

# Constructing Accurate and Efficient Deep Spiking Neural Networks with Double-threshold and Augmented Schemes

Qiang Yu, Chenxiang Ma, Shiming Song, Gaoyan Zhang,  
Jianwu Dang, Kay Chen Tan, *Fellow, IEEE*

**Abstract**—Spiking neural networks (SNNs) are considered as a potential candidate to overcome current challenges such as the high-power consumption encountered by artificial neural networks (ANNs), however there is still a gap between them with respect to the recognition accuracy on practical tasks. A conversion strategy was thus introduced recently to bridge this gap by mapping a trained ANN to an SNN. However, it is still unclear that to what extent this obtained SNN can benefit both the accuracy advantage from ANN and high efficiency from the spike-based paradigm of computation. In this paper, we propose two new conversion methods, namely TerMapping and AugMapping. The TerMapping is a straightforward extension of a typical threshold-balancing method with a double-threshold scheme, while the AugMapping additionally incorporates a new scheme of augmented spike that employs a spike coefficient to carry the number of typical all-or-nothing spikes occurring at a time step. We examine the performance of our methods based on MNIST, Fashion-MNIST and CIFAR10 datasets. The results show that the proposed double-threshold scheme can effectively improve accuracies of the converted SNNs. More importantly, the proposed AugMapping is more advantageous for constructing accurate, fast and efficient deep SNNs as compared to other state-of-the-art approaches. Our study therefore provides new approaches for further integration of advanced techniques in ANNs to improve the performance of SNNs, which could be of great merit to applied developments with spike-based neuromorphic computing.

**Index Terms**—Deep spiking neural networks, double thresholds, augmented spikes, ANN-to-SNN conversion, pattern recognition, neuromorphic computing.

## I. INTRODUCTION

AS a subset of artificial neural networks (ANNs), deep neural networks (DNNs) [1] have shown significant improvements in a wide range of tasks such as image classification [2], speech recognition [3], natural language processing [4] and robotics [5], etc. However, with the complexity of neural networks increasing progressively, running such deep networks often requires large amounts of computational resources such as memory and power, thus limiting their applied developments in battery-constrained devices such as cell phones and embedded electronics [6]. Some studies focus

on reducing the network connections and using low-precision parameters [7]–[10], but the computational consumption is still large. Hence, the challenge still remains for low-power paradigms to enlarge the applicability of DNNs.

Different from numerical values used by traditional ANNs, spiking neural networks (SNNs) emulates the brain in a way to utilize discrete spikes for information representation and transmission, and thus are more brain-like and computationally powerful [11], [12]. Moreover, owing to the discrete feature of spikes over an additional time dimension, SNNs are capable of asynchronous and sparse computation under an event-based manner where a computational budget is paid only at the appearance of a spike event [13]–[16]. This motivates the development of neuromorphic computing platforms that have successfully shown a remarkable performance of orders of magnitude more efficient in terms of power consumption than conventional computing platforms [17]–[19]. The attractive potential of spike-based computation is drawing increasing efforts to the development of SNNs [20]–[24], which yet is still very much in its infancy.

Although SNNs are promising for low-power and fast inference by their nature, training such deep networks is difficult due to their inherent characteristics of discontinuity, non-linearity and complexity on dynamics, let alone to achieve a competitive recognition accuracy to ANNs [25]–[27]. There are two mainstream approaches developed to overcome the challenge of training deep SNNs: direct and indirect training. The indirect training approaches are also often referred as conversion or mapping methods.

An early attempt of the direct training methods is based on spike-timing-dependent-plasticity (STDP) where synaptic modification is controlled by the local correlation of pre- and postsynaptic spike timings [28]. However, SNNs trained with STDP are normally limited to shallow structures, and cannot be scaled up to large networks with high performances due to the lack of a global instructor [29]–[32]. Backpropagation (BP), a prevalent learning algorithm in ANNs [33], has been successfully applied to train deep SNNs by addressing the inherent non-differentiable obstruction during the backward propagation of an error instructor [34]–[40]. A surrogate is usually designed to approximate the gradients in these BP-based approaches. This approximation is effective with relatively shallow structures, but normally gets worse for more challenging tasks and deeper networks. Additionally, direct training methods are often time-consuming during the

Q. Yu, C. Ma, S. Song, G. Zhang and J. Dang are with Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China.

J. Dang is also with Japan Advanced Institute of Science and Technology, Japan.

K.C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Corresponding author: Q. Yu (e-mail: yuqiang@tju.edu.cn).

adaptation of neural parameters. Moreover, as compared to ANNs, there is still a big gap with respect to the recognition accuracy for these direct training methods [25], [26].

