



OPEN ACCESS

EDITED BY
Manolis Sifalakis,
Imec, Netherlands

REVIEWED BY
Octavian Melnic,
Micron, United States
Jesse Hagenaars,
Delft University of Technology, Netherlands

*CORRESPONDENCE
Xiaoxin Cui
✉ cuixx@pku.edu.cn
Yuan Wang
✉ wangyuan@pku.edu.cn

RECEIVED 14 June 2024
ACCEPTED 19 November 2024
PUBLISHED 17 December 2024

CITATION
Zou C, Cui X, Feng S, Chen G, Zhong Y, Dai Z
and Wang Y (2024) An all integer-based
spiking neural network with dynamic
threshold adaptation.
Front. Neurosci. 18:1449020.
doi: 10.3389/fnins.2024.1449020

COPYRIGHT
© 2024 Zou, Cui, Feng, Chen, Zhong, Dai and
Wang. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

An all integer-based spiking neural network with dynamic threshold adaptation

Chenglong Zou^{1,2}, Xiaoxin Cui^{3*}, Shuo Feng³, Guang Chen³,
Yi Zhong³, Zhenhui Dai³ and Yuan Wang^{3*}

¹Peking University Chongqing Research Institute of Big Data, Chongqing, China, ²School of Mathematical Science, Peking University, Beijing, China, ³School of Integrated Circuits, Peking University, Beijing, China

Spiking Neural Networks (SNNs) are typically regarded as the third generation of neural networks due to their inherent event-driven computing capabilities and remarkable energy efficiency. However, training an SNN that possesses fast inference speed and comparable accuracy to modern artificial neural networks (ANNs) remains a considerable challenge. In this article, a sophisticated SNN modeling algorithm incorporating a novel dynamic threshold adaptation mechanism is proposed. It aims to eliminate the spiking synchronization error commonly occurred in many traditional ANN2SNN conversion works. Additionally, all variables in the proposed SNNs, including the membrane potential, threshold and synaptic weights, are quantized to integers, making them highly compatible with hardware implementation. Experimental results indicate that the proposed spiking LeNet and VGG-Net achieve accuracies exceeding 99.45% and 93.15% on the MNIST and CIFAR-10 datasets, respectively, with only 4 and 8 time steps required for simulating one sample. Due to this all integer-based quantization process, the required computational operations are significantly reduced, potentially providing a substantial energy efficiency advantage for numerous edge computing applications.

KEYWORDS

spiking neural network, dynamic threshold adaptation, ANN2SNN conversion, network quantization, neuromorphic computing

1 Introduction

In recent years, neuromorphic computing based on spiking neural networks (SNNs) (Bouvier et al., 2019) has attracted more and more attention across computer vision (CV), automatic speech recognition (ASR) and natural language processing (NLP) tasks. SNNs are a special type of neural network that operate based on the principles of biological neural systems (Kasabov, 2019). At the functional level, unlike traditional artificial neural networks (ANNs) that use continuous-valued activation functions, SNNs employ spiking neurons that communicate and compute through discrete (0 or 1) spikes. These spikes represent the neural activity and information transmission in the network. At the structure level, SNNs are usually composed of LIF neurons (Andrew, 2003), each of which has a membrane potential which integrates incoming signals from other neurons. When the membrane potential exceeds a certain threshold, the neuron will generate a spike, and then propagated to other connected neurons. This spiking behavior allows SNNs to capture rich spatio-temporal dynamics and feature patterns in data, making them particularly suitable for processing time-varying signals and sequences.

In general, SNNs can offer several advantages compared to traditional ANNs. Firstly, they have a more biologically plausible representation of neural computation, which can lead to better generalization and adaptability to many complex tasks (Schuman et al., 2017). Secondly, SNNs are typically more energy-efficient because they usually communicate through sparse spikes and compute in an event-driven style (Taherkhani et al., 2020). This could significantly reduce the amount of data transfer and computation energy requirements (Bouvier et al., 2019; Schuman et al., 2017).

However, on the one hand, training SNNs remains a big challenge due to their discrete and event-driven characteristics. Traditional gradient-based optimization methods used in ANNs are not directly applicable to SNNs (Taherkhani et al., 2020; Tavanaei et al., 2019). Therefore, researchers have developed specialized learning algorithms and techniques to train SNNs effectively, such as spike-timing-dependent plasticity (STDP) (Neftci et al., 2019), surrogate gradient learning (SGL) (Lee et al., 2019) and ANN2SNN conversion (Dampfhofer et al., 2023). Direct SGL algorithms originate from the training strategy of the recurrent neural networks (RNN) (Hochreiter and Schmidhuber, 1997) and can be directly applied on some complex network architectures, such as residual neural network (ResNet) (He et al., 2016) with batch normalization (BN) (Ioffe and Szegedy, 2015). However, their performance is limited by the fitness of the gradient

approximation function, the training process is usually time or memory-consuming. Local STDP algorithm and its variants are only suitable for shallow learning and couldn't achieve end-to-end training for deep networks. ANN2SNN conversion is a kind of two-stage SNN modeling method, which can convert a decent ANN trained with back-propagation (BP) algorithm (Li et al., 2021) to an equivalent SNN based on the firing rate approximation. However, these converted SNNs usually suffer from uncertain accuracy loss (Bodo et al., 2017) when tested on some larger-scale datasets such as CIFAR-10 (Krizhevsky and Hinton, 2009) or ImageNet (Russakovsky et al., 2015) within fewer time steps.

On the other hand, only when SNNs are deployed on some specialized neuromorphic hardware such as TrueNorth (Esser et al., 2016) and Loihi (Massa et al., 2020) chip, can they really show the advantages of high energy efficiency. This means the low-bit quantization of synaptic weights, firing thresholds and leakage terms of an SNN must be necessary. However, many SNN works (Dampfhofer et al., 2023; Li et al., 2021; Bodo et al., 2017) only concentrate on the improvement of inference accuracy and speed, but ignore the hardware friendliness of their models. In this article, we present a delicate SNN architecture with dynamic threshold adaptation mechanism, to eliminate the common synchronization errors existed in many other ANN2SNN conversion works. Besides, all SNN parameters including membrane potential and firing threshold can be quantized to integers, which is very friendly

