Training a General Spiking Neural Network with Improved Efficiency and Minimum Latency

Yunpeng Yao 202132748@MAIL.SDU.EDU.CN
Information Science and Engineering, Shandong University, Shandong, China

Man Wu* WU.MAN.WI5@AM.ICS.KEIO.AC.JP
Department of Information and Computer Science, Keio University, Kanagawa, Japan

Zheng Chen CHENZ@SANKEN.OSAKA-U.AC.JP
SANKEN, Osaka University, Osaka, Japan

Renyuan Zhang RZHANG@IS.NAIST.JP
Division of Information Science, Nara Institute of Science and Technology, Nara, Japan
∗ *Corresponding Author*

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Spiking Neural Networks (SNNs) that operate in an event-driven manner and employ binary spike representation have recently emerged as promising candidates for energy-efficient computing. However, a cost bottleneck arises in obtaining high-performance SNNs: training a SNN model requires a large number of time steps in addition to the usual learning iterations, hence this limits their energy efficiency. This paper proposes a general training framework that enhances feature learning and activation efficiency within a limited time step, providing a new solution for more energy-efficient SNNs. Our framework allows SNN neurons to learn robust spike feature from different receptive fields and update neuron states by utilizing both current stimuli and recurrence information transmitted from other neurons. This setting continuously complements information within a single time step. Additionally, we propose a projection function to merge these two stimuli to smoothly optimize neuron weights (spike firing threshold and activation). We evaluate the proposal for both convolution and recurrent models. Our experimental results indicate state-of-the-art visual classification tasks, including CIFAR10, CIFAR100, and TinyImageNet. Our framework achieves 72.41% and 72.31% top-1 accuracy with only 1 time step on CIFAR100 for CNNs and RNNs, respectively. Our method reduces 10× and 3× joule energy than a standard ANN and SNN, respectively, on CIFAR10, without additional time steps.

Keywords: List of keywords separated by semicolon.

1. Introduction

Spiking Neural Networks (SNNs) have garnered attention due to the capacity of low-power computing, which is inspired by the event-driven nature of human brain. The neurons in SNNs operate by transmitting information through event-oriented *binary spike* as opposed to analog value, mitigating the computational cost associated with multiplication operations that are ubiquitous in ANNs Datta et al. [2022]; Wu et al. [2022].

Despite their notable energy efficiency, high-performance SNNs are notoriously difficult to develop, particularly for continuous and high-dimensional signals Chowdhury et al. [2022a]. SNN neurons utilize an accumulating-and-firing mechanism, known as the membrane potential, which is a dynamic variable that evolves over time, unlike the fixed weights. This mechanism allows neurons to receive and transmit information through activated spike sequences. However, guiding representative activation for continuous input is tough. The sparsity that arises from discrete spike representation inherently results in a consequent loss of information Datta et al. [2022], making SNNs more difficult for learning complex patterns in real-world signals. Meanwhile, the discrete nature of spikes also poses an obstacle in model convergence, leading to a gradual decay of effective spike activation in deeper layers Rathi and Roy [2021a].

SNNs trained using the ANN-to-SNN conversion, which involves transferring the trained weights of an ANN onto an SNN, is a dominant method in the field Rueckauer et al. [2017]. Coupled with appropriate learning algorithms, e.g., surrogate gradient learning (SGL) Neftci et al. [2019], conversion SNNs can perform comparable results to state-of-the-art (SOTA) ANNs methods Cao et al. [2015]. Since additional training phases and meticulous attention of modeling are required Simonyan and Zisserman [2014], and computational complexity is inevitably increased. Additionally, conversion SNNs typically cooperate with pre-encoding techniques (e.g., rate coding, rank-order coding, etc) to transform analog input into well-initialized spike sequences Kiselev [2016]. These coding methods are iteratively applied over a large number of *time steps* [1] for training, enabling continuous complementation of information source and optimizing membrane potential variable. Computationally, the time steps (usually >100) lead to a queuing of serial processing that increases end-to-end latency (proportional to the number of time steps) and costly memory Datta et al. [2022].

To tackle this cost bottleneck, some works aim to develop a direct training framework that integrates the advanced computational mechanism to improve the optimization of SNN neurons. For instance, strengthening the local feature receptive fields by convolutional neural networks (CNNs), some works reduce 50% time steps required to achieve satisfactory results Rathi et al. [2020]. Recent studies employ hierarchical computation in CNN-based architectures, such as VGG and ResNet, within SNN frameworks Kim et al. [2018], which can train very deeper SNNs. More recent studies Ponghiran and Roy [2022] utilize the sequential-order capabilities of recurrent neural networks (RNNs) with SNNs to improve the recurrence dynamics for sequential learning. These investigations show efficacy with fewer time steps (< 20), which can result in improved computational efficiency Ponghiran and Roy [2022]. While the use of advanced computational mechanisms has yielded various SOTA results to SNNs research and reduced certain computational costs, it is still challenging to maintain accuracy in the direct training method as the number of time steps decreases. Typically, a reduction in time steps leads to a drop in accuracy, it leads to a *trade-off between accuracy and the time steps used.* This trade-off issue imposes a limit on fully exploiting the low-power and energy-efficient computing capabilities of SNNs. This paper provides a new perspective on this issue by proposing a novel and general learning framework for the direct training method.

