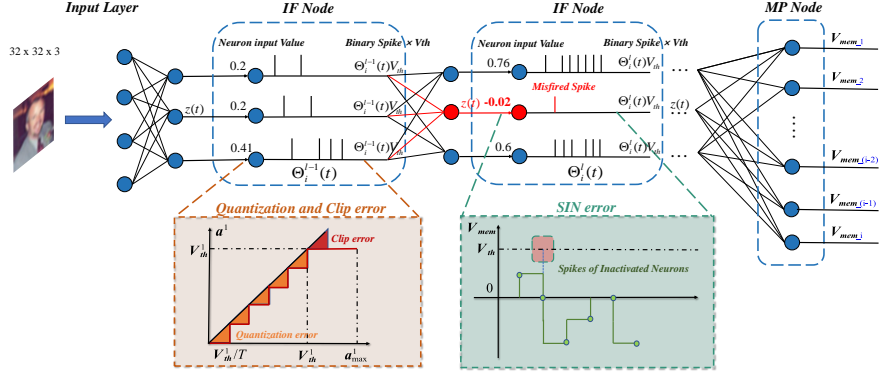$$e_i^l = r_i^l - a_i^l = e_{i,QC}^l + e_{i,SIN}^l. \tag{5}$$

Figure 1: A multilayer perceptron spiking neural network within 10 time steps for demonstrating conversion error. The red box shows the quantization error and clip error, caused by the discrete and insufficient time steps. The green box shows the Spikes of Inactivated Neurons error, caused by the dynamic transients of the neurons.

3.2.1. quantization and clip Error

For Eq. 2, accumulating the input over the simulation time step T, we can derive the firing rate $r_i^l(T)$ relationship layer by layer,

$$r_i^l(T) = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}(T) - \frac{V_i^l(T)}{T}. \tag{6}$$

Here $r_i^l(T) = V_{th}^l \Theta_{t,i}^l / T$. When V_{th}^l is larger than the maximum activation value, $V_i^l(T)$ will be less than V_{th}^l. Thus, the residual membrane potential cannot be output, that is why information transmission suffers a loss. Because of the discreteness of the time steps, the mean firing rate can be described as a step floor function and cannot precisely approximate the continuous ReLU function, known as quantization error or flooring error. For example, as shown in Fig. 1, activation value 0.42 should be accurately represented by 42 spikes within 100 time steps, while 4.2 spikes cannot be achieved in discrete 10 time steps because the number of spikes should be an integer.

If the voltage threshold is set smaller than the maximum activation value, then when the $\mathbf{V}_i^l(t)$ exceeds the threshold, the emitted spike will not transmit

8

effective information to distinguish membrane potential above the threshold, which is known as clip error. Setting voltage threshold to maximum activation value can avoid this but suffers a notable latency.

Considering the clip operation, the actual firing rate can be expressed as

$$r_i^l(T) = \text{clip}\left(\frac{V_{th}^l}{T}\left\lfloor\frac{\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} T}{V_{th}^l}\right\rfloor, 0, V_{th}^l\right), \tag{7}$$

where r_j^l means $r_j^l(T)$ for simplifying the statement and $\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}$ is a_i^l. Eq. 7 shows that even assuming the same inputs in the previous layer, the average firing rate of the SNN is still differ from the activation values in the ANN due to discrete quantization and clip, thus generating quantization and clip errors $e_{i,QC}^l$.

3.2.2. Spikes of Inactivated Neurons error

Inactivated Neurons refer to the neurons whose activation value counterparts in ANN are negative. Theoretically, they should not fire spike in all time steps to achieve ReLU filtering the negative value. However, once they elicit spikes, the corresponding mean firing rate r_i^l will be larger than zero. As shown in Fig. 1, the adjacent weights of the red neuron are respectively 0.5, 0.5, and -0.5, then the simulation activation value is -0.01, meaning that no spike should have been fired. However, a spike fired by mistake results in SIN error.

We use $\mathcal{R} = \left\{j\,\middle|\,\sum_{t=0}^T \Theta_{t,j}^l > 0,\ a_j^l(t) < 0\right\}$ to denote neurons with SIN, then SIN error can be expressed as

$$e_{i,SIN}^l = \mathbf{0} - \frac{\sum_{j\in\mathcal{R}}\sum_{t=0}^T W_{ij}^l \Theta_{t,j}^l}{T}. \tag{8}$$

By observing outputs of each neuron at different time steps, we find that the neurons with SIN usually fire at an early stage and then keep silent at a later stage during the conversion process. We count neurons with SIN proportion in each layer, as shown in Fig. 2. It shows that neurons responsible for SIN are prevalent and take the larger proportion in the deeper layer. Moreover, it is clear that most SIN appear in the early time steps. Since SIN mean undesired error