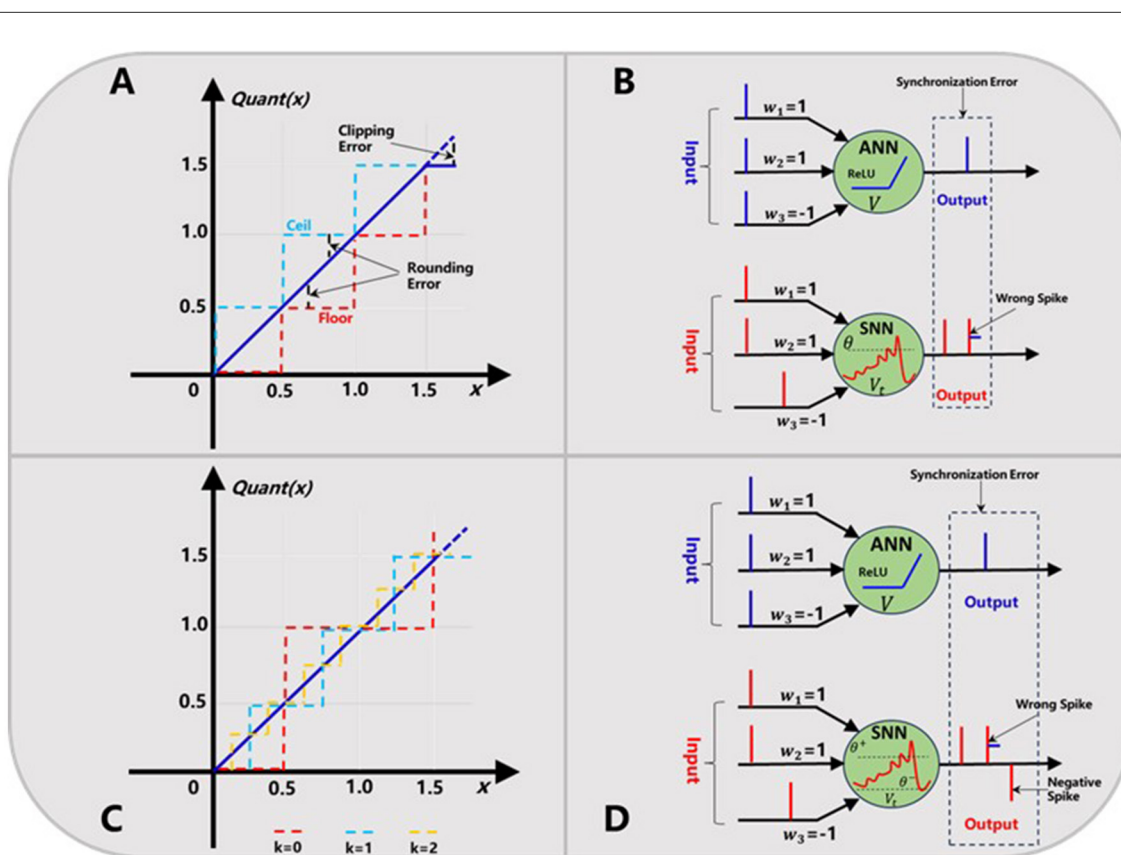
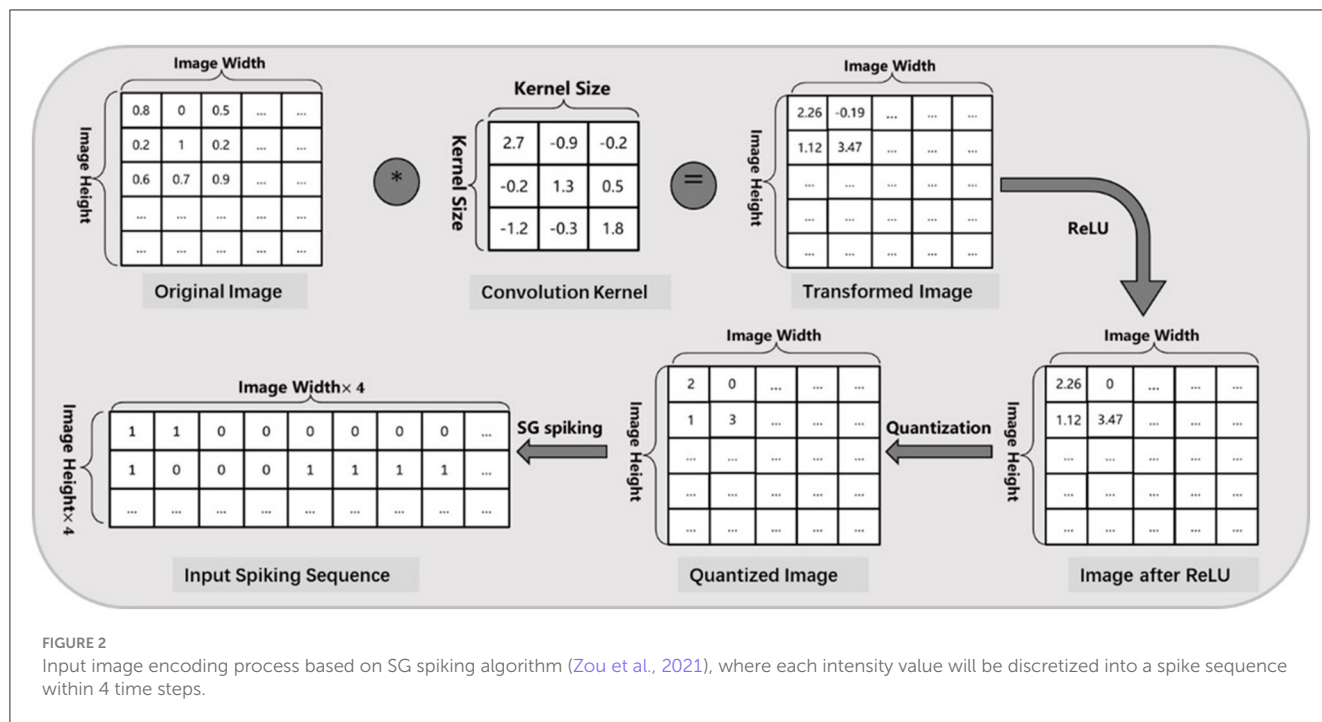


FIGURE 1 Quantization error (A) and synchronization error (B) in ANN2SNN conversion. Trainable quantization function (C) and dual-threshold spiking mechanism (D) for solving the above two problems, respectively.



for hardware implementation. Experimental results show that the proposed spiking LeNet (Lecun and Bottou, 1998) and VGG-Net (Simonyan and Zisserman, 2014) can obtain more than 99.45% and 93.15% classification accuracy on MNIST (Lecun and Bottou, 1998) and CIFAR-10 (Krizhevsky and Hinton, 2009) datasets with only 4 and 8 time steps, respectively.

