Ternary Spike: Learning Ternary Spikes for Spiking Neural Networks

Yufei Guo*, **Yuanpei Chen***, **Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, Zhe Ma**†

Intelligent Science & Technology Academy of CASIC, China
yfguo@pku.edu.cn, rop477@163.com, mazhe_thu@163.com

Abstract

The Spiking Neural Network (SNN), as one of the biologically inspired neural network infrastructures, has drawn increasing attention recently. It adopts binary spike activations to transmit information, thus the multiplications of activations and weights can be substituted by additions, which brings high energy efficiency. However, in the paper, we theoretically and experimentally prove that the binary spike activation map cannot carry enough information, thus causing information loss and resulting in accuracy decreasing. To handle the problem, we propose a ternary spike neuron to transmit information. The ternary spike neuron can also enjoy the event-driven and multiplication-free operation advantages of the binary spike neuron but will boost the information capacity. Furthermore, we also embed a trainable factor in the ternary spike neuron to learn the suitable spike amplitude, thus our SNN will adopt different spike amplitudes along layers, which can better suit the phenomenon that the membrane potential distributions are different along layers. To retain the efficiency of the vanilla ternary spike, the trainable ternary spike SNN will be converted to a standard one again via a reparameterization technique in the inference. Extensive experiments with several popular network structures over static and dynamic datasets show that the ternary spike can consistently outperform state-of-the-art methods. Our code is open-sourced at https://github.com/yfguo91/Ternary-Spike.

Introduction

Nowadays, Artificial Neural Network (ANN) has been widely used in many fields, *e.g*., object recognition (He et al. 2016), object segmentation (Ronneberger, Fischer, and Brox 2015), object tracking (Bewley et al. 2016), etc. However, to achieve better performance, the network size is evolving to more and more complex (Huang et al. 2017; Devlin et al. 2018). To address this problem, quantization (Gong et al. 2019; Li, Dong, and Wang 2019), pruning (He, Zhang, and Sun 2017), knowledge distillation (Polino, Pascanu, and Alistarh 2018; Zhang, Bao, and Ma 2022), spiking neural networks (SNNs) (Li et al. 2021a; Xiao et al. 2021; Wang et al. 2022; Bohnstingl et al. 2022), and so on, have been proposed. Especially, SNNs, recognized as one of the next-generation neural networks, provide a unique way to reduce energy consumption by mimicking the information processing of the brain. SNNs adopt spikes to transmit information and thus can convert multiplications of weights and activations to additions, enjoying multiplication-free inference. Furthermore, the event-driven-based computation manner shows higher energy efficiency on neuromorphic hardwares (Ma et al. 2017; Akopyan et al. 2015; Davies et al. 2018; Pei et al. 2019).

However, SNN's binary spike activation maps suffer the limited information capacity compared to full precious activation maps of the ANN and are unable to carry enough useful information from membrane potentials in the quantization process of the SNN, thus causing information loss and resulting in accuracy decreasing. A detailed proof will be given in Sec. 4.1 in the paper. Meanwhile, we also find that the membrane potential distributions of different layers of an SNN are much different. thus quantizing these membrane potentials to the same spike values is unnatural, which is ignored by the prior work.

To solve these problems, we first propose the ternary spike neuron, called **Ternary Spike**. Different from the current way using $\{0, 1\}$ spikes, it utilizes the $\{-1, 0, 1\}$ spikes to transmit information. A detailed design and a theoretical proof will be provided in Sec. 4.1 that the **Ternary Spike** enjoys the greater information capacity than the binary spike but will keep the multiplication-addition transform and event-driven advantages still. Furthermore, we also extend the ternary spike to a learnable ternary spike form, which is not limited to $\{-1, 0, 1\}$, but $\{-\alpha, 0, \alpha\}$, where α is layer-wise learnable value. In this way, different layers' neurons will fire different magnitudes of spikes in a learning manner, corresponding to different membrane potential distributions. In the inference phase, the α factor can be folded into the weights by a re-parameterization technique, hence will retain the multiplication-free inference again. The difference between our ternary spike neuron and the vanilla binary spike neuron is illustrated in Fig. 1.

In summary, our main contributions are threefold:

- We prove that the binary spike activation map cannot carry enough information with theoretical justifications and in-depth experimental analysis and propose the ternary spike neuron, a simple yet effective improved spike neuron for SNNs to increase the information capacity. It still enjoys the multiplication-free and event-driven advantages of the

[*] Equal contribution.

[†] Corresponding author.