Our focus is on maximizing the effective utilization of information in SNN neurons within limited time steps. To achieve this, we propose a learning framework that involves (i) learning

[1] time steps refer to the times of forward propagation in each spiking neuron.

feature from different input receptive fields and (ii) optimizing the spike stimuli through a projection function. Specifically, SNN neurons receive multiple local inputs, which are extracted using a sliding window approach and grouped accordingly in each layer. This enables SNN neurons to learn information at different levels of granularity to model dependencies at different scales. We record the group-wise membrane potentials and recurrently utilize them to optimize SNN neurons within a single time step. We further propose a projection function to smooth neuron activation, optimizing the membrane potential by both previous group membrane potential and current stimuli. This facilitates layer communication and ensures the effective transmission of information. The main contributions are summarized as follows:

- This study tackles the trade-off between *time-step* and *performance*, and proposes a novel training framework for SNNs that involves the refinement of input information and the optimization of spiking neurons to enhance effective spike activation.
- Our framework incorporates a projection function to generalize its applicability to both Convolutional - and Recurrent- based architectures, based on an analysis of the activated operation of leaky integrate-and-fire (LIF) neurons.
- Our method shows high effectiveness and efficiency in experimental results. Our framework achieves top-1 accuracy of 72.31% on CIFAR100, with only `1` time step. Moreover, to the best of our knowledge, this is the first RNN-SNN model to achieve competitive results in visual tasks. On CIFAR10, our SNN significantly reduces computational cost with $(10 \sim 102)\times$ and $(1.4 \sim 3)\times$ joules compared to previous ANN and SNN approaches respectively.

2. Preliminary: Leaky Integrate and Fired Model

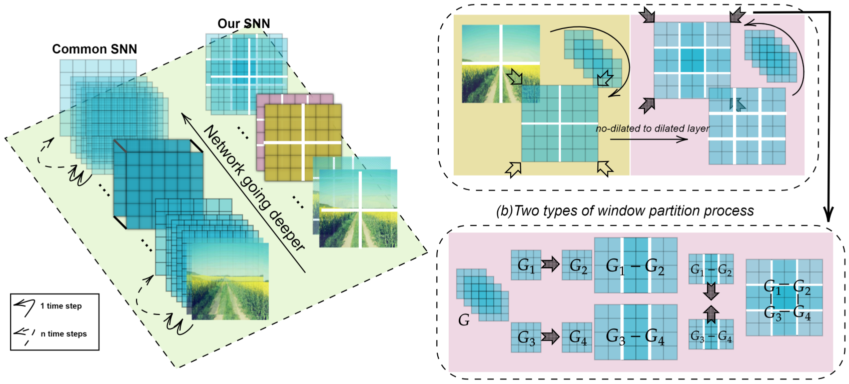
SNNs are computationally capable of universal approximation and can mimic any feed-forward neural network by adjusting synaptic weights. Thus, they excel at processing spatio-temporal and fluctuation information as spike flows.

There have been many proposals for different types of SNN neurons, in this work, we follow the works of Lotfi Rezaabad and Vishwanath [2020]; Chen et al. [2022]; Datta et al. [2022] and use the well-known iterative leaky integrate-and-fire (LIF) SNN model in this work. The conventional LIF describes spiking nature of neurons from feature stimuli, membrane potential accumulation, and spike firing, to membrane potential reset as membrane current $I(t)$ and membrane potential $V(t)$. Neuronal dynamics can be described by

$$\frac{\partial I(t)}{\partial t} = -\,\tau_{syn} I(t) + \sum_{j=1}^{J} (w_j, X_j(t-1)) \tag{1}$$

$$\frac{\partial V\ (t)}{\partial t} = -\tau_{mem} V\ (t) +\ I(t)\ - \delta o_l^{t-1} \tag{2}$$

where t and l denotes the t-th time-step and the i-th layer in the architecture, respectively. $V(t)$ is the corresponding membrane potential and τ is the decay rate of the current and potential. Hence, o_{t-1}^{i} represents the i-th firing spike with weight $w_{j,n}$ from the previous layer and time step $(t-1)$. Given the above information, $I(t)$ is the pre-synaptic input,



(a)Comparison between forward propagation of common SNN and ours

(b)Two types of window partition process

(c)Weight condense in dilated layer

Figure 1: Windows partition for spiking neuron layers.

and its information carrier is activated by the previous time index. When the membrane potential $V(t)$ reaches the firing threshold δ, the neuron will output a firing spike, i.e., *1*.

$$o^i_{t+1} = \begin{cases} 1 & \text{if } V(t) > \delta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

SNNs place less emphasis on the number of spikes indicated by 1, and instead, consider the number of 0 between two 1 spikes as a reflection of the accumulation of event status, which is incorporated into information representations. The spike generated by o^i_{t+1} propagates forward and activates the neurons of the next layer.

3. Methodology

The goal of this work is to enable SNNs to learn visual features without requiring a large number of time steps. To achieve this, we use an input stream $X \in \mathbb{R}_x^{\left(G_x^1,\ldots,G_x^P\right)\times T}$ that consists of a binary signal (0/1), which is divided into P groups up to time step T, with each group having a feature space of G_x. Equally, we also use a membrane potential stream $M \in \mathbb{R}_m^{\left(Gm^1,\ldots,G_m^P\right)\times T}$ that corresponds to X_l. Based on these, the proposed SNN model $F\left(X|\beta;\Theta;T\right)$ aims to fit the target distribution Y, where Θ is a set of model parameters (such as weight $w_{j,n}$ from the previous layer) and β is an auxiliary parameter that describes membrane potential of SNN. In this sense, we propose a novel framework to optimize the learning ability of F with a minimum iteration time of T.

3.1. Dilated Window for Spiking Neuron.

Inspired by the works of Dosovitskiy et al. Dosovitskiy et al. [2020] and Liu et al. Liu et al. [2021], which regionalize global information, our approach employs complementary paired-windows. However, we modify one of them to accommodate the operation of spiking neurons between consecutive layers as shown in Fig. 1(b). Specifically, non-dilated windows are designed to accumulate membrane potential locally, and dilated windows are proposed to enhance the interaction between neighboring windows.