The rest of this article is organized as follows. Section II outlines the details of the proposed ANN2SNN conversion algorithm, Section III introduces the parameter quantization and spike input encoding techniques. Section IV provides the experiment results and section V draws the conclusion.

2 ANN2SNN conversion

In this section, we would first briefly analyze the two kinds of common conversion loss among many previous ANN2SNN works. And then, a novel dual-threshold spiking approach which is also called dynamic threshold adaptation, is introduced to eliminate these errors from the source.

2.1 Conversion loss analysis

Conventional ANN2SNN conversion algorithms such as the weight normalization (Bodo et al., 2017; Diehl et al., 2015) and threshold normalization (Xu et al., 2017; Zou et al., 2020) inevitably suffer from some accuracy loss, due to the commonly existed quantization error and synchronization error. As shown in Figure 1A, the quantization error usually includes the rounding error and clipping error. The rounding error can be further divided into the ceiling and flooring error depending on the actual approximation method between the ReLU activation (Glorot et al.,

2011) of ANN neuron and the firing rate of SNN neuron. The clipping error occurs because when the upper bound of the ReLU activation function is usually truncated to achieve a fast simulation within fewer time steps (Chen et al., 2022).

Then, the synchronization error is also called sequential error in other work (Chen et al., 2022; Hu et al., 2023), which is caused by the difference of sequential firing mechanism of spikes in SNNs and static activation propagation in ANNs. Usually, the synchronization error is accumulated in higher layers at some earlier time steps, and causes the serious output mismatch between ANNs and their SNN counterparts. For example, as shown in Figure 1B, if we set the firing threshold to 1 and use a threshold subtraction scheme after a spike generation, this SNN neuron will fire for twice. This means a wrong spike will be generated, which is obviously not equivalent with the original ANN neuron outputs. This may be a more fundamental and much tougher problem which needs to be addressed.

Generally, both quantization error and synchronization error would degrade the accuracy performance to some extent in many previous converted SNNs (Diehl et al., 2015; Xu et al., 2017). As far as we know, the quantization error including the rounding error and clipping error can be alleviated by means of some quantization-aware training (QAT) methods (Chen et al., 2022; Hu et al., 2023) and longer simulation time, however the synchronization error serves as an inherent attribute in ANN2SNN conversion algorithms, which is key problem to be dealt with.

2.2 Dynamic threshold adaptation

To overcome the effects of the synchronization error, many researchers have tried various useful ways. For example, Bodo et al. (2017) and Sengupta et al. (2019) adopt a conventional method

TABLE 1 Classification accuracy for LeNet of different configurations on MNIST.

LeNet		Weight quantization	Threshold quantization	Final accuracy	Final time step
Architecture					
ANN		×	-	99.45%	-
		✓		99.42%	
SNN	k = 0	×	×	99.44%	7
		✓	×	99.40%	6
		×	✓	99.41%	6
		✓	✓	99.42%	6
	k = 1	×	×	99.41%	13
		✓	×	99.40%	11
		×	✓	99.45%	12
		✓	✓	99.43%	10
k = 2	×	×	99.49%	25	
	✓	×	99.42%	22	
	×	✓	99.43%	24	
	✓	✓	99.45%	21	

The **bold** number represents the best in the column.

of increasing simulation time steps to cover up the wrongly fired spikes. This method could improve the final accuracy to some extent, but bring longer simulation latency. Zou et al. (2020) and Meng et al. (2022) regularizes the input spikes of the first SNN layer to obtain a more uniform spike sequence representation. However, for much deeper networks, the spiking activities in the middle layers of SNNs are usually very complicated and exactly unpredictable and thus these spikes are almost impossible to be regularized.

In this work, we present a novel dual-threshold spiking approach together with a median quantization constraint to eliminate the two errors described above simultaneously. Firstly, each ReLU output value will be quantized with a hyper-parameter k called quantization level as Equation (1). It should be noted this parameter is trainable to minimize the quantization error and determines the quantization precision of ANN outputs as in Figure 1C. Then, we can convert this ANN to a firing-rate based SNN based on the following procedures. As in Equation 2–4, the converted spiking neuron works as the normal LIF behavior (Andrew, 2003), but has two thresholds i.e. θ^+ and θ^- , which determines if a positive or negative spike will be generated respectively. What's different is that these two thresholds are dynamic along with simulation time step t and will be updated synchronously. This dual-threshold spiking mechanism can be further elaborated in Figure 1D, where one negative spike is produced to correct the wrongly fired spike when the membrane potential V_t falls below the negative threshold θ^- .

$$Quant(y) = \frac{Round(y \times 2^k)}{2^k} \tag{1}$$

$$V_{t+1} = V_t + \sum(o_t \times w) \tag{2}$$

$$o_{t+1} = \begin{cases} 1, & \text{if } V_{t+1} \geq \theta_t^+ \\ -1, & \text{if } V_{t+1} < \theta_t^- \\ 0, & \text{else} \end{cases} \tag{3}$$

$$\theta_{t+1}^+ / \theta_{t+1}^- = \begin{cases} \theta_t^+ / \theta_t^- + \delta, & \text{if } o_{t+1} = 1 \\ \theta_t^+ / \theta_t^- - \delta, & \text{if } o_{t+1} = -1 \\ \theta_t^+ / \theta_t^-, & \text{else} \end{cases} \tag{4}$$

$$\delta = \frac{\frac{1}{2^k} \times (\sigma' + \epsilon)}{\gamma} \tag{5}$$

$$\theta_0^+ = \frac{(\frac{1}{2^{k+1}} - \beta)(\sigma' + \epsilon)}{\gamma} + \mu' \tag{6}$$

$$\theta_0^- = \theta_0^+ - \delta \tag{7}$$

Besides, the regulating term of two thresholds i.e. δ , and initial value θ_0^+ / θ_0^- in Equation 5–7 are derived from Equation (1) and the BN parameters, just like Zou et al. (2023). It should be noted that this mechanism extends from the previous work (Zou et al., 2023), but features with following key characteristics:

- **More simplified computation and less memory consumption:** The original high-precision shadow membrane potential of each spiking neuron in Zou et al. (2023) is removed in this work. This improvement could significantly reduce the computational complexity, memory burden of SNNs and speed up the calculation process.
- **Higher biological plausibility:** The positive and negative threshold change synchronously, if a positive (+1) or negative spike (-1) is generated at time step t , both of the two thresholds will change accordingly to prevent firing again at this level. This working mechanism may be more biologically

TABLE 2 Comparison for the proposed spiking LeNet on MNIST with other works.