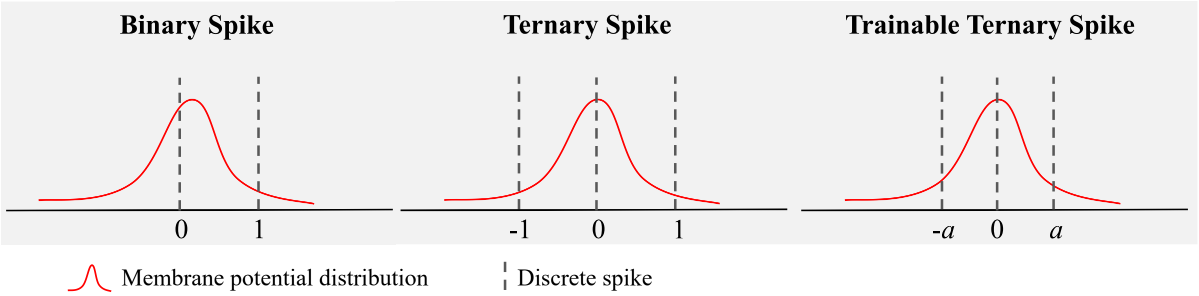


Figure 1: The difference between our ternary spike neuron and the vanilla binary spike neuron. In binary spike neurons, the membrane potentials will be quantized to $\{0, 1\}$ spikes, which will cause severe information loss. In our ternary spike neuron, the membrane potentials will be quantized to $\{-1, 0, 1\}$ spikes, which will boost the information capacity, while still enjoying the event-driven and multiplication-addition transformation advantages. In trainable spike neurons, the membrane potentials will be quantized to $\{-\alpha, 0, \alpha\}$ spikes, where α is learned in the training phase. While in the inference phase, the trainable ternary spike SNN can be converted to a standard ternary spike SNN and retain these advantages of standard SNN still.

standard SNN. Rather, the ternary spike neuron can be seen as a new paradigm of spike neurons.

- Furthermore, we also propose a learnable **Ternary Spike** version, where the spike magnitude, α can be learned in the training phase. While in the inference phase, the magnitude can be folded into the weights, thus the ternary spikes $\{-\alpha, 0, \alpha\}$ will be transformed to the standard ternary spikes $\{-1, 0, 1\}$ again and retain the addition-mostly advantage.
- We evaluate our methods on both static (CIFAR-10 (Krizhevsky, Nair, and Hinton 2010), CIFAR-100 (Krizhevsky, Nair, and Hinton 2010), ImageNet (Deng et al. 2009)) and spiking (CIFAR10-DVS (Li et al. 2017)) datasets with widely used backbones. Results show that the SNN trained with the proposed **Ternary Spike** is highly effective and efficient. *e.g.*, our method reach 70.74% top-1 accuracy on the ImageNet using ResNet34 with only 4 timesteps, about 3% improvement compared with other state-of-the-art SNN models.

Related Work

In this section, we briefly overview recent works of SNNs in two aspects: learning methods of SNNs and information loss in SNNs.

Learning Methods of Spiking Neural Networks

There are mainly two routers to obtain high-performance deep SNNs. The first is converting a well-trained ANN to an SNN, called ANN-SNN conversion (Han and Roy 2020; Li et al. 2021a; Kim et al. 2020; Han, Srinivasan, and Roy 2020; Liu et al. 2022; Li and Zeng 2022; Yu et al. 2021; Hao et al. 2023a,b). The principle of the ANN-SNN conversion method is to map the parameters from a pre-trained ANN model to its SNN counterpart, based on the matching of the ANN activation values and the SNN average firing rate. Since the training of an SNN is much more resource and time-consuming than that of the corresponding ANN, this method is one of the widely used ways to obtain a well-performed SNN. Nevertheless, There are still several inherent deficiencies in ANN-SNN conversion that are difficult to solve. First, it is limited in the rate-coding scheme and ignores the rich temporal dynamic behaviors from SNNs, thus it cannot handle these neuromorphic datasets well. Second, it usually requires many timesteps to approach the accuracy of pre-trained ANNs. This will increase energy consumption correspondingly, which is contrary to the original intention of SNN's low-power consumption design. Third, the SNN accuracy cannot exceed the ANN accuracy in this paradigm. This will limit the imagination and possibility of the SNN, thus reducing its research values.

Training SNNs directly from scratch is suitable for neuromorphic datasets and can greatly reduce timesteps, even less than 5 in recent work (Deng et al. 2022; Guo et al. 2022d; Fang et al. 2021; Wu et al. 2018; Rathi and Roy 2020; Wu et al. 2019; Neftci, Mostafa, and Zenke 2019). Hence this kind of method, as another router of SNN learning has received more research attention recently. Besides, the "hybrid" learning method (Rathi and Roy 2020; Wu et al. 2021), which combines the advantages of ANN-SNN conversion method and direct training method, has also drawn much attention recently. In this work, we focus on improving the performance of the directly training-based SNNs by reducing information loss, which is rarely studied in other works.