Membrane potential in non-dilated windows. The non-dilated windows are arranged to evenly partition the membrane potential in a non-overlapping manner. Under this condition, the tensor stream of input feature X_l and membrane potential M_l are first partitioned into p groups, denoted as G_x and G_m respectively. Then, each group spiking neuron G_x accumulates membrane potential, activates a spike, and resets in order. The accumulated membrane potential in G_x^p will be delivered to the next G_x^{p+1} in sequence to introduce cross-neuron connections up to iteration T. Finally, every G_x^p is activated by the membrane potential G_m^{p-1} of last group and the weight w corresponding to G_x^p, and $G_x^{1,\cdots,p}$ is recomposed into origin shape as X_l. Layers with partition windows can be represented as follow:

$$F_l(X|\beta;\Theta;t) = F_l\left(G_x^1|G_m^1;\Theta;t\right) \rightarrow F_l\left(G_x^2|G_m^2;\Theta;t\right) \xrightarrow{\cdots} F_l\left(G_x^p|G_m^p;\Theta;t\right) \tag{4}$$

Membrane potential in dilated windows. The window-based LIF module mentioned earlier has a limitation in its modeling power, as it lacks connectivity across regions. To address this issue, we propose a dilated window that alternates between two partitioning configurations in consecutive SNN blocks, thereby enhancing connectivity across windows. The dilated window size dilates 1/2 compared to the normal window, and the updating scheme of membrane potential is consistent with the normal window. Each dilated window is therefore able to interact with all neighboring non-overlapping windows in the preceding layer as it expands the scope of partitioning. Notice that the size of G_x and G_m partitioned by the dilated window is larger than origin X_{l-1} and M_{l-1} from non-dilated window after recomposing. Consequently, the overlapping information will be overutilized since more additional feature space, and calculations will also be introduced.

To tackle this issue, we propose a more efficient computation approach that condenses the overlapping part with a certain weight, as shown in Fig. 1. Weighted-condensed compresses the information streams X_l and M_l into the original feature size with certain weights, namely 4 and 2. The weight assigned to a region increases as the number of overlaps increases. More specifically, due to the division operation of weighted-condense, "median spikes (Ms)" (such as 0.25, 0.5, and 0.75) will be produced in overlaps region of recomposed X_{l+1}. For instance, if a spike appears in the overlapping area with weight 4, it becomes a Ms as 0.25. To maintain the low power advantage of event-driven, we set a region threshold (Th_R) to integrate these Ms into spikes. This threshold is set to 0.1. Pseudo code for reorganizing a dilated window is presented in Appendix.

Due to the locality of dilated windows, a fully connected layer is established after each dilated layer to complete the learning of global features. We therefore reduced the number of layers in the network to maintain the same amount of parameters with other researches.

3.2. Fusion for Multi-Direction Membrane Potential

As mentioned above, spiking neurons update their own activation status X_l partially from the last window region. Although the membrane potential has accumulated more information by utilizing the rest feature region in the same layer, the information in the original area is still underutilized. To further enhance the descriptive ability of β, the fusion of multi-directional membrane potentials is critical. The most common fusion method is to add two sets of membrane potentials.

As shown in Fig. 2, two issues emerge in this simplest fusion method (i.e., membrane potential streams M in both directions are projected onto $y = 2x$): (i) Since G_m always

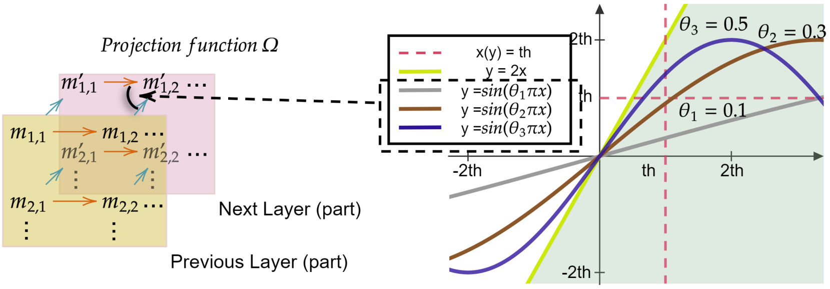


Figure 2: Fusing membrane from two directions and projection function Ω.

accumulates, including negative values that are not reset within one time step, it can lead to negative values accumulating towards infinity. (ii) The simple linear sum function causes the fused membrane potential to increase monotonically and even exceed multiple LIF thresholds. This means that neurons in further regions will be over activated in advance without receiving any new information. To fuse the membrane potential smoothly, we designed a projection function, denoted by $\Omega(\cdot)$, which projects the mixed membrane potential into an optimized space, as shown in Eq. 5:

$$\Omega(G_\beta) = \Omega\left(G_m, G_{m'}\right) = 2sin\left(\frac{\theta\pi}{2}G_m\right) * cos\left(\frac{\theta\pi}{2}G_{m'}\right) \tag{5}$$

where the variables G_m and $G_{m'}$ represent multi-directional membrane potentials from G_{p-1} and G_{l-1}, respectively. Different values of θ correspond to different projection spaces. As shown in Fig. 2, Eq. 5 deforms the function $y = \sin(\theta\pi x)$, going beyond simple linear summation of membrane potential from two directions. First, it effectively limits the trend toward negative infinity, allowing neurons in the negative area to be reactivated. Second, the membrane potential no longer exceeds multiple LIF thresholds, greatly reducing the influence of spiking neuron $G_m^{p,l}$ on the "remote future" $G_x^{P,L}$. Additionally, non-linearity is introduced, which smooths neuron activation and improves the representation of complex distributions by the neurons. It is worth noting that the function $\Omega(\cdot)$ used for the projection is not necessarily the optimal expression for achieving multi-directional membrane potential fusion. Several other possible functions and different values of θ are experimented with in section 4.3.