References	Network structure	ANN (full precision)	SNN (all integer for ours and high precision for others)						Accuracy loss
			T = 2	T = 4	T = 8	T = 16	T = 32	T ≥ 64	
Zou et al. (2020)	LeNet	99.29%	-	99.29%	-	-	-	-	0.00%
Bu et al. (2022)	LeNet	99.07%	-	-	-	-	-	99.09%	-0.02%
Tong et al. (2022)	LeNet	99.19%	-	-	-	99.17%	-	-	0.02%
Howard et al. (2017)	LeNet	99.21%	-	-	99.19%	-	-	-	0.02%
Diehl et al. (2015)	LeNet	99.14%	9.80%	9.80%	9.84%	93.73%	98.93%	98.99%	0.15%
Bodo et al. (2017)	LeNet	99.44%	12.98%	88.03%	98.17%	99.25%	99.44%	99.44%	0.00%
Zou et al. (2023)	LeNet	99.29%	99.18%	99.31%	99.29%	99.29%	99.29%	99.29%	0.00%
k = 0 (This work)	LeNet	99.45%	99.09%	99.45%	99.42%	99.42%	99.42%	99.42%	0.03%

The bold number represents the best in the column.

plausible because of its similarity to the refractory period phenomenon (Kasabov, 2019).

Compared with the negative spiking mechanism proposed in Hu et al. (2023), there are at least following differences: (1) The signed IF neuron in Hu et al. (2023) needs an auxiliary variable to record the number of spikes each neuron has fired at the current time step, while our spiking neurons doesn't require extra variable. (2) The threshold (positive and negative) in Hu et al. (2023) is static along time step, while ours is dynamic and also more biologically plausible.

3 Neural network quantization

In this section, we first introduce the input spike encoding algorithm, i.e. the conversion process of the static image pixels to input sequential spikes. Then, an all integer-based parameter quantization method will be presented to build a hardware-friendly SNN model.

3.1 Input spike encoding

In general, lots of static data in nature such as image and text, would be collected and used for ANN training, but must be converted to spiking signals before it can be fed into SNNs during inference. In this article, we adopt a constant scatter-and-gather (SG) encoding algorithm as in Zou et al. (2021) to minimize the information loss of the input layer between ANN and SNN models. As shown in Figure 2, any intensity value will be processed through a normal convolutional layer and then discretized into a spike sequence with only 4 time steps. This skill could further accelerate SNN simulation (smaller time window) and decrease the total computation cost. It should be noted that the weight parameters of this encoding layer are trainable but not quantized, and these spiking neurons only act as the original LIF behavior (Andrew, 2003) without negative spiking mechanism.

3.2 Parameter quantization

Ultimately, the ultra-high energy efficiency strength of SNNs comes from two important features: (1) **low parameter precision** and (2) **sparsity calculation**. This means that only when SNNs are deployed on some dedicated neuromorphic hardware, can they reach their real low-power potential. However, many SNN works (Taherkhani et al., 2020; Neftci et al., 2019; Dampfhofer et al., 2023) mainly concentrate on the model accuracy improvements but omit their hardware friendliness. In this article, we present an all integer-based parameter quantization approach for both synaptic weights and firing thresholds, which would greatly facilitate the hardware implementation for the proposed SNN models.

3.2.1 Weight quantization

We adopt a lightweight ternary quantization scheme as in Liu et al. (2023) for the synaptic weights in convolutional and fully

TABLE 3 Classification accuracy for VGG-Net of different configurations on CIFAR-10.

VGG-Net Architecture		Weight quantization	Threshold quantization	Final accuracy	Final time step
ANN		×	-	93.32%	-
		✓		93.29%	
SNN	k = 0	×	×	93.26%	13
		✓	×	93.19%	11
		×	✓	93.12%	12
		✓	✓	93.11%	10
	k = 1	×	×	93.22%	18
		✓	×	93.19%	16
		×	✓	93.20%	17
		✓	✓	93.15%	15
	k = 2	×	×	93.27%	32
		✓	×	93.21%	30
		×	✓	93.29%	32
		✓	✓	93.21%	29

The **bold** number represents the best in the column.

connected layers. As in Equation 8, the floating-point weights will be quantized to only 0,1 or -1 depending on a specific threshold $= 0.7 \times |W^f|$. Besides, we adopt a straight-through estimator (STE) function (Bengio et al., 2013) as in Equation 9 to pass the gradients backwards through the networks. Because both inputs and outputs of spiking neurons are in the form of spike sequence (0,1 or -1), the multiplication-addition (MAC) calculations in these layers could be implemented by only bit-operation such as *XNOR-popcount* in Courbariaux et al. (2016). As a typical quantization aware training (QAT) approach, the ternary weights in ANNs are trainable and will be unchanged when converted into the SNN version. This quantization process for weight would significantly increase the simplicity of massive spike integration calculation.

$$W^T = \begin{cases} 1, & \text{if } W^F > 0.7 \times |W^F| \\ 0, & \text{if } |W^F| \leq 0.7 \times |W^F| \\ -1, & \text{if } W^F < -0.7 \times |W^F| \end{cases} \quad (8)$$

$$\frac{\partial L}{\partial W^F} = \frac{\partial L}{\partial W^T} \times \frac{\partial W^T}{\partial W^F} \approx \begin{cases} \frac{\partial L}{\partial W^T} & \text{if } |W^F| \leq 1 \\ 0 & \text{if } |W^F| > 1 \end{cases} \quad (9)$$

3.2.2 Threshold quantization

As shown in Equations 5–7, each SNN neuron has two firing thresholds θ_0^+ , θ_0^- and one regulating term δ . These parameters consist of the mean μ , variance σ , scaling γ and shift β terms in BN (Ioffe and Szegedy, 2015) layers, which are originally all floating-point numbers. Here, we firstly scale the weights W^T and firing thresholds θ_0^+ , θ_0^- by 10,000 \times and then use a *Round* function as in Equation 10 to quantize them into all integers for hardware-equivalent mapping. It should be mentioned that the quantization test for firing threshold is very rare in many previous

SNN works. Based on our multiple experimental tries, we found these parameters are not sensitive to the choice of quantization function including *Floor*, *Round* and *Ceil*, etc. We will discuss the effects of quantization for these parameters in the next section.

$$Quant(\theta_0^+ / \theta_0^- / \delta) = Round(\theta_0^+ / \theta_0^- / \delta \times 10000) \quad (10)$$

4 Result and discussion

In this section, we firstly test the proposed ANN2SNN conversion method on both MNIST and CIFAR-10 dataset and present the experimental results including the inference accuracy and speed, together compared with several state-of-the-art works of similar network sizes. Then, we carry out an estimation of spiking sparsity and synaptic operations (SOPs) of the proposed spiking models, to verify their great energy efficiency. All models are trained using standard ADAM (Krizhevsky and Hinton, 2009) rule with an initial learning rate set to 0.01 and we don't use any data augmentation other than a standard random image flipping and cropping for CIFAR-10. It should be noted our ANN2SNN conversion method can be compatible with many other optimization and regularization skills (Cheng et al., 2017) and some advanced architectures like ResNet (Li et al., 2022) for further better performances. More details are available from our online Python implementation code (https://github.com/edwardzcl/All_INT_SNN).