In order to narrow the gap between SNNs and ANNs, a conversion scheme emerges by mapping the weights of a trained ANN to an SNN of the same architecture (see Fig. 1A). This conversion scheme leverages on advanced techniques in ANNs to achieve a comparable accuracy with SNNs. One of the most early works successfully introduces the conversion scheme with complicated spiking neurons, but suffers from a significant accuracy loss [41]. Later efforts discover that the firing rates of spiking neurons can approximate the activations of their counterparts in ANNs with sufficient time steps [42]. This finding has become the fundamental principle underlying the conversion scheme. A shallow convolutional network trained with certain constraints can be successfully deployed to an SNN, resulting in a good accuracy on traditional object recognition benchmarks [42]. Then, a data-based normalization (DataNorm) [43] is developed to achieve a nearly lossless performance by threshold-balancing for a proper information transmission with firing rates, but the techniques used for the conversion are rather limited. An extended variant [44] is thus developed later for further improvements by incorporating more techniques from ANNs, including biases, max pooling, softmax and batch normalization. Later, SpikeNorm [45] is developed to scale neural parameters according to the activities of SNNs rather than ANNs, leading to a nearly lossless conversion even in very deep networks. Recently, a channel-wise normalization is introduced to further minimize the conversion loss with an elaborate adjustment, and achieves comparable results to ANNs on the object detection task [46].

Despite remarkable achievements of nearly lossless conversion even in very deep networks, threshold-balancing techniques [43]–[46] require a large set of training data to extract auxiliary values used in the conversion. Therefore, these techniques are data-driven and susceptible to the selection of samples. Furthermore, the current conversion approaches encounter other **common challenges as follows:**

- A delicate balance between weights and thresholds to avoid information loss caused by over- or under-activation [43] makes the current conversion approaches quite complicated.
- The converted SNNs with current approaches are inefficient in both time and energy as they require a large number of spikes and time steps especially for significantly deep networks.
- Most of approaches cannot convert negative activations in ANNs to their spiking counterparts, and thus advanced variations [47] like LeakyReLU are unable to be utilized. This limits the potential of SNNs to take full advantages of advanced techniques in ANNs.

In order to improve the inference speed and energy efficiency, preliminary efforts are made to extend spikes with different forms, such as multi-strength spikes [48], weighted spikes [49] and burst spikes [50]. Despite their efficacy in reduction of classification latency and number of events, these works are still limited to constrained ANNs with positive

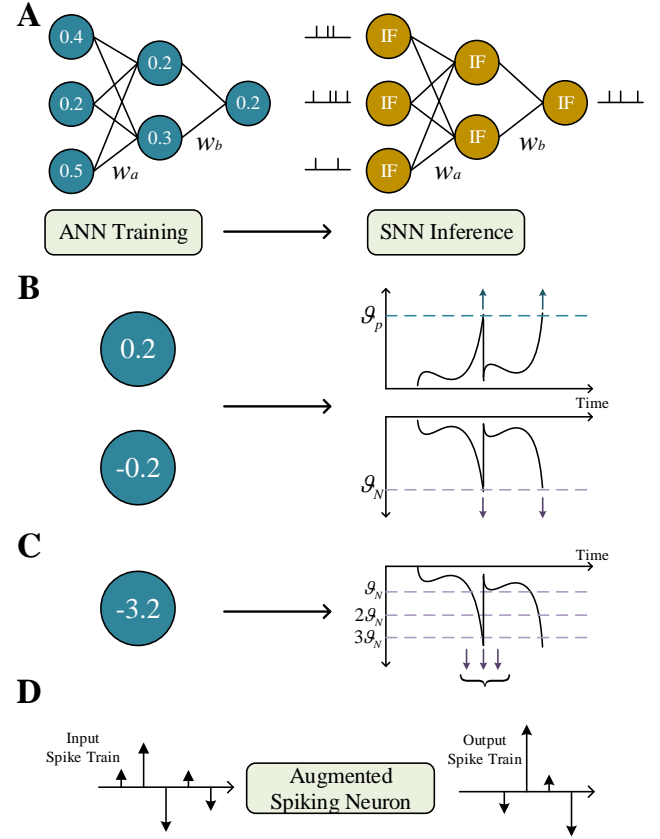


Fig. 1. Illustrations of ANN-to-SNN conversion. **A**, a standard conversion scheme. **B**, mapping both positive and negative activations in ANNs with a double-threshold scheme. **C**, augmented spikes to address the over-activation issue. **D**, augmented spiking neuron that receives and elicits augmented spikes.

activations only. Moreover, their conversions are based on either complicated schemes like DataNorm or an exhaustive strategy with manual exploration. A clear and simple rule to determine proper parameters for spiking neurons is still under exploring.