where G_m and $G_{m'}$ present multi-direction membrane potential from G_{p-1} and G_{l-1} respectively. And different θ represent different projection spaces. As Fig. 3 shown, Eq. 5 is a deformation of $y = sin(\theta\pi x)$, which surpasses the simple linear summation of membrane potential from two directions. First, it is obvious that the trend of negative infinity has been well restrained, that neurons in the negative area will have an opportunity to be reactivated. Second, membrane potential will not exceed multiple LIF thresholds anymore, limiting the influence of spiking neuron $G_m^{p,l}$ on the "remote future" $G_x^{P,L}$ to a great extent. In addition, non-linearity is introduced that smooths neuron activation and improves neuronal representation of complex distributions. As a note, it is not said that project function $\Omega(\cdot)$ is the optimal expression for achieving multi-direction membrane potential fusion. Several possible functions and different θ are experimented as ablation in section 5.3. As shown in

Fig.3, two issues will emerge in this simplest fusion way (i.e, membrane potential streams M in both directions are projected on $y = x$ or $z = x + y$): (i) Since G_m will always accumulate including negative values and not reset (in 1-time step), including negative values in the direction of infinity. (ii) The simple linear sum function makes the fused membrane potential monotonically increase and even exceed multiple LIF thresholds. This means that neurons in further regions are activated in advance without receiving any information. To fuse membrane potential smoothly, we design a project function $\Omega(\cdot)$ to project the mixed membrane potential into an optimized space as Eq. 5.

3.3. Theoretical Analysis

The impacts of utilizing multi-direction membrane potential on SNNs are analyzed in this section. With theoretical tools related to the firing reset mechanism of LIF, we discover that our framework can significantly improve spike rates during the training process. Furthermore, the effect of project function $\Omega(\cdot)$ with different radian factors θ is also explained.

Multi-direction membrane potential. Eq. 1 to Eq. 3 in the preliminary section demonstrate that conventional LIF neurons receive a spiking signal $X_l(t)$ and accumulate membrane potential $M_l(t)$ from the previous moment $t-1$, where t is an element of the set $(1, 2, ..., T)$. However, it is expensive to calculate neuron states by directly solving a continuous function, the neuronal dynamics can be generally done by discretely evaluating the equations over small time steps as:

$$V[n] = \alpha V[n-1] + \ I[n] \ - \delta o[n-1] \qquad o[n] = \begin{cases} 1 & \text{if } V[n] > \delta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where n represents an discrete index of simulation time step, $\alpha = -\tau_{mem}$ that a leakage coefficient. When the neurons w_j align with the preliminary findings, Eq. 7 and Eq. 8 can be utilized to show a contrast with Eq. 1 and Eq. 2. In particular, the term $I(l, \mathbf{t})$ denotes the synaptic current, which is a differential function that relates to layer l and time steps t (assuming that the time step is limited to 1, we treat t as a constant hereafter). And the term $V_p(G, \mathbf{t})$ represents the membrane potential, which is a differential function that depends on regions of multi-pole direction G and not merely on the inputs at each time steps t.

$$\frac{\partial I(l, \mathbf{t})}{\partial l} = -\tau_{syn} \lim_{\mathbf{t} \to 1} I(l, \mathbf{t}) + \sum_{j=1}^{J} (w_j, X_j(l, \mathbf{t})) \tag{7}$$

$$\frac{\partial V_p(G, \mathbf{t})}{\partial G} = -\tau_{mem} \lim_{\mathbf{t} \to 1} V_p(G, \mathbf{t}) + \ I(l, \mathbf{t}) \ - \delta o_{l-1}^{\mathbf{t}} \tag{8}$$

To demonstrate the superiority of the proposed framework explicitly, we employ the Taylor formula to approximate the truth by discretizing the continuous values $\partial I(t)$ and $\partial V(t)$, as shown in Eq. 9. This discretization preserves the same form as the common LIF model. The detailed mathematical derivation is shown in Appendix.

$$V[\eta, \iota, n] = \alpha V_p[\eta - 1, \iota - 1, n] + \ I_p[\iota, n] \ - \delta o[\iota - 1] \ \ (n = 1) \tag{9}$$

where η and ι represent the unit separation of two different potential directions, α is an estimated leakage coefficient after fusion.

The synaptic currents of the spiking neurons are multiplied by the same constant at every time step. This leads to fast-diminishing gradients during a backward pass, as shown in previous research Ponghiran and Roy [2022]. In our work, updating the synaptic currents is solely determined by the inputs at each layer, regardless of their existing values. This approach avoids a large number of extra calculations and latency ($\times T$), and compensates for the inadequacy-activation of neurons that accumulate membrane potential with limited time steps t. Neurons in each region accumulate membrane potential in both directions and the closer a region approaches P, the more times it will accumulate.

$$N_{convention}^{Accumulation}[m_{G_p} * T] \approx N_{bi-direction}^{Accumulation}[(m_{G_1} + m_{G_2} + ... + m_{G_p}) * 1] \tag{10}$$

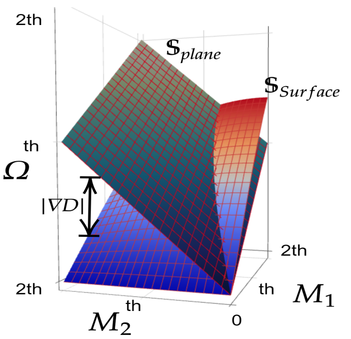


Figure 3: Example for $\theta = 0.5$.

Herein, as p increases, the number of times that the membrane potential accumulates approximately equals that obtained using a large number of T.