4.1 Experiments on MNIST

MNIST handwritten digit dataset (Lecun and Bottou, 1998) has been widely applied in image classification field, which was

TABLE 4 Comparison for the proposed spiking VGG-Net on CIFAR-10 with other works.

References	Network structure	ANN (full precision)	SNN (all integer for ours and high precision for others)						Accuracy loss
			T = 4	T = 8	T = 16	T = 32	T = 64	T ≥ 128	
Li et al. (2021)	VGG-16	93.51%	-	-	-	91.82%	92.57%	93.21%	0.30%
Meng et al. (2022)	ResNet-20	93.18%	-	-	89.17%	91.91%	92.68%	92.92%	0.26%
Zou et al. (2023)	VGG-11	92.66%	91.29%	92.88%	92.65%	92.64%	92.64%	92.64%	0.02%
Deng and Gu (2021)	VGG-16	95.60%	-	91.41%	93.64%	94.81%	-	-	0.79%
Gao et al. (2023)	VGG-16	92.09%	-	-	92.29%	92.29%	92.22%	92.24%	-0.15%
Bu et al. (2022)	VGG-16	94.02%	-	80.28%	90.35%	93.10%	93.32%	93.68%	0.34%
Yousefzadeh et al. (2019)	ResNet-20	92.74%	-	66.24%	87.22%	91.88%	92.57%	92.73%	0.01%
Tong et al. (2022)	ResNet-20	91.77%	83.75%	89.55%	91.62%	92.24%	92.35%	-	-0.58%
k = 0 (This work)	VGG-11	93.32%	92.05%	93.15%	93.11%	93.11%	93.11%	93.11%	0.21%

The **bold** number represents the best in the column.

TABLE 5 Comparison for the proposed spiking VGG-Net on CIFAR-100 with other works.

References	Network structure	ANN (full precision)	SNN (all integer for ours and high precision for others)						Accuracy loss
			T = 4	T = 8	T = 16	T = 32	T = 64	T ≥ 128	
Li et al. (2021)	VGG-16	70.21%	-	-	-	64.53%	67.14%	68.99%	1.22%
Li et al. (2021)	MobileNet	73.23%	-	-	-	40.06%	62.81%	69.41%	3.82%
Li et al. (2021)	ResNet-20	68.40%	-	-	-	65.14%	67.63%	68.28%	0.12%
Meng et al. (2022)	ResNet-20	70.15%	-	-	59.61%	66.24%	69.14%	69.99%	0.16%
Gao et al. (2023)	ResNet-20	77.16%	-	-	-	51.27%	70.12%	75.81%	1.35%
Gao et al. (2023)	VGG-16	77.89%	-	-	-	7.64%	21.84%	55.04%	22.85%
Gao et al. (2023)	MobileNet	73.23%	-	-	-	1.28%	4.88%	39.39%	33.84%
Bu et al. (2022)	VGG-16	72.47%	-	58.58%	66.32%	71.71%	72.68%	72.85%	-0.38%
Yousefzadeh et al. (2019)	ResNet-20	70.43%	-	23.09%	52.34%	67.18%	69.96%	70.51%	-0.08%
[42]	ResNet-20	69.94%	34.14%	55.37%	67.33%	69.82%	70.49%	-	-0.55%
k = 0 (This work)	VGG-11	67.40%	54.27%	60.85%	64.42%	65.21%	65.21%	65.21%	2.19%

The **bold** number represents the best in the column.

TABLE 6 Comparison for the proposed spiking MobileNet on ImageNet with other works.

References	Network structure	ANN (full precision)	SNN (all integer for ours and high precision for others)					Accuracy loss	
			T = 16	T = 32	T = 64	T = 128	T = 256		T ≥ 512
Li et al. (2021)	MobileNet	73.40%	-	37.43%	56.26%	65.40%	69.02%	72.38%	1.02%
Li et al. (2021)	ResNet-34	75.66%	-	50.21%	63.66%	68.89%	72.12%	75.44%	0.22%
Li et al. (2021)	VGG-16	75.36%	-	24.88%	56.77%	70.49%	73.66%	75.15%	0.21%
Bodo et al. (2017)	VGG-16	63.89%	-	-	-	-	-	49.61%	14.28%
Sengupta et al. (2019)	ResNet-34	70.69%	-	-	-	-	-	65.47%	5.22%
Gao et al. (2023)	ResNet-34	70.95%	-	33.01%	59.52%	67.54%	70.06%	70.98%	-0.03%
Gao et al. (2023)	VGG-16	72.40%	-	54.92%	66.51%	69.94%	71.35%	72.09%	0.31%
k = 0 This work	MobileNet	70.60%	47.32%	56.15%	61.89%	64.48%	65.21%	65.21%	5.39%

The bold number represents the best in the column.

collected from the postal codes including a training set of 60,000 examples, and a test set of 10,000 examples. Each example is an individual 28×28 -pixel grayscale image, labeled 0~9. For this task, we adopt a classical 6-layer LeNet architecture (Lecun and Bottou, 1998), i.e., $32C5-2P2-64C5-2P2-1000FC-10FC$ where C, P and FC denoted convolution, pooling and fully connected layer respectively and 32C5 represents a convolution layer with 32 output channels and 5×5 kernel size. It should be noted that the input layer and output layer are not quantized and used for the spike encoding and loss calculation respectively. We firstly conduct an ablation study on the quantization level k based on a choice of $\{0,1,2\}$ as well as the weight, threshold quantization process. These parameter configurations determine the quantization precision of an SNN model and how many spikes each neuron will fire, which is related to total power consumption and latency in network. As shown in Table 1, all predictions stabilize quickly, and there is only little impact on the final stable accuracies for different configurations, except for the final stable time steps. The proposed quantization has almost no effect on model accuracy, but higher quantization levels will bring longer simulation time. To our surprise, the SNN with $k=2$ can achieve an accuracy that is comparable to its full-precision ANN counterpart. For a comparison, we summarize our results (for $k=0$) and other state-of-the-art works in Table 2. It shows that the spiking LeNet with both weight and threshold quantization is nearly lossless with its ANN counterpart (smallest accuracy loss), and can even achieve great accuracy and speed advantages among many other works with full-precision parameters.