In this paper, we first propose a conversion approach called TerMapping by extending DataNorm [43] with a new double-threshold scheme. Then, we introduce an advanced neuron model that is capable of receiving and eliciting augmented spikes. Based on the model, another new conversion method called AugMapping is developed. We evaluate the performance of our methods with extensive experiments. Our major contributions can be enumerated as follows:

- A double-threshold scheme is introduced such that SNNs are enhanced to represent both positive and negative activations, relieving constraints on the selection of ANNs.
- A new scheme of augmented spike is introduced where a spike coefficient is used to represent additional information including both the polarity and the number of binary spikes occurring at one time step. This enables the correspondingly developed augmented neurons to completely overcome the pathological phenomenon of over-activation [43] on one hand, and to improve both the accuracy and efficiency with less number of spike events on the other hand.

- New conversion methods are developed with a clear and simple rule to assign suitable parameters for SNNs. More importantly, a detailed theoretical formulation about the approximation is provided, supporting the efficacy of our methods and paving the way for other related future works.
- We evaluate the effectiveness of our methods with various network structures based on different datasets including MNIST, FashionMNIST and CIFAR10. Experimental results show that our methods can achieve nearly lossless ANN-to-SNN conversion, and importantly are more fast, accurate and efficient than the current state-of-the-arts. Our work thus contributes to improve the performance of spike-based inference, which would be of great merit to neuromorphic computing.

The remainder of this paper is organized as follows. Section II presents the details of our proposed methods, followed by the experimental results and discussions in Section III and Section IV, respectively. Finally, a conclusion is provided in Section V.

## II. METHODS

### A. Double-Threshold Spiking Neural Networks

Direct training of SNNs to achieve competitive accuracies with ANNs is still challenging. Hence, a straightforward approach emerges by converting the weights of a pre-trained ANN to their counterparts in an SNN of the same structure. However, the setup of a single threshold in standard spiking neurons makes them only capable of representing information of a sole polarization. This makes it difficult to map negative activations from ANNs to SNNs, and thus constraints are often applied to ANNs to keep only positive activations. Such a constraint can decrease the performance of the pre-trained ANN, and so as the converted SNN.

In order to address this limitation, we introduce a double-threshold firing scheme (see Fig. 1B) with the integrate-and-fire (IF) model [51], one of the most widely studied neuron models in various conversion approaches [42]–[45]. Each spiking neuron maintains an internal state called membrane potential and continuously integrates postsynaptic potentials generated by afferent spikes into its membrane potential. The integration dynamics of the  $i$ -th neuron in the  $l$ -th layer at time  $t$  is described as:

$$V_i^l(t) = V_i^l(t-1) + \sum_{j=1}^{M^{l-1}} w_{ij}^l o_j^{l-1}(t) + b_i^l \quad (1)$$

where  $V_i^l(t)$  represents the membrane potential, and  $M^{l-1}$  denotes the number of neurons in the preceding layer.  $o_j^{l-1}(t)$  is the afferent spike, and  $w_{ij}^l$  is the weight connection from the  $j$ -th neuron in layer  $l-1$  to the  $i$ -th one in layer  $l$ .  $b_i^l$  is the corresponding bias.

Once the membrane potential of the neuron crosses either the positive threshold  $\vartheta_P$  or the negative one  $\vartheta_N$  (see Fig. 1B), a corresponding polarized spike will be elicited and propagated to downstream neurons, as formulated by:

$$o_i^l(t) = \begin{cases} 1 & \text{if } V_i^l(t) \geq \vartheta_P \\ -1 & \text{if } V_i^l(t) \leq \vartheta_N \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

At the occurrence of a polarized spike, the neuron instantaneously triggers a reset process where its membrane potential is changed by a value of the corresponding threshold, as given by:

$$V_i^l(t) = \begin{cases} V_i^l(t) - \vartheta_P & \text{if } o_i^l(t) = 1 \\ V_i^l(t) - \vartheta_N & \text{if } o_i^l(t) = -1 \\ V_i^l(t) & \text{otherwise} \end{cases} \quad (3)$$

Based on the above neuron model, a double-threshold SNN can thus be easily constructed.