9

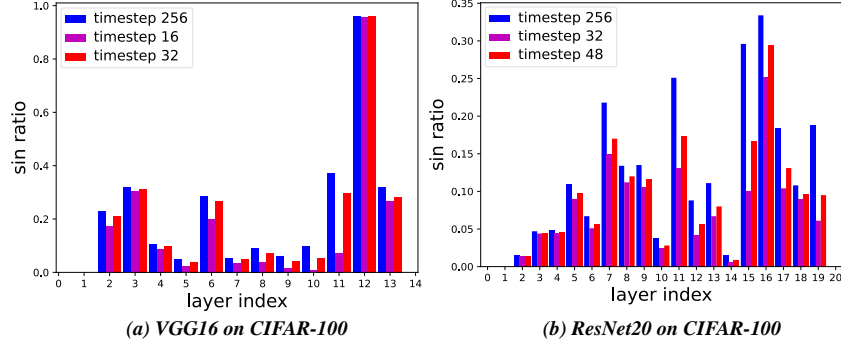*(a) VGG16 on CIFAR-100* *(b) ResNet20 on CIFAR-100*

Figure 2: Neurons with SIN ratio in each layer of VGG16 and ResNet20 in CIFAR-100.

existed in spike, thus SIN lead to inaccurate coding of signals and degradation of the accuracy.

3.3. Adaptive threshold optimization error

In this part we demonstrate that when the threshold is a function of time step t, spike trains are able to transmit the equivalent or more efficient information than the constant threshold.

With the threshold that changes over time steps, Eq. 2 can be rewritten in the following form:

$$V_i^l(t) = V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^{l-1}(t) W_{ij}^l \Theta_{t,j}^{l-1} - V_{th,i}^l(t) \Theta_{t,i}^l. \tag{9}$$

The firing rate during time step T, $r_i^l(T)$, is computed as $\sum_{t'=1}^{T} V_{th,i}^l(t') \Theta_{t',i}^l / T$, which means Eq. 6, the firing rate relationship in the higher layer still exists. On this basis, the residual neuron membrane potential $V_i^l(T)$ can be appropriately adjusted, so the spike information could be more efficient and thus shorten

10

the conversion latency. The optimization target in threshold adaptation is

$$\min_{V_{th,i}^l(T)} \left(\mathbf{o} - \mathbf{o}' \right),$$

$$\mathbf{o} = \text{clipfloor} \left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}, T, V_{th,i}^l(T) \right) \tag{10}$$

$$- \frac{\sum_{j \in \mathcal{R}} \sum_{t=0}^T W_{ij}^l \Theta_{t,j}^l}{T},$$

$$\mathbf{o}' = \text{ReLU} \left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} \right).$$

There is no closed-form solution to the above problem, Li et al. (2021a) use gird search to heuristically find the final solution. A trivial solution is to make $V_{th,i}^l = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}$, which means the voltage threshold equals to input value for each neuron. With this solution, ANN can be converted to SNN only one time step. However, such SNN elicits spike every time step and lose spike sparsity, so it is unreasonable and makes no sense, thus should not be used. In the next section, we will give a biologically rational method.

## 4. Methods

4.1. Multi-Stage Adaptive Threshold

Based on the above analysis, the adaptive threshold, which is multi-stage and varies with inference time, could better facilitate information from individual neurons and different time steps. For each neuron, the adaptive threshold varies with firing history and input properties. The adaptive threshold at time step $t + 1$, $V_{th,i}^l(t + 1)$ can be described as

$$V_{th,i}^l(t + 1) = \tau_{mp} V_{th\_mp,i}^l(t) + \tau_{rd} V_{th\_rd,i}^l(t + 1), \tag{11}$$