4.2 Experiments on CIFAR-10

CIFAR-10 (Krizhevsky and Hinton, 2009) is regarded as a more challenging real image classification dataset, which consists of total 60000 color images with 32×32 pixels. This dataset is divided into 50000 training images and 10000 test images with 10 classes. For this task, a VGG-Net (Simonyan and Zisserman, 2014) variant with 11 layers ($96C3-256C3-2P2-384C3-2P2-384C3-256C3-2P2-1024C3-1024FC-10FC$) is designed. No extra data augmentation technique is used other than standard random image flipping and cropping for training. Test evaluation is based solely on central 24×24 crop from test set.

Similarly, we give the ablation study results of different quantization configurations in Table 3 and compare the performance results with other works in Table 4. It also shows that higher quantization levels could bring slightly better accuracy while longer simulation time steps. Besides, the reported inference accuracy and speed of spiking VGG-Net in Table 4 indicates that our proposed conversion and quantization method can still maintain excellent performance (accuracy vs. speed) with the smallest accuracy loss for deeper VGG-Net with more than 10 layers and complex BN operations (Ioffe and Szegedy, 2015). Compared with many other high-precision SNN works, our proposed spiking models are all integer-based

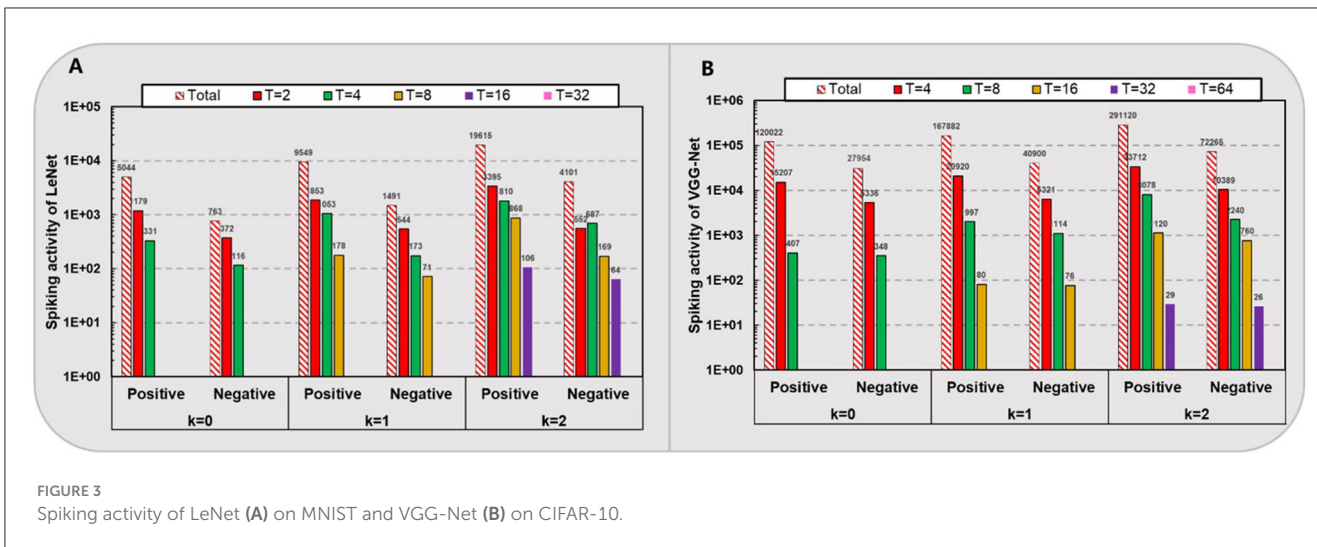


FIGURE 3 Spiking activity of LeNet (A) on MNIST and VGG-Net (B) on CIFAR-10.

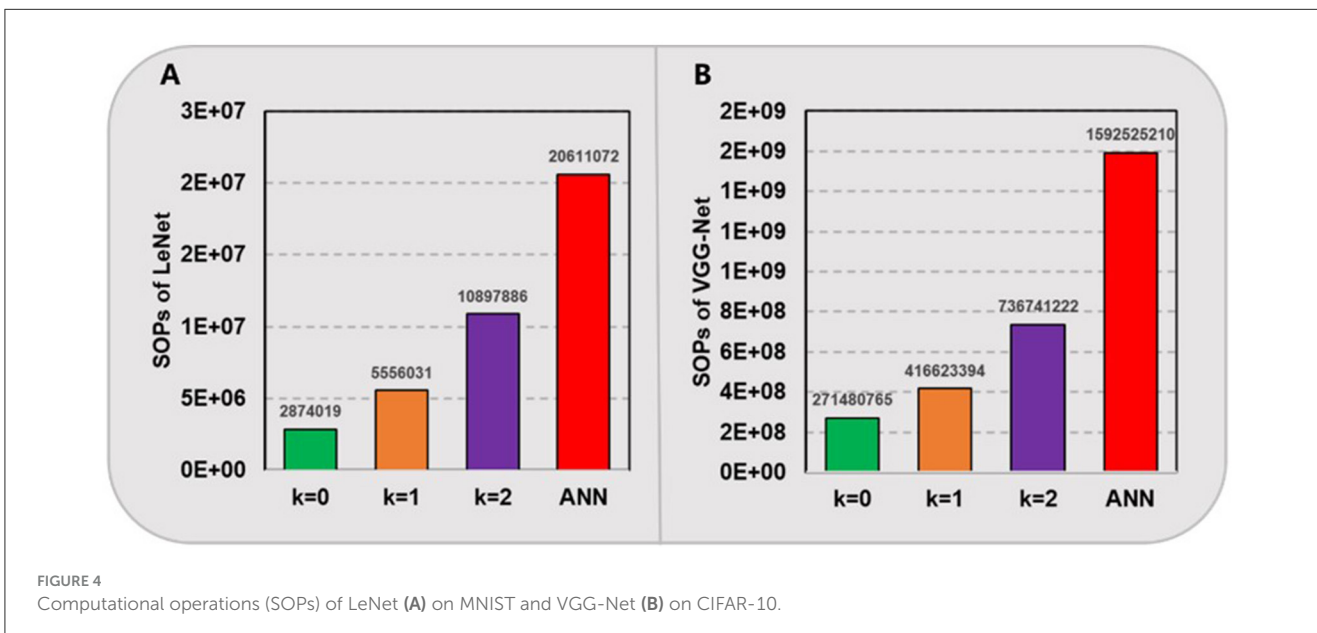


FIGURE 4 Computational operations (SOPs) of LeNet (A) on MNIST and VGG-Net (B) on CIFAR-10.

and show strong potential for direct implementation on some dedicated hardware.