### B. TerMapping

A fundamental guideline for the conversion approaches is that the firing rates of spiking neurons need to approximate the activations of their counterparts in an ANN. A standard spike form of all-or-nothing could inevitably lead to information loss during conversion due to its limited capacity for transmission at each time step. Both over-activation and under-activation in ANNs can result in improper representation with spikes, thus decreasing recognition performance of SNNs [43]. Appropriate balance between thresholds and input firing rates in SNNs provides an effective approach to relieve the loss to a certain extent, and can even achieve nearly lossless conversion with an elaborate configuration [42]–[45]. However, most of the current conversion approaches rely on the requirements of only positive activations, limiting the selection of ANNs to be probably sub-optimal. Our double-threshold scheme provides a solution to address this issue by incorporating an additional negative threshold to spiking neurons. Our double-threshold scheme can be applied to different conversion approaches, and here we select DataNorm [43] as a representative for extension, based on which we develop a new method named TerMapping.

After an ANN has been trained, the training set is fed to it again, and the maximum absolute values of both output activations and weights in every layer are recorded to obtain the scaling factors used for balancing procedures. In the inference with the converted SNN, firing thresholds in each layer are rescaled by their corresponding scaling factors recorded from the previous step. Our method is effective to control the firing rates of most converted neurons in a normalized range between 0 and 1, thus reducing the accuracy loss caused by improper activations. Pseudo-codes for the computation of scaling factors are shown in Algorithm 1.

### C. Augmented Spikes

Due to the representation constraint with all-or-nothing spikes at each time step, a delicate balance between thresholds and firing rates is required to reduce information loss. For example, if over-activation happens to result in more than one spike in a single time step, the standard spiking neuron can

---

**Algorithm 1:** Computation of Scaling Factors in TerMapping
 

---

**Input :** The number of layers  $n$ , each layer's weights  $w_i$  and output activations  $z_i$ ,  $i = 1, 2, \dots, n$ .

**Output:** Corresponding scaling factors  $\lambda_i$ .

```

1  $pre\_factor = 1$ 
2 for  $i \leftarrow 1$  to  $n$  do
3   /* Compute the maximum absolute value of
   weights and output activations */
4    $max\_weight = w_i.abs().max()$ 
5    $max\_output = z_i.abs().max()$ 
6    $post\_factor = max(max\_weight, max\_output)$ 
7   /* Obtain the scaling factor for the
   current layer */
8    $\lambda_i = post\_factor / pre\_factor$ 
9    $pre\_factor = post\_factor$ 
10 end for
  
```

---

only elicit maximally one accordingly, thus inevitably leading to the decrease in performance. In order to address this issue, we introduce a new scheme of augmented spike where a spike coefficient is employed to carry additional information including both polarity and the number of typical spikes occurring at one time step (see Fig. 1C). Specifically, the presenting form of  $o_i^l(t)$  at a time step is extended from binary to multiple stages. Augmented spikes extend the capability of spike-based representation, and thus could be useful to reduce information loss in conversion approaches.

Endowing spiking neurons with the ability of processing and eliciting augmented spikes, a new augmented spiking neuron model is developed (see Fig. 1D). Whenever a firing condition is reached, the neuron will elicit an augmented spike, as:

$$o_i^l(t) = \begin{cases} \left\lfloor \frac{V_i^l(t)}{\vartheta_P} \right\rfloor & \text{if } V_i^l(t) \geq \vartheta_P \\ -\left\lfloor \frac{V_i^l(t)}{\vartheta_N} \right\rfloor & \text{if } V_i^l(t) \leq \vartheta_N \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where the floor division operator  $\lfloor \cdot \rfloor$  returns the integer value of the quotient.

If an augmented spike is elicited, the membrane potential of the spiking neuron is instantaneously decreased or increased by a certain amount of the corresponding threshold levels, as given by:

$$V_i^l(t) = \begin{cases} V_i^l(t) - o_i^l(t)\vartheta_P & \text{if } o_i^l(t) \geq 1 \\ V_i^l(t) + o_i^l(t)\vartheta_N & \text{if } o_i^l(t) \leq -1 \\ V_i^l(t) & \text{otherwise} \end{cases} \quad (5)$$

#### D. AugMapping

Based on the augmented spiking neuron model, we propose a new AugMapping method to realize ANN-to-SNN conversion. The augmented firing scheme enables spiking neurons to represent both positive and negative activations of an ANN, being beneficial to adopt more advanced activation functions like LeakyReLU [47], [52] for a better recognition accuracy with the converted SNN. More importantly, complicated

balancing techniques required for a proper transmission of information with standard spikes can be eliminated under our augmented scheme thanks to its advanced capacity for information representation. As a result, a more simple and clear technique can be developed for direct construction of an SNN with a pre-trained ANN, as detailed in the following.