where τ_{mp} and τ_{rd} are the time constant of the dynamic tracking threshold $V_{th\_mp,i}^l(t)$ and dynamic evoked threshold $V_{th\_rd,i}^l(t + 1)$ separately. Specifically, the dynamic tracking threshold is positively correlated with the average
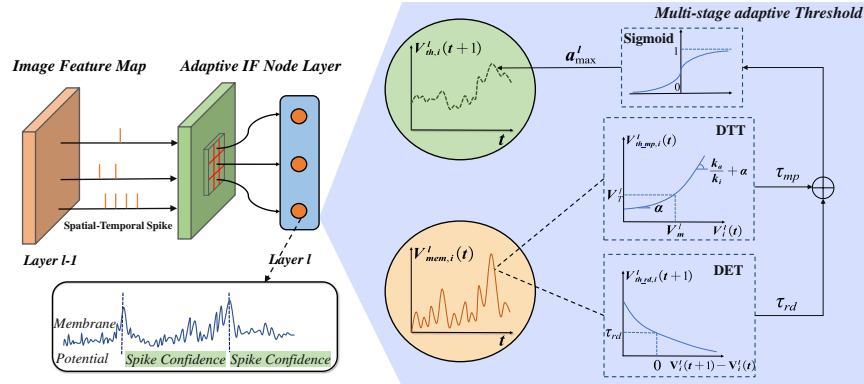
(a) MSAT schematic diagram
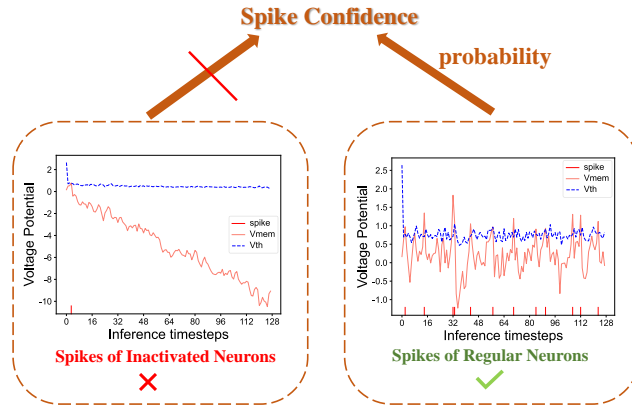
(b) Spike confidence

Figure 3: Summary of MSAT and spike confidence. Spike confidence is only implemented in the early stages.

preceding membrane potential and the dynamic evoked threshold is negatively correlated with the rate of depolarization.

Fig. 3(a) gives the multi-stage adaptive threshold schematic diagram: when a neuron receives current input, its threshold will take average membrane potential and rate of depolarization into consideration, then the threshold will vary with these two factors. Finally, it gives a changed threshold as the adaptation to membrane potential and rate of depolarization.

4.1.1. Dynamic Tracking Threshold

Dynamic Tracking Threshold (DTT) is a reflection of spiking threshold that varies with firing history. In Fontaine et al. (2014), the DTT is a similar first-order kinetic equation; here we use the steady-state threshold for fitting our SNNs. Let us use $V_{m,i}^l(t)$ to denote the average membrane potential at time step t in layer l neuron i, then DTT can be described as follows

$$V_{th\_mp,i}^l(t) = \alpha \left(V_i^l(t) - V_{m,i}^l(t) \right) + V_T^l + k_a ln \left(1 + e^{\frac{V_i^l(t) - V_{m,i}^l(t)}{k_i}} \right), \qquad (12)$$

where α, k_a, k_i are all time constant, V_T^l is the parameter to optimize. When residual membrane potential $V_i^l(t)$ is less than the average membrane potential $V_{m,i}^l(t)$, the slope of the curve around the inflection point is α and $\frac{k_a}{k_i} + \alpha$ respectively. $\alpha, k_a, k_i, V_T^l, V_i^l(t)$, these parameters together determine the curvature.

Eq. 12 shows that when the residual membrane potential increases, the threshold correspondingly increases and vice versa. There are several benefits that come along with such DTT scheme; for example, small voltage fluctuation generated by a small current input will have no effect on the output spike if it is smaller than the amplitude of the threshold adaption. In the early spike unstable phase, the neuron receives the synaptic current and adjusts the threshold according to the residual membrane potential, and if it receives a positive total input current but has not yet delivered a spike, the threshold is raised according to Eq. 12, thus alleviating the Spikes of Inactivated Neurons error.

4.1.2. Dynamic Evoked Threshold

Dynamic Evoked Threshold (DET) is a reflection of spiking threshold that varies with input properties, Azouz & Gray (2000) shows voltage threshold varies inversely with the preceding rate of depolarization dV_m/dt by plotting scatter. In IF neurons, we use variation of membrane potential before firing to express the preceding rate of depolarization thus DET can be expressed as

$$V_{th\_rd,i}^l(t+1) = \tau_{rd} e^{-\frac{\left(\mathbf{V}_i^l(t+1) - \mathbf{V}_i^l(t) \right)}{C}}. \qquad (13)$$

As mentioned before, here $\mathbf{V}_i^l(t)$ denotes membrane potential before firing

and makes a distinction with $V_i^l(t)$, which denote residual membrane potential after firing. C is the time constant representing the sensitivity to input membrane potential variation. The threshold decreases with input value exponentially.