4.3 Experiments on CIFAR-100 and ImageNet

CIFAR-100 (Krizhevsky and Hinton, 2009) is just like the CIFAR-10 but more challenging. It has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. ImageNet (Russakovsky et al., 2015) is a much larger dataset, which consists of more than one million image samples and falls into 1000 categories. To verify the effect of our conversion algorithm on these two datasets, we adopt the VGG-11

(the same as the network for CIFAR-10) and a 29-layer MobileNet-V1 (43) for experiment running, respectively. Similarly, we do not use any other optimization techniques for training and the test evaluation is based solely on central crop from test set. It should be noted we train MobileNet-V1 on ImageNet dataset for only 60 epochs, because it needs quite long simulation time and vast parallel computing resources. The experimental results on these two large-scale datasets are summarized in Tables 5, 6, and some comparison data of (Gao et al., 2023; Bu et al., 2022) are collected from self-implementation results (Li et al., 2021). It can be seen that the accuracies of both the proposed spiking VGG-Net and MobileNet could achieve much faster convergence along early time steps, when compared with other works respectively. This phenomenon may be attributed to our good solution of synchronization error which is discussed in Section 2.1. The final accuracy is slightly

damaged because our ANN counterparts are trained using some basic optimization techniques and fewer epochs.

4.4 Energy efficiency

As shown in Figure 3, we count the average amount of positive and negative spikes for one sample simulation of the spiking LeNet (total 10,728 neurons) on MNIST and VGG-Net (total 280,832 neurons) on CIFAR-10 except for the first input layer and last classification layer. It can be seen that for networks with higher quantization levels, higher spike activities occur while the negative/positive ratio slightly increases. For example, as quantization level k varies from 0 to 2, the spike amount on CIFAR-10 for one sample simulation increases from 147,976 to 363,385 (averagely), and the negative/positive ratio of spikes increases from 0.23 to 0.25 (nearly). Overall, there are only about 0.5, 0.74, and 1.3 spikes per neuron with respective $k \in \{0, 1, 2\}$. In contrast, the negative/positive ratio of spikes in spiking LeNet (nearly 0.15–0.2) is relatively smaller than VGG-Net (nearly 0.23 to 0.25), which means the negative spikes play a key role in deeper networks with higher quantization levels.

Furthermore, we compare the amount of needed computational operations in above spiking models and their ANN counterparts in Figure 4. For our proposed SNNs with ternary synaptic weights and integer thresholds, there is no need for any high-precision multiplication, only a low-bit SOP, i.e., addition is required when there is a pre-synaptic spike coming. In contrast, for ANNs running on traditional CPUs or GPUs, massive matrix MAC will be performed. Here, we hypothesize that a high-precision MAC is equivalent to 4 low-bit SOPs. In fact, the power and area cost of a floating-point multiplication are always much more expensive than that of several integer-based additions in most of hardware systems (Hu et al., 2023; Courbariaux et al., 2016; Howard et al., 2017). As shown in Figure 4, it can be seen that our proposed SNNs with quantization level $k \in \{0, 1, 2\}$ consume nearly 7.2, 3.7, and 1.9 times fewer computational operations for LeNet and 5.9, 3.8, and 2.2 times fewer for VGG-Net compared to their ANN counterparts, respectively. These results prove that the converted SNNs can achieve much higher energy efficiency than ANNs, while maintaining comparable accuracy.

Furthermore, because our proposed spiking models run with 0 or ± 1 weights and spikes, integer threshold and leakage variables, these integer-based operations could be replaced by the efficient bit-operation such as *XNOR-popcount*, which is introduced in the binary neural networks (BNNs) (Courbariaux et al., 2016) and ternary neural networks (TNNs) (Liu et al., 2023). Even though the computing cost and latency of SNNs may be greater than these two kinds of special ANN-domain models (Tavanaei et al., 2019), the high-accuracy and spatio-temporal processing abilities on some more complex applications still make them the first choice. Of course, a more fair or in-depth comparison between BNNs/TNNs and SNNs may be a perennial topic and will be considered in the future works.

5 Conclusion

In this work, we introduce a novel dynamic threshold adaptation technique into traditional ANN2SNN conversion process to eliminate common spike approximation error, and further present an all integer-based quantization method to obtain a lightweight and hardware-friendly SNN model. Experimental results show that the proposed spiking LeNet and VGG-Net can obtain more than 99.45% and 93.15% accuracy on MNIST and CIFAR-10 dataset with only 4 and 8 time steps, respectively. Besides, the captured spiking activity and computational operations in SNNs indicate that our proposed spiking models can achieve much higher energy efficiency with comparable accuracy than their ANN counterparts. Finally, our future works will concentrate on the conversion and quantization methods for some special architecture, such as ResNet, RNN and transformer-based models. More importantly, try to map these models onto some dedicated neuromorphic hardware is more rewarding, this will bring a real running performance improvement for some edge computing applications.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