Project function $\Omega(\cdot)$ and radian factors θ. Function $\Omega(\cdot)$ projects membrane potential onto an optimal fusion surface. Different radian factors θ determine the amplitude and steepness of the surface. To avoid the membrane potential tending towards negative infinity or over-activation (exceeding the threshold multiple times) after fusion, we select the surface by adjusting θ. Regardless of θ, the negative infinity of the fused membrane potential is resolved as mentioned above. Besides, to detailed analysis over-activation issue, we use the area difference ∇D ranged in $R \in (th, 2 * th)$ between the fusion surface ($\mathbb{S}_{Surface} = \Omega(\mathbb{X}, \mathbb{Y})$) and the linear plane ($\mathbb{S}_{Plane} = \mathbb{X} + \mathbb{Y}$) as an indicator. This range is chosen because linear fusion only causes over-activation within this range. The area difference can be represented by Eq. 12. Note that since both the surface and the plane are origin-symmetric, the formula only applies to two positive shaft directions (x and y).

$$\nabla D = \mathbb{S}_{Surface} - \mathbb{S}_{Plane} \tag{11}$$

$$\begin{aligned} \mathbb{S}_{Surface} &= \int_{th}^{2*th} \int_{th}^{2*th} \Omega(M_1, M_2|\theta) dm_1 dm_2 = \mathbb{S}(\theta) \\ \mathbb{S}_{Plane} &= \int_{th}^{2*th} \int_{th}^{2*th} (M_1 + M_2) dm_1 dm_2 = 3(th)^3 \end{aligned} \tag{12}$$

The area of the projection plane $\mathbb{S}_{Plane}$ is determined by the threshold of the LIF model, the difference ∇D depends only on the radian factors θ. Theoretically, the greater the absolute difference, the greater the possibility of over-activation.

Results and detailed calculation process of the difference ∇D are presented in the supplementary material (consistent with the experiment, th is set as 0.5). The over-activation of spiking neurons caused by membrane potential fusion can be suppressed with each θ. More specifically, according to the experimental results in section 4.4, CNN and RNN-based models shows different association with θ.

Table 1: Performance on Cifar10, Cifar100, and Tiny-ImageNet datasets.

Datasets	Method	Type	Baseline	Acc(%)	Time steps(T)
Cifar10	SDN-LIFHunsberger and Eliasmith [2015]	ANN Conversion	2C, 2L	82.95	6000
	IOMBu et al. [2022]	ANN Conversion	ResNet-20	92.75	512
	SpikeConverterLiu et al. [2022]	ANN Conversion	ResNet-20	91.47	16
	Diet-SNNRathi and Roy [2021b]	C&T*	ResNet-20	92.54	10
	AutoSNNNa et al. [2022]	SNN Training	NAS64	92.54	8
	STBP-tdBNZheng et al. [2021]	SNN Training	ResNet-19	93.16	6
	MLF(K=1)Feng et al. [2022]	SNN Training	DS-ResNet-12	92.46	4
	Dspike SNNLi et al. [2021]	SNN Training	ResNet-18	93.13	2
	Emporal pruningChowdhury et al. [2022b]	SNN Training	VGG16	93.05	5-1
	This work CNN	SNN Training	ResNET-12F	**93.07**	**1**
	LSTM		VIT-12	92.68	**1**
	CNN-MSp**		ResNET-12F	**93.27**	**1**
	LSTM-MSp		VIT-12	93.13	**1**
Cifar100	RMP-SNNHan et al. [2020]	ANN Conversion	VGG16	70.93	2048
	IOMBu et al. [2022]	ANN Conversion	ResNet-20	70.51	128
	Diet-SNNRathi and Roy [2021b]	C&T	ResNet-20	69.67	5
	AutoSNNNa et al. [2022]	SNN Training	NAS64	69.16	8
	Dspike SNNLi et al. [2021]	SNN Training	ResNet-18	71.68	2
	Emporal pruningChowdhury et al. [2022b]	SNN Training	VGG16	70.15	5-1
	This work CNN	SNN Training	ResNET-12F	**72.41**	**1**
	LSTM		VIT-12	**72.31**	**1**
Tiny ImageNet	Spike-NormNa et al. [2022]	ANN Conversion	VGG16	48.60	2500
	Spike-ThriftKundu et al. [2021]	C&T	VGG16	51.92	150
	CATLew et al. [2022]	SNN Training	VGG16	57.40	48
	AutoSNNNa et al. [2022]	SNN Training	NAS64	46.79	8
	This work CNN	SNN Training	ResNET-12F	**57.91**	**1**
	LSTM	SNN Training	VIT-12	**57.55**	**1**

* A combination method of ANN conversion and direct training

** The case where Median spike (Ms) is not activated to a complete spike.

4. Experiments and Results

Please refer to the supplementary material for details on the experimental settings.

4.1. Performance Comparison

We compare our models (CNN/LSTM) with various state-of-the-art (SOTA) SNNs, and the results are shown in Table 1. To adapt different models, we choose ResNet-12 Touvron et al. [2022] as the baseline network for the CNN model, we have show the reason in section 3. Besides, VIT-12 is for the RNN model (the networtk of residual series is not suitable for RNN). Since LSTM outperforms other RNN-based methods, we only report the results of LSTM-based models. The results of other RNN-based models are included in the Appendix. Our CNN-based models achieve top-1 accuracy of 93.07%, 72.41%, and 57.91% on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, with just 1 time step (T=1). Analogously, the top-1 accuracies for the LSTM-based models are 93.07%, 72.31%, and 57.55% regarding the same datasets. In terms of efficiency, SNN performs equally well or better than other models in terms of accuracy with our framework, while achieving significantly lower inference latency. Importantly, proposal enables us to reduce the SNN latency to the lowest possible limit (**time steps = 1**) without the need for pre-training. Furthermore, to our knowledge,

Table 2: Computing cost and energy consumption in 45nm CMOS on Cifar10.