4.1.3. MSAT with DET and DTT

MSAT shows that threshold-adapted neurons are insensitive to slow changes and selective to fast input variations with DET and DTT. In other words, adaptive thresholds filter out slow voltage fluctuations and corresponding neurons will not elicit a spike. The slow voltage fluctuations may come from the unexpected Spikes of Inactivated Neurons, so it relieves the SIN error partly and reduces the total spike number thus promoting energy efficiency. On the other hand, the threshold variation by DET makes threshold not increases endlessly and reduces appropriately for fear of large quantization error.

4.2. Spike confidence

As shown in Fig. 2, the early spikes elicited are not always reliable, and some parts of them are raised by SIN. Although threshold adaptive DTT can partly relieve this, neurons with SIN will not be distinguished from other normal neurons until they keep silent in longer time step. Inspired by many image recognition and detection works (Liu et al., 2016; Redmon et al., 2016), we import confidence to show how confident the elicited spikes are from the normal neurons to distinguish with SIN, as Fig. 3(b) shows. Within early time steps E, the spike confidence acts and every spike which should elicit use this spike confidence to generate a spike filter to determine fire spike or not rather than fire directly. The value of E is determined based on the statistics of the SIN ratio. Spike Confidence can be formulated as follows

$$c_i^l \sim Bernoulli(p^l) \tag{14}$$

where p^l is spike confidence of neurons in layer l and the value of p^l eaquals to the opposite proportion of SIN in layer l. Random variable c_i^l, which is sampled from

Algorithm 1 Conversion from ANN to SNN: Multi-stage adaptive threshold

Input: Pretrained ANN with L layer, training set

Parameter: inference time step T, spike confidence p, early time step E, threshold coefficient Cof

Output: The multi-stage adaptive SNN

1: **for** s = 1 to # of samples **do**
2:     $a^l \leftarrow$ layer-wise activation value
3:     **for** $l = 1$ to L **do**
4:         $V_{th}^l \leftarrow max[V_{th}^l, max(a^l)]$
5:     **end for**
6: **end for**
7: **for** t = 1 to T **do**
8:     **for** $l = 1$ to L **do**
9:         spike filter $c^l \leftarrow Bernoulli(p[l])$
10:         **for** $j = 1$ to neuron number of layer l **do**
11:             compute DTT and DET as Eq. 12 Eq. 13
12:             $\text{Cof}[l][j] \leftarrow sigmoid(DTT + DET)$
13:             $\text{SNN.layer}[l].V_{th}[j] \leftarrow \text{Cof}[l][j] * V_{th}^l$
14:             **if** fire spike and t < E **then**
15:                 spike = c^l * spike
16:             **end if**
17:         **end for**
18:     **end for**
19: **end for**

the Bernoulli distribution whose parameter is defined by the spike confidence, determines whether to fire a spike instead of firing directly. In this case the

accumulation of membrane potential becomes

$$\tilde{\Theta}_{t,i}^l = c_i^l * \Theta_{t,i}^l,$$

$$V_i^l(t) = V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^l(t)W_{ij}^l\tilde{\Theta}_{t,j}^{l-1} - V_{th,i}^l(t)\tilde{\Theta}_{t,i}^l. \qquad (15)$$

The MSAT and spike confidence ensure that all neurons adjust their thresholds and elicit spike according to the stimulus and in this way they become specialized. The pseudocodes of adaptive threshold are shown in Algorithm 1.

5. Experiment

Here we conduct experiments on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and ImageNet (Russakovsky et al., 2015) datasets to demonstrate the effectiveness of our proposed method. We use data augmentation, such as random horizontal flip, Cutout (DeVries & Taylor, 2017), and AutoAugment (Cubuk et al., 2019). Our batch size is 128 and the total training epoch is 300. The optimizer chooses stochastic gradient descent (SGD) with an initial learning rate of 0.1 and uses a cosine decay strategy. VGG16(Simonyan & Zisserman, 2014), ResNet20 and ResNet34(He et al., 2016) are used for target ANN as in previous works for comparison. Our code implementation of deep SNNs is based on the open-source SNN framework BrainCog (Zeng et al., 2022).

In the proposed adaptive thresholds, there are six hyperparameters and we give their meanings and values used in this paper in Table 2. Although hyperparameters are manually set, they do not vary with the dataset and setting these parameters is not difficult based on our experiments; our method works well for a wide value range of these parameters. Although these parameters can be set to trainable, we choose to manually set them for the consideration of biological plausibility according to (Fontaine et al., 2014; Azouz & Gray, 2000).