CZ: Methodology, Writing – original draft. XC: Funding acquisition, Writing – original draft, Writing – review & editing. SF: Formal analysis, Validation, Writing – review & editing. GC: Formal analysis, Validation, Writing – review & editing. YZ: Data curation, Software, Writing – review & editing. ZD: Data curation, Software, Writing – review & editing. YW: Funding acquisition, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Postdoctoral Research Station of CQBDRI.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Andrew, A. M. (2003). Spiking neuron models: single neurons, populations, plasticity. *Kybernetes* 32:7–8. doi: 10.1108/k.2003.06732gae.003
- Bengio, Y., Nicholas, L., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv [Preprint]* arXiv:1308.3432. doi: 10.48550/arXiv.1308.3432
- Bodo, R., Iulia-Alexandra, L., Hu, Y., Michael, P., and Liu, S. C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682
- Bouvier, M., Valentian, A., Mesquida, T., Rummens, F., Reyboz, M., Vianello, E., et al. (2019). Spiking neural networks hardware implementations and challenges: A survey. *ACM J. Emerg. Technol. Comp. Syst.* 15, 1–35. doi: 10.1145/3304103
- Bu, T., Ding, J., Yu, Z., and Huang, T. (2022). Optimized potential initialization for low-latency spiking neural networks. *arXiv [Preprint]*. arXiv:2202.01440. doi: 10.48550/arXiv.2202.01440
- Chen, L., Lei, M., and Steve, F. (2022). Quantization framework for fast spiking neural networks. *Front. Neurosci.* 16:918793. doi: 10.3389/fnins.2022.918793
- Cheng, Y., Wang, D., Zhou, P., and Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. *arXiv [Preprint]*. arXiv:1710.09282. doi: 10.48550/arXiv.1710.09282
- Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1. *arXiv [Preprint]*. arXiv:1602.02830. doi: 10.48550/arXiv.1609.03368
- Dampfhofer, M., Mesquida, T., Valentian, A., and Anghel, L. (2023). Backpropagation-based learning techniques for deep spiking neural networks: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* 35, 11906–11921. doi: 10.1109/TNNLS.2023.3263008
- Deng, S., and Gu, S. (2021). "Optimal conversion of conventional artificial neural networks to spiking neural networks," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
- Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney: IJCNN).
- Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. National Acad. Sci. USA.* 113, 11441–11446. doi: 10.1073/pnas.1604850113
- Gao, H., He, J., Wang, H., Wang, T., Zhong, Z., Yu, J., et al. (2023). High-accuracy deep ANN-to-SNN conversion using quantization-aware training framework and calcium-gated bipolar leaky integrate and fire neuron. *Front. Neurosci.* 17, 254–268. doi: 10.3389/fnins.2023.1141701
- Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)* 15: 315–323.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: IEEE), 770–778.
- Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv [preprint]* arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861
- Hu, Y., Zheng, Q., Jiang, X., and Pan, G. (2023). Fast-SNN: fast spiking neural network by converting quantized ANN. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 14546–14562. doi: 10.1109/TPAMI.2023.3275769
- Ioffe, S., and Szegedy, C. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (Lille: PMLR), 448–456.
- Kasabov, N. K. (2019). Evolving spiking neural networks. In: time-space, spiking neural networks and brain-inspired artificial intelligence. *Springer Series on Bio- and Neurosyst.* 7: 169–199. doi: 10.1007/978-3-662-57715-8_5
- Krizhevsky, A., and Hinton, G. (2009). "Learning multiple layers of features from tiny images," in *Handbook of Systemic Autoimmune Diseases*. Available at: <https://core.ac.uk/display/21817232>.
- Lecun, Y., and Bottou, L. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE.* 86, 2278–2324. doi: 10.1109/5.726791
- Lee, C., Srinivasan, G., Panda, P., and Roy, K. (2019). Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity. *IEEE Trans. Cogn. Dev. Syst.* 11, 384–394. doi: 10.1109/TCDS.2018.2833071
- Li, Y., Deng, S., Dong, X., Gong, R., and Gu, S. (2021). "A free lunch from ANN: towards efficient, accurate spiking neural networks calibration," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 6316–6325.
- Li, Y., Deng, S., Dong, X., and Gu, S. (2022). Converting artificial neural networks to spiking neural networks via parameter calibration. *arXiv [Preprint]*. arXiv:2205.10121. doi: 10.48550/arXiv.2205.10121
- Liu, B., Li, F., Wang, X., Zhang, B., and Yan, J. (2023). "Ternary weight networks," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Rhodes Island: IEEE), 1–5.
- Massa, R., Marchisio, A., Martina, M., and Shafique, M. (2020). "An efficient spiking neural network for recognizing gestures with a DVS camera on the Loihi neuromorphic processor," in *2020 International Joint Conference on Neural Networks (IJCNN)* (Glasgow: IJCNN), 1–9.
- Meng, Q., Yan, S., Xiao, M., Wang, Y., Lin, Z., and Luo, Z. (2022). Training much deeper spiking neural networks with a small number of time-steps. *Neural Netw.* 153, 254–268. doi: 10.1016/j.neunet.2022.06.001
- 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 Process. Mag.* 36, 51–63. doi: 10.1109/MSP.2019.2931595
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y
- Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., et al. (2017). A survey of neuromorphic computing and neural networks in hardware. *arXiv [Preprint]* arXiv:1705.06963. doi: 10.48550/arXiv.1705.06963
- Sengupta, A., Ye, Y., Wang, R., Liu, C., and Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* 13:95. doi: 10.3389/fnins.2019.00095
- Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556.
- Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., and McGinnity, T. M. (2020). A review of learning in biologically plausible spiking neural networks. *Neural Netw.* 122, 253–272. doi: 10.1016/j.neunet.2019.09.036
- Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. (2019). Deep learning in spiking neural networks. *Neural Netw.* 111, 47–63. doi: 10.1016/j.neunet.2018.12.002
- Tong, B., Wei, F., Jian, D., Peng, D., Zhao, Y., and Tie, H. (2022). "Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.
- Xu, Y., Tang, H., Xing, J., and Li, H. (2017). "Spike trains encoding and threshold rescaling method for deep spiking neural networks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (Honolulu, HI: IEEE), 1–6. doi: 10.1109/SSCI.2017.8285427
- Yousefzadeh, A., Hosseini, S., Holanda, P., Leroux, S., Werner, T., Serrano-Gotarredona, T., et al. (2019). "Conversion of synchronous artificial neural network

to asynchronous spiking neural network using sigma-delta quantization,” in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (Hsinchu: IEEE), 81–85.

Zou, C., Cui, X., Chen, G., Jiang, Y., and Wang, Y. (2023). Towards a lossless conversion for spiking neural networks with negative-spike dynamics. *Adv. Intellig. Syst.* 5, 1–12. doi: 10.1002/aisy.202300383

Zou, C., Cui, X., Ge, J., Ma, H., and Wang, X. (2020). “A novel conversion method for spiking neural network using median quantization,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* (Seville: IEEE), 1–5.

Zou, C., Cui, X., Kuang, Y., Liu, K., Wang, Y., Wang, X., et al. (2021). A scatter-and-gather spiking convolutional neural network on a reconfigurable neuromorphic hardware. *Front. Neurosci.* 15:694170. doi: 10.3389/fnins.2021.694170