Method	Model	#Add	#Mult	Energy
ANN	Res-19	2285M	2285M	12.6J
ANN(ours)	ResNET-12F	247M	247M	1.35J
ANN(ours)	VIT-12	251M	251M	1.38J
SNN	Res-20(T=5)	142M	8.80M	168mJ
SNN [26]	Res-19(T=2)	360M	6.80M	355mJ
SNN(ours)	Res12F-CNN(T=1)	132M	2.14M	128mJ
SNN(ours)	VIT-12-LSTM(T=1)	79M	5.31M	96mJ

the RNN-SNN method is the first to achieve comparable results to CNN-based models for visual classification tasks.

4.2. Energy Consumption

In ANN, each operation computes a dot product involving one multiplication and addition (MAC) in floating-point (FP) format, whereas, in SNNs, the multiplication is eliminated by the binary spike. Energy is therefore saved due to a large number of more expensive multiplication has been replaced by addition operations. Namely, SNN demonstrates the energy efficiency gains by the cheaper multiplexer and compactor due to the event-driven paradigm Horowitz [2014]. In this manner, the addition operation is a primary energy consumer in SNNs, and its frequency is largely determined by the spike rate. The spike rate in iso-architecture SNN is commonly specified as Eq. 13.

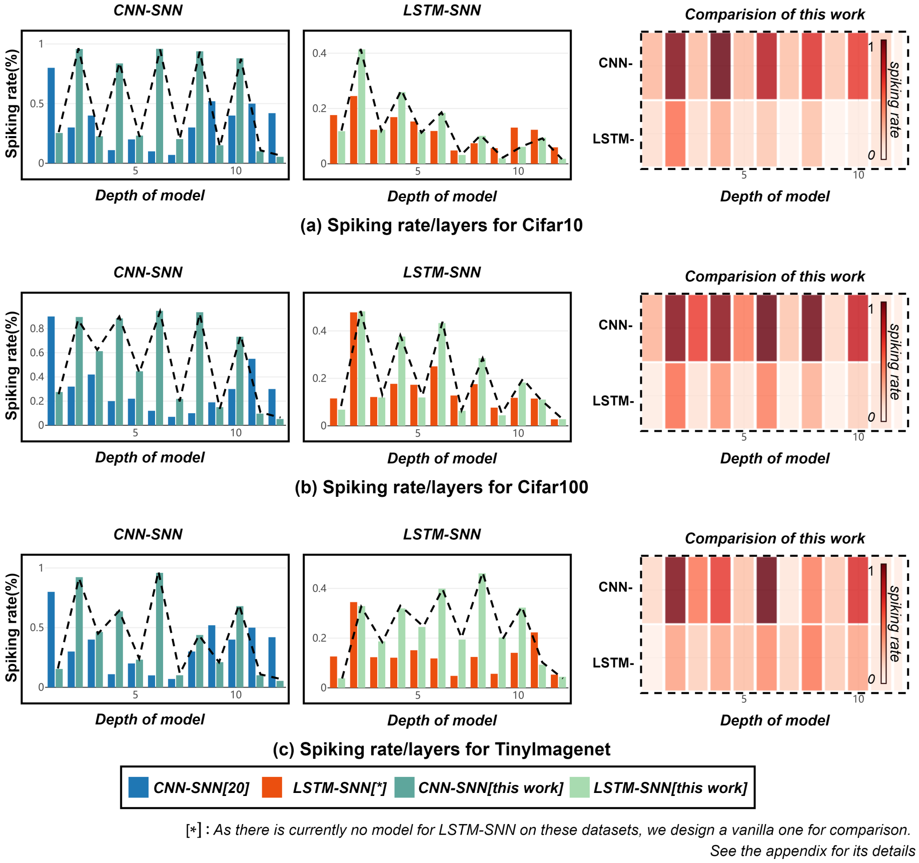
$$\#Spiking - Rate_l = \frac{\#TotalSpike_l \times Timesteps}{\#Neurons_l} \tag{13}$$

where $\#Spike - Rate$ is the total spikes in layer l over all times steps averaged over the number of neurons in layer l. Spike rate may exceed 1 in some studies (by over numerous time steps) implying that the number of operations for SNN exceeds the ANN (operations are MAC in ANN while still adding in SNNs). Explicitly, lower spike rates denote more sparsity in spike events, lower operations, and higher energy efficiency for SNN.

SNNs training form our method, the average spike rate is 0.54 and 0.32 for CNN-SNN and LSTM-SNN, respectively. Following previous work Horowitz [2014], we estimate the energy consumption of SNNs by comparing the energy consumption of SNN and ANN in 45 nm CMOS technology. The energy cost for a 32-bit ANN MAC operation (4.6 pJ) is 5.1 × higher than that of an SNN addition operation (0.9 pJ). The number of operations or layers in an ANN is defined by Eq. 14.

$$\#OP_{ANN} = \begin{cases} k_w \times k_h \times c_{in} \times h_{out} \times w_{out} \times c_{out} & , Convolution\ layer \\ (c_{in} + c_{out}) \times c_{out} \times 4 & , LSTM\ layer \\ f_{in} \times f_{out} & , Fully - connected\ layer \end{cases} \tag{14}$$

where $k_w(k_h)$ is kernel width (height), $c_{in}(c_{out})$ is the number of input (output) channels, $h_{out}(w_{out})$ is the height (width) of the output feature map, and $f_{in}(f_{out})$ is the channels of input (output) features.



(a) Spiking rate/layers for Cifar10

(b) Spiking rate/layers for Cifar100

(c) Spiking rate/layers for TinyImagenet

[*] : *As there is currently no model for LSTM-SNN on these datasets, we design a vanilla one for comparison. See the appendix for its details*

Figure 4: Comparison for spiking rate/layers based on various frameworks.