Information Loss in Spiking Neural Networks

There are some works involving reducing the information loss in SNNs(Guo et al. 2022c,a,b; Wang, Zhang, and Zhang 2023). However, they do not analyze the problem systematically. In InfLoR-SNN (Guo et al. 2022b), the quantization process of the SNN was thought would cause information loss, and then a membrane potential rectifier that can adjust the membrane potential to a new value closer to quantization spikes than itself before firing activity was proposed. IM-Loss (Guo et al. 2022a) argued that improving the activation information entropy can reduce the information loss, and proposed an information maximization loss function that can maximize the activation information entropy. In RecDis-SNN (Guo et al. 2022c), a loss for membrane potential distribution to explicitly penalize their undesired shifts was proposed. Though the work is not designed for reducing information loss, it still will result in a bimodal membrane potential distribution, which is proven can mitigate the problem. In MT-SNN (Wang, Zhang, and Zhang 2023), a multiple threshold (MT) algorithm was proposed at Leak-Integrate-Fire(LIF) neuron to partly recover the information loss in the quantization process of the SNN. Nevertheless, these above works all still quantize the membrane potentials to the binary spikes and ignore the difference in membrane potential distributions along layers. In the paper, we present the ternary spike neuron to transmit information with $\{-1, 0, 1\}$ spikes, which can improve the activation information capacity of the SNN and retain the multiplication-addition transform advantage at the same time. We also improve the proposed ternary spike neuron to a trainable ternary spike neuron, which treats the membrane potential distributions differently. Note that, a ternary spike neuron was also proposed in (Sun et al. 2022), however, this neuron transmits information with $\{0, 1, 2\}$ spikes, and cannot enjoy multiplication-addition transform advantage.

Preliminary

The brain-inspired spike neuron is the fundamental and special computing unit of SNNs. It mimics the behavior of the brain neuron and is described by the interaction of the membrane potential and input current. We consider the commonly used Leaky-Integrate-and-Fire (LIF) neuron model in the paper. It can be governed by the following iterative model (Wu et al. 2019),

$$u^{\text{before},t} = (1 - \frac{dt}{\tau})u^{t-1} + \frac{dt}{\tau}R \cdot I^t, \tag{1}$$

$$u^t = \begin{cases} 0, & \text{if } u^t \geq V_{\text{th}} \\ u^{\text{before},t}, & \text{otherwise} \end{cases}, \tag{2}$$

and

$$o^t = \begin{cases} 1, & \text{if } u^{\text{before},t} \geq V_{\text{th}} \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $u^{\text{before},t}$ and u^t are the membrane potential before and after the firing processing of the spiking neuron at the t-th timestep, o^t is the membrane potential and output spike at the t-th timestep, dt and τ are time constants, the product of I and R denotes the charging voltage by the input current and resistance corresponding to the previous layer signals, and V_{th} is the firing threshold. These above equations describe the behaviors of spike neurons in an updating-firing-resetting manner. When the membrane potential increases up to the firing threshold, the LIF spike neuron will fire a spike and reset the membrane potential as 0; otherwise, the neuron updates its membrane potential by the addition of the membrane potential after the leakage and the received charging voltage. Since the factor $(1 - \frac{dt}{\tau})$ is a constant, it can still be denoted as τ and is set as 0.25 in the paper as (Guo et al. 2022d,b). Considering the SNN is composed of connected spiking neurons with connection coefficients, the input voltage $\frac{dt}{\tau}R \cdot I^t$ can be expanded to the weighted summation of previous layer spike signals as $\sum_j w_j o^t_{j,\text{pre}}$, where w_j denotes the connected weight of the j-th previous layer neuron and current neuron, and $o^t_{j,\text{pre}}$ indicates the binary spike from the j-th previous layer neuron at t-th timestep. Further, the iterative LIF model can be simply updated as

$$u^t = \tau u^{t-1}(1 - o^{t-1}) + \sum_j w_j o^t_{j,\text{pre}}, \tag{4}$$

and

$$o^t = \begin{cases} 1, & \text{if } u^t \geq V_{\text{th}} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Methodology

Information Loss in Spiking Neural Networks

Though the binary spike information processing paradigm is highly energy efficient, it will also result in unsatisfactory task performance compared with ANNs. We think one of the most reasons is that the binary spike activation maps cannot carry enough information, thus will cause information loss. To verify our assumption, we first provide a theoretical analysis here by using the information entropy concept. Given a set, $\mathbf{S}$, its representation capability, $\mathcal{R}(\mathbf{S})$ can be measured by the information entropy of $\mathbf{S}$, as follows

$$\mathcal{R}(\mathbf{S}) = \max \mathcal{H}(\mathbf{S}) = \max(-\sum_{s \in \mathbf{S}} p_{\mathbf{S}}(s) log p_{\mathbf{S}}(s)), \tag{6}$$