5.1. Comparison with the state of the art

As mentioned in related work, some works train a modified ANN to get short latency when converting ANN to SNN. In order not to do too many restrictions

Symbol	Definition	VGG16	ResNet20	ResNet34
α	left side slope	0.03	0.3	1.0
k_a	right side slope hyperparameter	1	1	1.0
k_i	right side slope hyperparameter	1.0	1.0	1.0
C	Input sensitivity	5.0	5.0	5.0
τ_{mp}	coefficient of DTT	1	0.5	0.5
τ_{rd}	coefficient of DET	1	0.5	0.5

Table 2: Summary of hyperparameters on different network.

on origin ANN, we choose to save the topology and use the original weights in the target ANN for universal conversion without retraining ANN. In this way, we compare the state-of-the-art approaches including TSC (Han & Roy, 2020), RMP-SNN (Han et al., 2020), Opt (Deng & Gu, 2021), Calibration (Li et al., 2021a), Burst (Li & Zeng, 2022).

As shown in Table 3, with the proposed method, all conversions are achieved with minimal accuracy degradation and shorter latency. In particular, on CIFAR-10, VGG16, we achieved a lossless conversion in only 112 time steps. We also validated the robustness of our method on ImageNet. As shown in Table 4, we also achieve comparative performance with SOTA methods and nearly lossless conversion from the target ANN. It is worth noting that though calibration (Li et al., 2021a) also uses the dynamic threshold, our method does not need to search for the best voltage threshold, which could result in less computational cost.

5.2. Performance on non-visual domains

Our work aims to contribute a biologically adaptive threshold system for ANN-SNN conversion, not limited to computer vision tasks and so MSAT is equally competent in other tasks. To show this, we further evaluate our method on non-visual domains, including natural language processing and speech.

For natural language processing, we choose IMDB dataset(Maas et al., 2011) for the sentiment classification task, and the network structure is three-layer

Method	Use DT	ANN	SNN	Loss	Step	ANN	SNN	Loss	Step
		VGG16, CIFAR-10				**ResNet20, CIFAR-10**			
RMP-SNN(Han et al., 2020)	✗	93.63%	93.63%	<0.01%	2048	91.47%	91.36%	0.11%	2048
TSC(Han & Roy, 2020)	✗	93.63%	93.63%	<0.01%	2048	91.47%	91.42%	0.05%	2048
Opt.(Deng & Gu, 2021)	✗	92.34%	92.24%	0.1%	128	93.61%	93.56%	0.05%	128
Burst.(Li & Zeng, 2022)	✗	95.74%	95.75%	0.02%	256	96.56%	96.59%	-0.03%	<256
Ours	✓	95.45%	95.45%	**0.00%**	**112**	96.37%	96.36%	**0.01%**	**174**
		VGG16, CIFAR-100				**ResNet20, CIFAR-100**			
RMP-SNN(Han et al., 2020)	✗	71.22%	70.93%	0.29%	2048	68.72%	67.82%	0.9%	2048
TSC(Han & Roy, 2020)	✗	71.22%	70.97%	0.25%	2048	68.72%	68.18%	0.54%	2048
Opt.(Deng & Gu, 2021)	✗	70.49%	70.47%	0.02%	128	69.80%	69.49%	0.31%	128
Calibration(Li et al., 2021a)	✓	77.89%	77.79%	0.1%	>512	77.16%	77.29%	-0.13%	64
Burst(Li & Zeng, 2022)	✗	78.49%	78.66%	-0.17%	128	80.69%	80.72%	-0.03%	<256
Ours	✓	78.49%	78.50%	**-0.01%**	**224**	80.69%	80.70%	**-0.01%**	**252**

Table 3: Experimental results on CIFAR-10 and CIFAR-100, DT means dynamic threshold.

Method	Use DT	ANN	SNN	Loss	Step
RMP-SNN	✗	70.64%	69.89%	0.75%	>2048
TSC	✗	70.64%	69.93%	0.71%	>2048
Opt.	✗	75.66%	75.44%	0.22%	>2048
Calibration	✓	75.66%	75.45%	0.21%	>2048
Burst.	✗	75.16%	74.94%	0.22%	256
Ours	✓	75.16%	74.93%	0.23%	2045

Table 4: Experimental results on ImageNet with ResNet34.

bi-directional LSTM we built ourselves. For the speech processing, we evaluate on the speech recognition task and select the google speech command dataset(Warden, 2018). The dataset consisted of 105,829 utterances of 35 words and each stored as a one-second (or less) WAVE format file. The specific architecture is modeled after the M5 network architecture described in Dai et al. (2017). We report the performance of MSAT on these two tasks and compare it with RMP-SNN. The results can be seen in Table 5.