For an SNN, the number of operations per layer is given by $\#OP_{SNN,l} = \#Spiking - Rate_l \times \#OP_{ANN}$, where $\#OP_{SNN,l}$ represents the total number of MAC operations in layer l. Using these equations, the addition count is calculated by $s * T * \#OP_{ANN}$ in SNN, where s is the mean spike rate, and T is the time steps. For multiplication in the SNN, we set it to the MAC of the first layer and some fully-connected layers in our model and scale it by T. We calculate the operation number and energy consumption for both the ANN and SNN, which is shown in Table 2. Our model only costs 123 mJ for a single forward pass, which is 11 × lower in energy consumption than our ANN, and 102 × lower than Res-19 in the work Zheng et al. [2021]. As shown in Fig. 4, the proposed framework enables spike rate changes alternately between different layers. Leaving out the last block, the spike rate will be greatly enhanced in even layers (dilated window) and remain low in odd layers (non-dilated window). However, as the network is continuously halved, the feature maps in the last layers generally become small, and the dilated window may not be appropriately divided (such as the size is 2×2). Therefore, we adopt a non-dilated window in the last block of the framework.

From a spike rate perspective, we use a higher average spike rate than other methods. However, this higher spike rate is compensated for by our model's lightweight and low-latency

Table 3: Spike/image comparison between this work and SOTA SNN Tang et al. [2022].

Method	**CIFAR10**			**CIFAR100**			**Tiny-ImageNet**		
	Acc (%)	**Steps**	**Spikes/Image**	**Acc (%)**	**Steps**	**Spikes/Image**	**Acc (%)**	**Steps**	**Spikes/Image**
RNL-RIL	92.50	250	4.24×10^6	72.90	250	5.94×10^6	56.10	250	2.03×10^7
STBP-IF	84.20	5	1.11×10^6	57.77	4	1.35×10^6	54.53	3	2.17×10^6
	71.54	8	2.01×10^6	39.86	8	2.69×10^6			
STBP-PLIF	91.63	5	9.67×10^5	70.94	4	1.05×10^6	53.08	3	1.74×10^6
S2A-ReSU	92.62	5	1.68×10^6	71.10	4	6.69×10^5	54.91	3	1.02×10^6
S2A-STSU	92.18	5	4.52×10^5	68.96	4	6.18×10^5	54.33	3	1.11×10^6
Ours-CNN	93.07	1	$\mathbf{6.38 \times 10^4}$	72.41	1	$\mathbf{6.19 \times 10^4}$	57.91	1	$\mathbf{2.58 \times 10^5}$
Ours-LSTM	92.68	1	$\mathbf{2.13 \times 10^4}$	72.31	1	$\mathbf{2.44 \times 10^4}$	57.55	1	$\mathbf{1.85 \times 10^5}$

Table 4: Ablation study of proposed window partition.

CNN-SNN					LSTM-SNN				
Groups	g=1	g=2	g=4	Acc%)	Groups	g=1	g=2	g=4	Acc(%)
Normal Windows	✓			91.80	Normal Windows	✓			90.37
		✓		92.15			✓		90.02
			✓	92.28				✓	90.58
Dilated Windows	✓			91.80	Dilated Windows	✓			90.37
		✓		92.78			✓		91.82
			✓	**93.13**				✓	**92.58**

nature. When noticing the total amount of spike activity, as 12 network layers and only one-time step in our model, the total spike activity remains on a small scale, as shown in Table 3. In most cases, our SNNs achieve comparable accuracy with fewer spikes than other methods.

4.3. Ablations

We present an ablation study of our proposal, divided into two aspects corresponding to Sections 3.1 and 3.2. First, we ablate the substructure dilated windows of the proposed framework. Second, we experiment with other fusion functions and compare them to our proposed fusion function using different values of θ.

Dilated Window Ablations of the dilated window approach on Cifar10 for both frameworks are reported in Table 4. We first analyze the effect of windowing times (i.e., the size of i in G_m^i) on the CNN-based model without dilated window. Next, we verify the dilated window of different windowing times. Finally, we do the same experiment with the LSTM-SNN model.

From the results, we observe that reducing the number of windowing partitions leads to a decrease in performance, as we explained in Section 3.1. The multiple uses of membrane potential are positively correlated with the performance of SNN within a limited number of time steps. Additionally, we verify the effectiveness of our proposed Dilated Window approach. Due to the strengthened connection between different windows, its performance is better than that of the normal Window partition method.

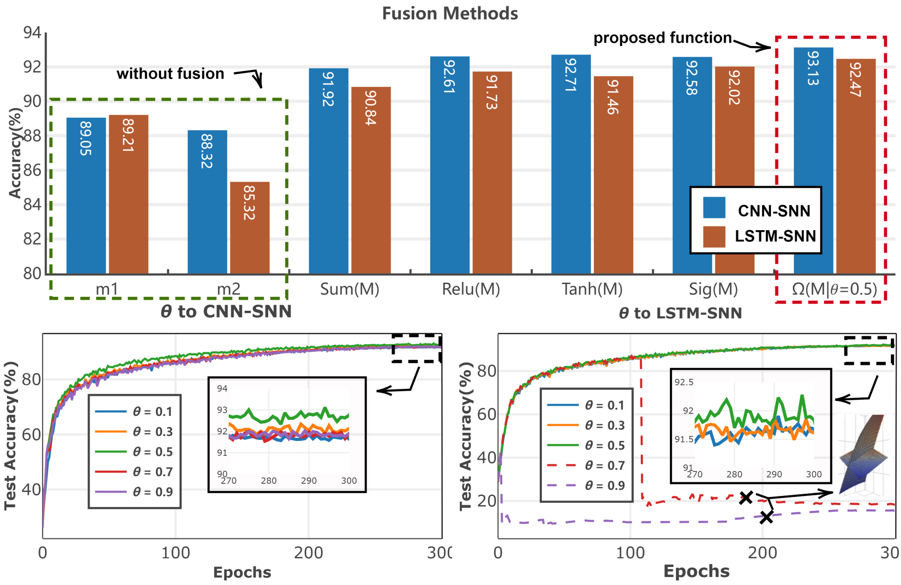


Figure 5: Ablation study of project functions and various θ.