In our AugMapping, the firing rate of an augmented spiking neuron should approximate the output activation of its counterpart in ANN. Here, we present an analytical description for this approximation, and on its basis, we can derive a simple rule to assign appropriate parameters for the converted spiking neurons.

The output activation of a neuron in ANNs with LeakyReLU or ReLU activation function can be formulated by:

$$z_i^l = \alpha \left( \sum_{j=1}^{M_{l-1}} w_{ij}^l z_j^{l-1} + b_i^l \right) \quad (6)$$

where  $z_i^l$  indicates the output activation of the  $i$ -th neuron in the  $l$ -th layer.  $z_j^{l-1}$  and  $b_i^l$  are the corresponding input and bias, respectively.  $\alpha$  is a coefficient controlling the slope of the activation function. For simplicity, we fix the bias to zero in both ANN and SNN, as is similar to other previous works [42], [43].

We analyze the correlation between the output activations in the ANN and the firing status in the SNN. Each input pattern is presented for a total number of  $T$  time steps. Note that, we only describe the case where the neuron crosses its positive threshold for the sake of simplicity, while a similar procedure can be easily applied to describe the negative one.

From an initial position of zero, the membrane potential at  $T$  can be obtained by recursively applying Eq. (1) and Eq. (5), given by:

$$V_i^l(T) = \sum_{j=1}^{M_{l-1}} w_{ij}^l \sum_{t=1}^T o_j^{l-1}(t) - \vartheta_P \sum_{t=1}^{T-1} o_i^l(t) \quad (7)$$

According to Eq. (4), this potential can also be given as:

$$V_i^l(T) = \vartheta_P(o_i^l(T) + \sigma) \quad (8)$$

where  $\sigma$  is a residual item, given by  $\sigma = \frac{V_i^l(T)}{\vartheta_P} - \left\lfloor \frac{V_i^l(T)}{\vartheta_P} \right\rfloor$ .

Taking Eq. (7) and Eq. (8), we can further get:

$$\vartheta_P \left( \sum_{t=1}^T o_i^l(t) + \sigma \right) = \sum_{j=1}^{M_{l-1}} w_{ij}^l \sum_{t=1}^T o_j^{l-1}(t) \quad (9)$$

In order to correlate the output activation of the neuron in ANN with the firing rate of the one in SNN, we define the firing rate as

$$r_i^l(T) = \frac{N_i^l(T)}{T} = \frac{\sum_{t=1}^T o_i^l(t)}{T} \quad (10)$$

where  $N_i^l(T)$  is the number of spikes generated during the total  $T$  time steps. Eq. (9) can thereafter be converted into:

$$\vartheta_P r_i^l(T) = \sum_{j=1}^{M_{l-1}} w_{ij}^l r_j^{l-1}(T) - \frac{\sigma \vartheta_P}{T} \quad (11)$$

Eq. (11) is also a recursive expression, based on which we can take approximations layer by layer. In the first hidden layer, for simplicity, the inputs of both ANN and SNN are identical, which satisfies  $r^0(T) = z^0$ . Equating Eq. (6) and Eq. (11) yields:

$$r_i^1(T) = \frac{z_i^1}{\alpha \vartheta_P} - \frac{\sigma}{T} \quad (12)$$

which indicates that as  $T \rightarrow \infty$ , the firing rate  $r_i^1(T)$  approaches to its target value  $z_i^1$  when  $\alpha \vartheta_P = 1$ .

Augmented neurons in higher layers continuously integrate spikes from their preceding layer. We can evaluate the approximation errors in higher layers with the recursive expression of Eq. (11). Incorporating the first layer as described by Eq. (12), the firing rates in higher layers can be given as:

$$r_i^l(T) = \frac{z_i^l}{(\alpha \vartheta_P)^l} - \frac{\sigma}{T} \left( 1 + \sum_{n=2}^l \prod_{l'=n}^l \frac{\sum_{j=1}^{M^{l'-1}} w_{ij}^{l'}}{\vartheta_P} \right) \quad (13)$$

Eq. (13) implies that the approximation error accumulates in deeper layers, and as a result, more time steps are required for lossless conversion as the network structure becomes deeper.