Compared to RMP-SNN, MSAT can achieve lossless conversion much faster, even in only half the time steps, e.g., 7 steps on IDBM dataset for LSTM. On

Task	Dataset	Network	ANN Accuracy	Method	Step	SNN accuracy
NLP: sentiment classification	IDBM	LSTM	87.22%	RMP-SNN[1]	11	87.23%
				MSAT	7	87.25%
Speech: speech recognition	speech commands	M5	87.81%	RMP-SNN[1]	2048	86.83%
				MSAT	2048	87.33%

1. Our implemented.

Table 5: Experimental results on non-visual task.

complex tasks, such as speech recognition with 35 classes, MSAT performs better at the same time step. These results are consistent with earlier analysis and validate the effectiveness of MSAT.

5.3. Ablation study

In our method, V_{th}^l is chosen as the maximum activation value and as the initial value of the multi-stage adaptive threshold. In Fig. 4, the dotted lines indicate the target ANN accuracy, from which we obverse the DTT, DET, and fixed threshold impact on classification accuracy. It is intuitive that with either DTT or DET, we could achieve faster convergence rate, as shown in Fig. 4(b), 4(d). Instead, the heuristic method is hard to achieve high accuracy and fast inference speed simultaneously. For example, $0.2V_{th}$ in Fig. 4(a), 4(b) (yellow lines) lie above other color lines in early stage and have a shorter latency; however, yellow lines do not appear in the zoomed-in box and can not touch dotted lines, which means that the converted SNN threshold with $0.2V_{th}$ suffers from larger performance degradation. Although a manually selected threshold might achieve a satisfactory result, it needs lots of computation to search or carefully design for an appropriate threshold. The adaptive threshold MSAT with both DTT and DET could have a faster inference to achieve the target ANN accuracy and shows that the MSAT could find an appropriate threshold at each time step to reduce the quantization error while not importing larger clip error.

Moreover, we validate the effect of spike confidence on accuracy in early time steps. In our experiment, we apply spike confidence to the last IF neuron layer because the last layer is directly responsible for calculating results. The spike
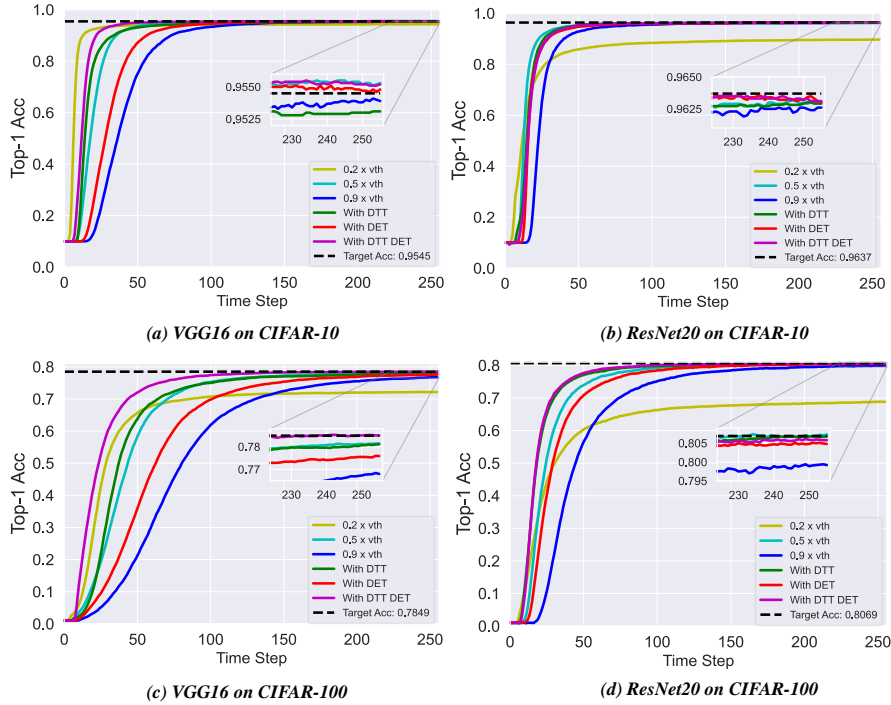
Figure 4: The accuracy curves on CIFAR-10 and CIFAR-100 datasets.

confidence stage holds 16 time steps. As we mentioned before, SIN takes a large proportion, so a large part of the error comes from SIN besides quantization and clip error. We use averaged number of SIN (ANS) over each neuron and the exclusion of spike confidence (SC) in both ANS and classification accuracy as baseline. Table 6 shows that spike confidence could reduce the averaged number of SIN and improve classification accuracy, validating that SNN in the early time steps could elicit fewer error spikes and benefit from spike confidence.