Projection Function $\Theta(\cdot)$ To verify its effectiveness, the projection function is ablated in two aspects, as shown in Fig. 5. First, several mainstream nonlinear functions (Relu, Sigmoid, Tanh) are experimented with and compared to the proposed functions to explore an optimal fusion space for membrane potential. Second, the impact of different angles θ on the proposed projection function is investigated.

As shown in Fig. 5(a), the proposed function performs best under insufficient time step constraints. Creating a remarkable SNN through only one direction of membrane potential is challenging; other nonlinear functions have not yielded better results than the proposed function. This is because nonlinear functions avoid negative infinity of membrane potential, resulting in better performance than the linear function. Additionally, while over-activation can only be prevented from worsening in other functions, the symmetry of the trigonometric function allows neurons to inhibit membrane potential fusion within a certain range. In contrast, as shown in Fig. 5(b)(c), the model achieves the best convergence effect when $\theta = 0.5$. Varying θ will affect the final convergence of the model, as previously analyzed. However, it should be noted that the convergence of RNN-SNN is relatively unstable and may even fail, as indicated by the dotted lines. This is due to the rapid attenuation of membrane potential after fusion caused by excessive θ as illustrated in the auxiliary spatial map next to the dotted line. It should be noted that the threshold selection of LIF is critical for fusion. We chose $th = 0.5$ because it can make the double threshold not greater than the maximum value of the trigonometric function.

5. Conclusion & Discussion

In this paper, we propose a novel training framework for Spiking Neural Networks, which results in low inference latency from T time steps down to unity (1) while maintaining competitive performance. Through enhancing the utilization of membrane potential information in two directions, the learning ability of SNNs has been significantly improved in an extremely limited time step as shown in both experimental results and theoretical analysis. Furthermore, the proposed framework is also applicable to RNN-based models, achieving unprecedented results across multiple datasets. Although our model demonstrates significant performance improvement, it does not result in a more sparse activation spike matrix when compared to previous models. The advantages in energy consumption mainly arise from the extremely minimal time steps used in our work. Developing a more sparse model to directly deal with the inherent information loss in discrete spikes remains a problem. Additionally, the design of the projection function and θ is based on previous research and experience on spiking neurons, and may not be optimal. Other types of fusion methods for membrane potential are worth exploring further.

Acknowledgments

This research was supported by the JSPS KAKENHI Grant Number JP22K21285, Japan.

References

Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11–20, 2022.

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.

Zheng Chen, Lingwei Zhu, Ziwei Yang, and Renyuan Zhang. Multi-tier platform for cognizing massive electroencephalogram. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 2464–2470, 2022.

Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning. In *ECCV*, pages 709–726, 2022a.

Sayeed Shafayet Chowdhury, Nitin Rathi, and Kaushik Roy. Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 709–726. Springer, 2022b.

Gourav Datta, Haoqin Deng, Robert Aviles, and Peter A. Beerel. Towards energy-efficient, low-latency and accurate spiking lstms, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Lang Feng, Qianhui Liu, Huajin Tang, De Ma, and Gang Pan. Multi-level firing with spiking ds-resnet: Enabling better and deeper directly-trained spiking neural networks. *arXiv preprint arXiv:2210.06386*, 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13558–13567, 2020.

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

Eric Hunsberger and Chris Eliasmith. Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*, 2015.

Jaehyun Kim, Heesu Kim, Subin Huh, Jinho Lee, and Kiyoung Choi. Deep neural networks with weighted spikes. *Neurocomputing*, 311:373–386, 2018.

Mikhail Kiselev. Rate coding vs. temporal coding-is optimum between? In *2016 international joint conference on neural networks (IJCNN)*, pages 1355–1359. IEEE, 2016.

Souvik Kundu, Gourav Datta, Massoud Pedram, and Peter A Beerel. Spike-thrift: Towards energy-efficient deep spiking neural networks by limiting spiking activity via attention-guided compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2021.

Dongwoo Lew, Kyungchul Lee, and Jongsun Park. A time-to-first-spike coding and conversion aware training for energy-efficient deep spiking neural network processor design. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 265–270, 2022.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34:23426–23439, 2021.

Fangxin Liu, Wenbo Zhao, Yongbiao Chen, Zongwu Wang, and Li Jiang. 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*, volume 36, pages 1692–1701, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

Ali Lotfi Rezaabad and Sriram Vishwanath. Long short-term memory spiking networks and their applications. In *International Conference on Neuromorphic Systems 2020*, pages 1–9, 2020.

Byunggook Na, Jisoo Mok, Seongsik Park, Dongjin Lee, Hyeokjun Choe, and Sungroh Yoon. Autosnn: towards energy-efficient spiking neural networks. In *International Conference on Machine Learning*, pages 16253–16269. PMLR, 2022.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Wachirawit Ponghiran and Kaushik Roy. Spiking neural networks with improved inherent recurrence dynamics for sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8001–8008, 2022.

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*, pages 1–9, 2021a.

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*, 2021b.

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.

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

Jianxiong Tang, Jianhuang Lai, Xiaohua Xie, Lingxiao Yang, and Wei-Shi Zheng. Snn2ann: A fast and memory-efficient training framework for spiking neural networks. *arXiv preprint arXiv:2206.09449*, 2022.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Man Wu, Zheng Chen, and Yunpeng Yao. Learning local representation by gradient-isolated memorizing of spiking neural network. In *2022 IEEE 24th Int Conf on High Performance Computing & Communications; (HPCC)*, pages 733–740, 2022.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11062–11070, 2021.