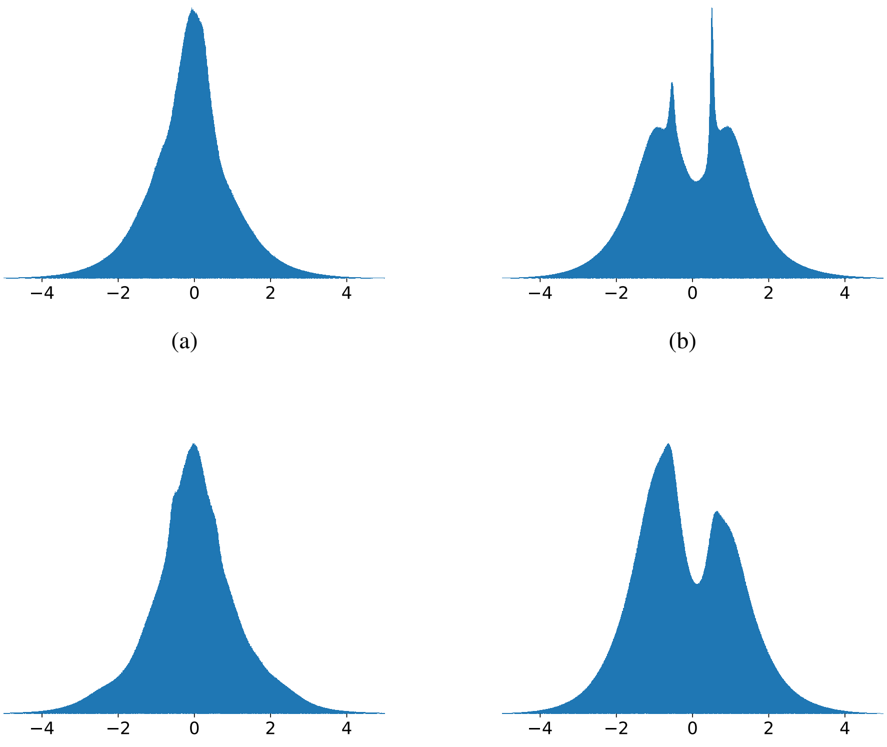
where $p_{\mathbf{S}}(s)$ is the probability of a sample, s from $\mathbf{S}$. Then we have the following proposition:

Proposition 1 *When* $p_{\boldsymbol{S}}(s_1) = p_{\boldsymbol{S}}(s_2) = p_{\boldsymbol{S}}(s_3) \cdots = p_{\boldsymbol{S}}(s_N)$, $\mathcal{H}(\boldsymbol{S})$ *reaches its maximum,* $log(N)$.

Here, N denotes the total number of the samples from $\mathbf{S}$. With this conclusion, we can calculate the representation capability of the binary spike feature map and the real-valued membrane potential map. Let $\mathbf{F}_B \in \mathbb{B}^{C \times H \times W}$ denote as a binary feature map and $\mathbf{M}_R \in \mathbb{R}^{C \times H \times W}$ denote as a real-valued membrane potential map. For a binary spike output o, it can be expressed with 1 bit, thus the number of samples from o is 2. Then, the number of samples from $\mathbf{F}_B$ is $2^{(C \times H \times W)}$ and $\mathcal{R}(\mathbf{F}_B) = \mathbf{log} 2^{(C \times H \times W)} = C \times H \times W$. While a real-valued membrane potential needs 32 bits, which consists of 2^{32} samples. Hence, $\mathcal{R}(\mathbf{M}_R) = \mathbf{log} 2^{32 \times (C \times H \times W)} = 32 \times C \times H \times W$. It is thus clear that the representation capability of binary spike feature map is much limited and quantizing the real-valued membrane potentials to binary spikes induces excessive information loss. A common consensus is that increasing the timesteps of the SNN can improve the accuracy. This can also be proved by our information theory here. Increasing the timesteps is equivalent to increasing the neuron output spike bits through the temporal dimension, thus increasing the representation capability of the output feature map.

Ternary Spike Neuron Model

The above theoretical analysis shows that improving the spike neuron activation information capacity can increase task performance. And one cannot discard the spike information processing paradigm to increase representation capability, otherwise, the event-driven and addition-only-based energy efficiency will be lost. To boost the information capacity



(a)

(b)

(c)

(d)

Figure 2: The distributions of potential membranes along layers for spiking ResNet20 on CIFAR-10. (a) and (b) show the distributions for the second and third layers of the network with 1 timestep respectively. (c) and (d) show the distributions for the second and third layers of the network with 2 timesteps respectively.

while keeping these advantages, here, we present a ternary LIF spike neuron, given by

$$u^t = \tau u^{t-1}(1 - |o^{t-1}|) + \sum_j w_j o^t_{j,\text{pre}}, \tag{7}$$

and

$$o^t = \begin{cases} 1, & \text{if } u^t \geq V_{\text{th}} \\ -1, & \text{if } u^t \leq -V_{\text{th}} \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

Representation capacity improvement of ternary spike neuron. We argue that firing ternary spikes can help increase the representation capacity of the SNNs. To verify our assumption, we also resort to the information entropy theory. Let $\mathbf{F}_T \in \mathbb{T}^{C\times H\times W}$ denote as a ternary feature map. The ternary spike $\mathbf{F}_T$ consists of $3^{C\times H\times W}$ samples. Hence, $\mathcal{R}(\mathbf{F}_T) = \mathbf{log}_2 3^{C\times H\times W}$, while $\mathcal{R}(\mathbf{F}_B) = C \times H \times W$, according to Eq. 6. Obviously, the representation capability of ternary spikes far exceeds that of binary spikes. This indicates that ternary spikes will enhance the information expressiveness of SNNs, which obviously benefits performance improvement.