Fig. 5 demonstrates the impact of adding spike confidence on performance. The improvement on CIFAR-10 is not as significant as on CIFAR-100, mainly because the SIN proportion in CIFAR-10 is already quite smaller than that in CIFAR-100. To show this, we statistic the neurons with SIN ratio in the last layer for CIFAR-10-VGG16, CIFAR-10-ResNet20, CIFAR-100-VGG16 and

Metric	VGG16		ResNet20	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
w/o SC ANS	2.92	4.467	1.972	6.309
w/ SC ANS	$\mathbf{2.343}_{-0.577}$	$\mathbf{2.514}_{-1.953}$	$\mathbf{1.558}_{-0.414}$	$\mathbf{3.063}_{-3.246}$
w/o SC Acc (T=32)	93.48	61.16	93.30	71.92
w/ SC Acc (T=32)	$\mathbf{93.85}_{+0.37}$	$\mathbf{66.73}_{+5.57}$	$\mathbf{94.02}_{+0.72}$	$\mathbf{73.25}_{+1.33}$

Table 6: Averaged Number of SIN and classification accuracy on different networks and datasets. The subscripts mean the decreased ANS and increased accuracy compared to baselines (1st and 3rd rows).

(a) Spike Confidence on CIFAR-10

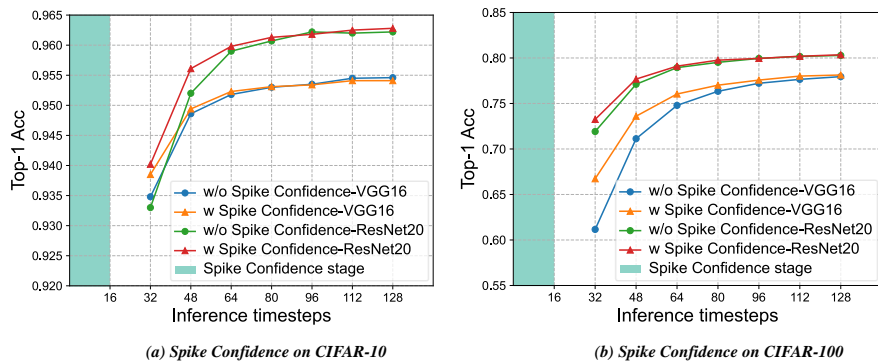*(b) Spike Confidence on CIFAR-100*

Figure 5: classification accuracy improves with spike confidence. Spike confidence is used within the time steps of the green area.

CIFAR-100-ResNet20 respectively. The ratio is 0.059, 0.063, 0.617, and 0.506 respectively, which means that SIN error is a very small part in CIFAR-10 whose influence on performance degradation is not as significant as in CIFAR-100. This also explains the different degrees of performance improvement on the two datasets by spike confidence.

5.4. Energy-efficiency and sparsity

In this section, we compute the firing rate of each layer to evaluate our spike sparsity. We use the same way in Rathi & Roy (2020) to compute energy

consumption. 32-bit floating-point AC and MAC per operation consume 0.9pJ and 4.6pJ individually. Furthermore, there is no operation in SNN if no spike elicits and the first layer is the real-valued input so the energy consumption for this layer is still calculated as 4.6pJ per operation. We chose the VGG16 on CIFAR-100 dataset with time step T=64, the average firing rate is 0.0585 and we have 43.20% energy consumption compared to the target ANN. This is more energy efficient than 69.30% reported in Li & Zeng (2022). Low firing rate and less than half the energy consumption reflects the spike sparsity of MSAT.

6. Conclusion

This paper introduces a multi-stage adaptive threshold into ANN-SNN conversion and demonstrates its advantages in terms of accuracy-latency trade-off. This is the first work to build a biologically inspired multi-stage adaptive threshold for converted SNN. In addition, we focus on the SIN error that occupies a large proportion in the early period and accounts for accuracy degradation but is rarely explored in existing works. Through statistical analysis of SIN errors, we propose to mitigate SIN errors by using spike confidence. Our experiments show that the converted SNN with MSAT and spike confidence has comparable accuracy to the state of the art while with lower latency and power consumption. Good performance on classification tasks in non-visual domains also demonstrates the universality of the proposed approach.

7. Acknowledgments

References

Azouz, R., & Gray, C. M. (2000). Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. *Proceedings of the National Academy of Sciences*, *97*, 8110–8115.

Azouz, R., & Gray, C. M. (2003). Adaptive coincidence detection and dynamic gain control in visual cortical neurons in vivo. *Neuron*, *37*, 513–523.