Event-driven and addition-only advantages retaining. The SNN's event-driven signal processing characteristic makes it much energy-efficient. In specific, only if the membrane potential exceeds the firing threshold, V_{th}, the spiking neuron will present a signal and start subsequent computations, otherwise, it will keep silent. For the ternary spike neuron, it enjoys the event-driven characteristic too. Rather, only if the membrane potential is greater than V_{th} or less than $-V_{\text{th}}$, the ternary spike neuron will be activated to fire the 1 or -1 spikes. Multiplication-addition transform is another advantage of SNNs to keep energy-efficient. In a binary spike neuron, when a spike is fired, it will be multiplied by a weight connected to another neuron to transmit information, which can be expressed as

$$x = 1 \times w. \tag{9}$$

Since the spike amplitude is 1, the multiplication can be replaced by an addition operation as

$$x = 0 + w. \tag{10}$$

For a ternary spike neuron, since the spike is 1 or -1, the multiplication will be

$$x = 1 \times w, \text{or}, -1 \times \text{w}. \tag{11}$$

It can be replaced by an addition operation too as

$$x = 0 + w, \text{or}, 0 - \text{w}. \tag{12}$$

To conclude, the proposed ternary spike neuron will enhance the expression ability of the SNN, at the same time, will retain the event-driven and addition-only advantages as the vanilla SNN too.

Trainable Ternary Spike

Another problem as we aforementioned is that most prior work quantizes the membrane potentials to the same spike values. However, in the paper, we find that the membrane potential distributions along layers vary greatly. Here, we show the membrane potential distribution differences with some experiments. We train a spiking ResNet20 (He et al. 2016) with 1&2 timesteps on the CIFAR-10 (Krizhevsky, Nair, and Hinton 2010) dataset and show the different layers' potential membrane distributions in Fig. 2. It can be seen that the distributions are very different along layers, thus quantizing the different layer's membrane potentials to the same spike values is unreasonable. Consequently, we advocate that different layers' membrane potentials should be quantized to different spike values. Then, we present the trainable ternary spike neuron, where the firing spike amplitude can be learned in the training phase given by

$$u^t = \tau u^{t-1}(1 - |b^{t-1}|) + \sum_j w_j o^t_{j,\mathrm{pre}}, \tag{13}$$

and

$$o^t = \begin{cases} 1 \cdot a, & \text{if } u^t \geq V_{\mathrm{th}} \\ -1 \cdot a, & \text{if } u^t \leq -V_{\mathrm{th}} \\ 0 \cdot a, & \text{otherwise} \end{cases}, \tag{14}$$

where a is a trainable factor, $b \in \{-1, 0, 1\}$, and $o^t = a \cdot b^t$. With the learnable a, the proposed neuron can find a better spike amplitude and treat different layers' firing activity with different strategies. The trainable factor is set in a layer-wise manner in our SNN models, *i.e.*, $\boldsymbol{a} \in \mathbb{R}^{1\times1\times1}$.

Note that though the trainable factor manner we use in this paper is the same as that used in (Guo et al. 2022d), they are derived from different perspectives. We use the trainable factor to suit the difference of membrane potential distributions in the paper, while the trainable factor is used in (Guo et al. 2022d) to learn unshared convolution kernels.

However, a new problem will be introduced by the trainable spike that the multiplication of weight and activation cannot be transformed to addition and the advantage of computation efficiency of SNNs will be lost. To deal with the problem, we follow a training-inference decoupled technique (Guo et al. 2022d), which can convert the different amplitude spikes into the same normalized spike in the inference phase by a re-parameterization technique, thus these advantages of the normalized spike will be retained still. Here, we take the convolution layer to introduce the re-parameterization technique.

Re-parameterization technique. For a convolution layer, we denote its input feature map and output feature map as $\mathbf{F}$ and $\mathbf{G}$ respectively. In the convolution layer, the input map will be convolved with a group of convolution kernels to form the output feature map, which can be written as

$$\mathbf{G} = \mathbf{K} * \mathbf{F}, \tag{15}$$

where $\mathbf{K}$ is the convolution kernel tensor and $(*)$ is the convolution operation.

Architecture	Method	Time-step	Accuracy
ResNet18	Binary spike	2	58.30%
	Ternary spike	2	**65.87%**
	Trainable ternary spike	2	**66.40%**
	Binary spike	4	61.07%
	Ternary spike	4	**66.90%**
	Trainable ternary spike	4	**67.68%**
ResNet34	Binary spike	2	62.81%
	Ternary spike	2	**69.48%**
	Trainable ternary spike	2	**69.51%**
	Binary spike	4	63.82%
	Ternary spike	4	**70.12%**
	Trainable ternary spike	4	**70.74%**

Table 1: Ablation study for the ternary spike on ImageNet.

For standard ternary SNNs, the input map consists of normalized ternary spikes. While in trainable ternary SNNs, the SNN is trained with real-valued spikes for the purpose of measuring the difference of membrane potential distributions. In this case, the input feature map can be denoted as below

$$\mathbf{F} = a \cdot \mathbf{B}. \tag{16}$$

In inference, we can extract a part of the values from $\mathbf{F}$ and fold them into the convolution kernels for the trained SNN. Then a new SNN that can emit normalized ternary spikes will be obtained without changing the values of the output maps. This process can be illustrated as follows:

$$\mathbf{G} = \mathbf{K} * (a \cdot \mathbf{B}) = (a \cdot \mathbf{K}) * \mathbf{B} = \tilde{\mathbf{K}} * \mathbf{B}, \tag{17}$$

where $\tilde{\mathbf{K}}$ is the transformed convolution kernel tensor. The re-parameterization provides us with a solution to convert a trainable ternary spike-based SNN into an output-invariant normalized ternary spike-based SNN by decoupling the training-time SNN and inference-time SNN.

Experiment

We performed extensive experiments to evaluate the proposed **Ternary Spike** and compared it with other recent SoTA methods over several widely used architectures including spiking ResNet20 (Rathi and Roy 2020; Sengupta et al. 2019) and ResNet19 (Zheng et al. 2021) on CIFAR-10(100) (Krizhevsky, Nair, and Hinton 2010), ResNet18 (Fang et al. 2021) and ResNet34 (Fang et al. 2021) on ImageNet, and ResNet20 and ResNet19 on CIFAR10-DVS (Li et al. 2017).

Ablation Study

We first conducted several ablation experiments to verify the effectiveness of the proposed ternary spike neuron model on the ImageNet dataset using ResNet18 and ResNet34 as the backbone respectively under different timesteps. The results are shown in Tab. 1. It can be seen that the highest accuracy for vanilla ResNet18 and ResNet34 are 61.07% and 63.82%, similar to existing works. If we use the ternary spike neuron, the performance would boost to 66.90% and 70.12%, which are huge improvements (about 6.0%). Moreover, with the

Dataset	Method	Type	Architecture	Timestep	Accuracy
CIFAR-10	DSR (Meng et al. 2022)	SNN training	ResNet18	20	95.40%
	KDSNN (Xu et al. 2023)	SNN training	ResNet18	4	93.41%
	Diet-SNN (Rathi and Roy 2020)	SNN training	ResNet20	5	91.78%
				10	92.54%
	Dspike (Li et al. 2021b)	SNN training	ResNet20	2	93.13%
				4	93.66%
	STBP-tdBN (Zheng et al. 2021)	SNN training	ResNet19	2	92.34%
				4	92.92%
	TET (Deng et al. 2022)	SNN training	ResNet19	2	94.16%
				4	94.44%
	RecDis-SNN (Guo et al. 2022c)	SNN training	ResNet19	2	93.64%
				4	95.53%
	Real Spike (Guo et al. 2022d)	SNN training	ResNet19	2	95.31%
				4	95.51%
			ResNet20	4	91.89%
	Ternary Spike	SNN training	ResNet19	1	**95.28%**±0.10
				2	**95.60%**±0.09
			ResNet20	2	**94.29%**±0.08
				4	**94.46%**±0.08
	Trainable Ternary Spike	SNN training	ResNet19	1	**95.58%**±0.08
				2	**95.80%**±0.10
			ResNet20	2	**94.48%**±0.09
				4	**94.96%**±0.10
CIFAR-100	RMP (Han, Srinivasan, and Roy 2020)	ANN2SNN	ResNet20	2048	67.82%
	Real Spike (Guo et al. 2022d)	SNN training	ResNet20	5	66.60%
	LTL (Yang et al. 2022)	Tandem Learning	ResNet20	31	76.08%
	Diet-SNN (Rathi and Roy 2020)	SNN training	ResNet20	5	64.07%
	RecDis-SNN (Guo et al. 2022c)	SNN training	ResNet19	4	74.10%
	Dspike (Li et al. 2021b)	SNN training	ResNet20	2	71.68%
				4	73.35%
	TET (Deng et al. 2022)	SNN training	ResNet19	2	72.87%
				4	74.47%
	Ternary Spike	SNN training	ResNet19	1	**78.13%**±0.11
				2	**79.66%**±0.08
			ResNet20	2	**73.00%**±0.08
				4	**73.85%**±0.11
	Trainable Ternary Spike	SNN training	ResNet19	1	**78.45%**±0.08
				2	**80.20%**±0.10
			ResNet20	2	**73.41%**±0.12
				4	**74.02%**±0.08

Table 2: Comparison with SoTA methods on CIFAR-10(100).

trainable ternary spike neuron, the models even further get another performance lift, amounting to 67.68% and 70.74% final accuracy, respectively.

Comparison with SoTA Methods

In this section, we compared our method with the prior SoTA works. We report the top-1 accuracy results with the mean accuracy and standard deviation of 3 trials. We first evaluated our method on CIFAR-10 and CIFAR-100 datasets. The results are summarized in Tab. 2. On the CIFAR-10 dataset, the highest accuracy from prior work using ResNet19 and ResNet20 as backbones are 95.51% and 93.66% respectively. While our ternary spike achieves 95.60% and 94.46% with fewer timesteps. With the trainable ternary spike, Our SNN models can reach a better accuracy. On the CIFAR-100 dataset, our trainable ternary spike using the ResNet19 and ResNet20 with only 2 timesteps outperforms the current best method, TET and RecDis-SNN by about 7% but with 4 timesteps. These experimental results clearly show our method's efficiency and effectiveness.