Bu, T., Ding, J., Yu, Z., & Huang, T. (2022). Optimized potential initialization for low-latency spiking neural networks. *arXiv preprint arXiv:2202.01440*, .

Bu, T., Fang, W., Ding, J., Dai, P., Yu, Z., & Huang, T. (2021). Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*.

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, *95*, 1–19.

Caporale, N., Dan, Y. et al. (2008). Spike timing-dependent plasticity: a hebbian learning rule. *Annual review of neuroscience*, *31*, 25–46.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 113–123).

Dai, W., Dai, C., Qu, S., Li, J., & Das, S. (2017). Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 421–425). IEEE.

Deng, S., & Gu, S. (2021). Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, .

DeVries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, .

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, *9*, 99.

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

Ding, J., Dong, B., Heide, F., Ding, Y., Zhou, Y., Yin, B., & Yang, X. (2022). Biologically inspired dynamic thresholds for spiking neural networks. *arXiv preprint arXiv:2206.04426*, .

Fontaine, B., Peña, J. L., & Brette, R. (2014). Spike-threshold adaptation predicted by membrane potential dynamics in vivo. *PLoS computational biology*, *10*, e1003560.

Han, B., & Roy, K. (2020). Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision* (pp. 388–404). Springer.

Han, B., Srinivasan, G., & Roy, K. (2020). Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13558–13567).

Hao, Y., Huang, X., Dong, M., & Xu, B. (2020). A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule. *Neural Networks*, *121*, 387–395.

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

Henze, D., & Buzsáki, G. (2001). Action potential threshold of hippocampal pyramidal cells in vivo is increased by recent spiking activity. *Neuroscience*, *105*, 121–130.

Kim, S., Park, S., Na, B., Kim, J., & Yoon, S. (2020). Towards fast and accurate object detection in bio-inspired spiking neural networks through bayesian optimization. *IEEE Access*, *9*, 2633–2643.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images, .

Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience*, *10*, 508.

Li, Y., Deng, S., Dong, X., Gong, R., & Gu, S. (2021a). A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning* (pp. 6316–6325). PMLR.

Li, Y., & Zeng, Y. (2022). Efficient and accurate conversion of spiking neural network with burst spikes. *arXiv preprint arXiv:2204.13271*, .

Li, Y., Zeng, Y., & Zhao, D. (2021b). Bsnn: Towards faster and better conversion of artificial neural networks to spiking neural networks with bistable neurons. *arXiv preprint arXiv:2105.12917*, .

Liu, F., Zhao, W., Chen, Y., Wang, Z., & Jiang, L. (2022). Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37). Springer.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th an-*

nual meeting of the association for computational linguistics: Human language technologies (pp. 142–150).

Pena, J. L., & Konishi, M. (2002). From postsynaptic potentials to spikes in the genesis of auditory spatial receptive fields. *Journal of Neuroscience*, *22*, 5652–5658.

Pérez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., & Linares-Barranco, B. (2013). Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *IEEE transactions on pattern analysis and machine intelligence*, *35*, 2706–2719.

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

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*, 211–252.

Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, *13*, 95.

Shen, G., Zhao, D., & Zeng, Y. (2022). Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks. *Patterns*, (p. 100522).

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

Wang, Z., Lian, S., Zhang, Y., Cui, X., Yan, R., & Tang, H. (2022). Towards lossless ann-snn conversion under ultra-low latency with dual-phase optimization. *arXiv preprint arXiv:2205.07473*, .

Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, .

Wilent, W. B., & Contreras, D. (2005). Stimulus-dependent changes in spike threshold enhance feature selectivity in rat barrel cortex neurons. *Journal of Neuroscience*, *25*, 2983–2991.

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

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., & Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 1311–1318). volume 33.

Yu, Q., Ma, C., Song, S., Zhang, G., Dang, J., & Tan, K. C. (2021). Constructing accurate and efficient deep spiking neural networks with double-threshold and augmented schemes. *IEEE Transactions on Neural Networks and Learning Systems*, *33*, 1714–1726.

Zeng, Y., Zhao, D., Zhao, F., Shen, G., Dong, Y., Lu, E., Zhang, Q., Sun, Y., Liang, Q., Zhao, Y. et al. (2022). Braincog: A spiking neural network based brain-inspired cognitive intelligence engine for brain-inspired ai and brain simulation. *arXiv preprint arXiv:2207.08533*, .

Zhang, W., & Li, P. (2020). Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, *33*, 12022–12033.

Zhao, D., Zeng, Y., Zhang, T., Shi, M., & Zhao, F. (2020). Glsnn: A multi-layer spiking neural network based on global feedback alignment and local stdp plasticity. *Frontiers in Computational Neuroscience*, *14*, 576841.