Next, we conducted experiments on the ImageNet dataset, a more complex dataset than CIFAR. The comparison results

Method	Type	Architecture	Timestep	Accuracy
RecDis-SNN (Guo et al. 2022c)	SNN training	ResNet34	6	67.33%
GLIF (Yao et al. 2022)	SNN training	ResNet34	4	67.52%
DSR (Meng et al. 2022)	SNN training	ResNet18	50	67.74%
Real Spike (Guo et al. 2022d)	SNN training	ResNet18	4	63.68%
		ResNet34	4	67.69%
SEW ResNet (Fang et al. 2021)	SNN training	ResNet18	4	63.18%
		ResNet34	4	67.04%
Ternary Spike	SNN training	ResNet18	4	**66.90%**±0.19
		ResNet34	4	**70.12%**±0.15
Trainable Ternary Spike	SNN training	ResNet18	4	**67.68%**±0.13
		ResNet34	4	**70.74%**±0.11

Table 3: Comparison with SoTA methods on ImageNet.

Method	Type	Architecture	Timestep	Accuracy
GLIF (Yao et al. 2022)	SNN training	7B-wideNet	16	78.10%
RecDis-SNN (Guo et al. 2022c)	SNN training	ResNet19	10	72.42%
Real Spike (Guo et al. 2022d)	SNN training	ResNet19	10	72.85%
Ternary Spike	SNN training	ResNet19	10	**78.40%**±0.21
		ResNet20	10	**78.70%**±0.17
Trainable Ternary Spike	SNN training	ResNet19	10	**79.80%**±0.16
		ResNet20	10	**79.80%**±0.19

Table 4: Comparison with SoTA methods on CIFAR10-DVS.

are shown in Tab. 3. There are many SoTA baselines proposed on this dataset recently. For example, RecDis-SNN (Guo et al. 2022c), GLIF (Yao et al. 2022), DSR (Meng et al. 2022), Real Spike (Guo et al. 2022d), and SEW ResNet (Fang et al. 2021) obtain 67.33%, 67.52%, 67.74%, 67.69%, and 67.04%, respectively. Yet our method still achieves higher accuracy up to 70.74%, which is a noteworthy improvement. This shows the effectiveness of our method in handling the large-scale dataset.

Finally, we also ran our SNN on the widely used neuromorphic dataset, CIFAR10-DVS. With ResNet19 and ResNet20 as the backbone, our method scores 79.80% and 79.80% accuracy, close to 80%, also a huge improvement.

Energy Estimation

In this section, we measure the hardware energy cost of the binary spike and ternary spike using ResNet20 on CIFAR10 with 2 timesteps for one image inference. Since the first rate-encoding layer does not enjoy the multiplication-free, it will produce the FLOPs (floating point operations). While other layers are calculated by SOPs (synaptic operations). The SOPs are calculated by $s \times T \times A$, where s is the mean sparsity, T is the timestep and A is the addition number in ANN. The sparsity for the SNN with the binary spike is 16.42%, while for the SNN with the ternary spike is 18.27%. Note that we see both -1 and 1 as firing. The sign function is only in LIFs, the number of LIFs is limited compared to that of convolution operations. We calculate the energy following (Hu, Tang, and Pan 2021), that one FLOP needs 12.5pJ, one SOP needs 77fJ, and one Sign (using Energy per spike to calculate) needs 3.7pJ. We put the summary of energy cost in Tab. 5. It can be seen that only 2.11% extra cost is added for the ternary spike.

Method	#Flops	#Sops	#Sign	Energy
Binary Spike	3.54M	71.20M	0.11M	50.14uJ
Ternary Spike	3.54M	79.21M	0.23M	51.20uJ

Table 5: Energy estimation.

Conclusion

In the paper, we proved that the binary spike activation map of the SNN cannot carry enough information with theoretical justifications and in-depth experimental analysis. To mitigate the problem, we proposed a ternary spike neuron. This neuron can increase the information capacity greatly and enjoys event-driven and addition-only advantages still. we also presented an improved ternary spike neuron by embedding a trainable factor in it to learn the suitable spike amplitude. With learned spike values, this kind of neuron can better suit the different membrane potential distributions along layers. Furthermore, these different learned spikes can be converted to standard binary spikes in the inference, thus will still enjoy the inherent advantages of SNNs. We conducted various experiments to verify the effectiveness of our method.

References

Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Merolla, P.; Imam, N.; Nakamura, Y.; Datta, P.; Nam, G.-J.; et al. 2015. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10): 1537–1557.

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, 3464–3468. IEEE.

Bohnstingl, T.; Woźniak, S.; Pantazi, A.; and Eleftheriou, E. 2022. Online Spatio-Temporal Learning in Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, S.; Li, Y.; Zhang, S.; and Gu, S. 2022. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting. *arXiv preprint arXiv:2202.11946*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.

Gong, R.; Liu, X.; Jiang, S.; Li, T.; Hu, P.; Lin, J.; Yu, F.; and Yan, J. 2019. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4852–4861.

Guo, Y.; Chen, Y.; Zhang, L.; Liu, X.; Wang, Y.; Huang, X.; and Ma, Z. 2022a. IM-Loss: Information Maximization Loss for Spiking Neural Networks. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Guo, Y.; Chen, Y.; Zhang, L.; Wang, Y.; Liu, X.; Tong, X.; Ou, Y.; Huang, X.; and Ma, Z. 2022b. Reducing Information Loss for Spiking Neural Networks. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 36–52. Cham: Springer Nature Switzerland. ISBN 978-3-031-20083-0.

Guo, Y.; Tong, X.; Chen, Y.; Zhang, L.; Liu, X.; Ma, Z.; and Huang, X. 2022c. RecDis-SNN: Rectifying Membrane Potential Distribution for Directly Training Spiking Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 326–335.

Guo, Y.; Zhang, L.; Chen, Y.; Tong, X.; Liu, X.; Wang, Y.; Huang, X.; and Ma, Z. 2022d. Real spike: Learning real-valued spikes for spiking neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, 52–68. Springer.

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

Han, B.; Srinivasan, G.; and Roy, K. 2020. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13558–13567.

Hao, Z.; Bu, T.; Ding, J.; Huang, T.; and Yu, Z. 2023a. Reducing ANN-SNN Conversion Error through Residual Membrane Potential. arXiv:2302.02091.

Hao, Z.; Ding, J.; Bu, T.; Huang, T.; and Yu, Z. 2023b. Bridging the Gap between ANNs and SNNs by Calibrating Offset Spikes. arXiv:2302.10685.

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

He, Y.; Zhang, X.; and Sun, J. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, 1389–1397.

Hu, Y.; Tang, H.; and Pan, G. 2021. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11270–11277.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2010. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4): 1.

Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11: 309.

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

Li, Y.; Dong, X.; and Wang, W. 2019. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. *arXiv preprint arXiv:1909.13144*.

Li, Y.; Guo, Y.; Zhang, S.; Deng, S.; Hai, Y.; and Gu, S. 2021b. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 23426–23439.

Li, Y.; and Zeng, Y. 2022. Efficient and Accurate Conversion of Spiking Neural Network with Burst Spikes. *arXiv preprint arXiv:2204.13271*.

Liu, F.; Zhao, W.; Chen, Y.; Wang, Z.; and Jiang, L. 2022. SpikeConverter: An Efficient Conversion Framework Zipping the Gap between Artificial Neural Networks and Spiking Neural Networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 1692–1701. AAAI Press.

Ma, D.; Shen, J.; Gu, Z.; Zhang, M.; Zhu, X.; Xu, X.; Xu, Q.; Shen, Y.; and Pan, G. 2017. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of Systems Architecture*, 77: 43–51.

Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; and Luo, Z.-Q. 2022. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12444–12453.

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

Pei, J.; Deng, L.; Song, S.; Zhao, M.; Zhang, Y.; Wu, S.; Wang, G.; Zou, Z.; Wu, Z.; He, W.; et al. 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572(7767): 106–111.

Polino, A.; Pascanu, R.; and Alistarh, D. 2018. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*.

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

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

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

Sun, C.; Chen, Q.; Fu, Y.; and Li, L. 2022. Deep spiking neural network with ternary spikes. In *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 251–254. IEEE.

Wang, S.; Schmutz, V.; Bellec, G.; and Gerstner, W. 2022. Mesoscopic modeling of hidden spiking neurons. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Wang, X.; Zhang, Y.; and Zhang, Y. 2023. MT-SNN: Enhance Spiking Neural Network with Multiple Thresholds. arXiv:2303.11127.

Wu, J.; Chua, Y.; Zhang, M.; Li, G.; Li, H.; and Tan, K. C. 2021. A Tandem Learning Rule for Effective Training and Rapid Inference of Deep Spiking Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*.

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

Xiao, M.; Meng, Q.; Zhang, Z.; Wang, Y.; and Lin, Z. 2021. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *Advances in Neural Information Processing Systems*, 34: 14516–14528.

Xu, Q.; Li, Y.; Shen, J.; Liu, J. K.; Tang, H.; and Pan, G. 2023. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. *arXiv preprint arXiv:2304.05627*.

Yang, Q.; Wu, J.; Zhang, M.; Chua, Y.; Wang, X.; and Li, H. 2022. Training Spiking Neural Networks with Local Tandem Learning. *arXiv preprint arXiv:2210.04532*.

Yao, X.; Li, F.; Mo, Z.; and Cheng, J. 2022. GLIF: A Unified Gated Leaky Integrate-and-Fire Neuron for Spiking Neural Networks. *arXiv preprint arXiv:2210.13768*.

Yu, Q.; Ma, C.; Song, S.; Zhang, G.; and 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*, PP(99): 1–13.

Zhang, L.; Bao, C.; and Ma, K. 2022. Self-Distillation: Towards Efficient and Compact Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8): 4388–4403.

Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; and Li, G. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11062–11070.