Both Eq. (12) and Eq. (13) suggest a correlation between the firing threshold and the activation slope, i.e.  $\vartheta_P = \frac{1}{\alpha}$ , to make the firing rate of the neuron in SNN approach to the output activation of that in ANN. Similarly, the negative firing threshold can be determined by  $\vartheta_N = -\frac{1}{\alpha}$ . Therefore, the firing thresholds of converted spiking neurons can be assigned with a clear and simple rule according to our analytical descriptions rather than exhaustive manual explorations or time-consuming configurations.

Similar to the routines of a typical conversion scheme, our AugMapping will convert a pre-trained ANN to an SNN, but differently with a more direct and simple method. Importantly, our theoretical analysis supports that the as-proposed AugMapping is able to achieve nearly lossless conversion under certain conditions.

### III. EXPERIMENTAL RESULTS

#### A. Experimental Setup

The performance of our methods is extensively examined with six different networks based on various datasets including MNIST [53], and the more challenging Fashion-MNIST [54] and CIFAR10 [55].

MNIST is a handwritten digit image dataset and consists of 60,000 images for training and 10,000 images as the test set. Each sample labeled 0-9 is a grayscale image with a size of  $28 \times 28$ . Differently, Fashion-MNIST is a rather new dataset with different classes of clothing. It shares the same image size and structure of training and testing splits with MNIST, but is a more challenging image classification task. CIFAR10 contains 60,000 color images belonging to 10 classes. Each image consists of  $32 \times 32$  pixels. No data augmentation is applied on MNIST and Fashion-MNIST. For CIFAR10, we utilize a standard augmentation where each training sample are padded with 4 pixels on each side of the image, followed by a  $32 \times 32$  crop and a random horizontal flip.

Table I. Experimental network configurations

Dataset	Network	Topology
MNIST	Net1	1200-1200-10
	Net2	12c5-p2-64c5-p2-10
Fashion-MNIST	Net3	6400-10
	Net4	32c5-p2-64c5-p2-1024-10
CIFAR10	Net5	128c3-128c3-p2-256c3-256c3-p2-512c3-512c3-p2-1024-10
	Net6	VGG16 [56]

As is shown in Table I, both types of fully connected (Net1 and Net3) and convolutional networks (Net2 and Net4) are applied for MNIST and Fashion-MNIST. Two deeper networks with the VGG architecture (Net5 and Net 6) are used for CIFAR10. The detailed network structures are shown in the table, where c and p represent the convolutional and pooling layer, respectively. For example, 15c5 stands for a convolutional layer with 15 feature maps of  $5 \times 5$  kernel size, and p2 denotes a pooling layer with a receptive window of  $2 \times 2$ . A single digit such as 250 represents a fully connected layer with 250 neurons. In VGG networks, a dropout layer is used after every LeakyReLU layer except for those layers which are followed by a pooling one.

Both the training of ANNs and the inference with converted SNNs are implemented with the mainstream framework PyTorch [57]. During inference with SNNs, pixel values of images are directly fed into the first hidden layer in order to remove variability [44]. The categorical decision made by the output layer is determined by the unit that has the biggest firing rate.

#### B. Results

In order to test the efficacy of our double-threshold scheme, we first compare our TerMapping with DataNorm [43] under the same conditions. As is shown in Table II, our TerMapping is more accurate than DataNorm thanks to its extended ability with the double-threshold scheme to represent both positive and negative activations in advanced ANNs.

Table II. Accuracy compasiron between DataNorm and TerMapping

Network	Method	ANN acc.	SNN acc.
Net1	DataNorm [43]	98.68%	98.64%
	<b>TerMapping</b>	98.77%	<b>98.77%</b>
Net2	DataNorm [43]	99.14%	99.10%
	<b>TerMapping</b>	99.35%	<b>99.35%</b>

Then, we focus more on investigating the efficacy of our augmented scheme by providing more detailed comparisons between TerMapping and AugMapping. Different measurement metrics are presented in Table III.

As can be seen from the table, both AugMapping and TerMapping can achieve a nearly lossless conversion for various types of networks ranging from shallow to deep structures over different datasets. For the relatively simple task of MNIST where shallow networks are sufficient enough,

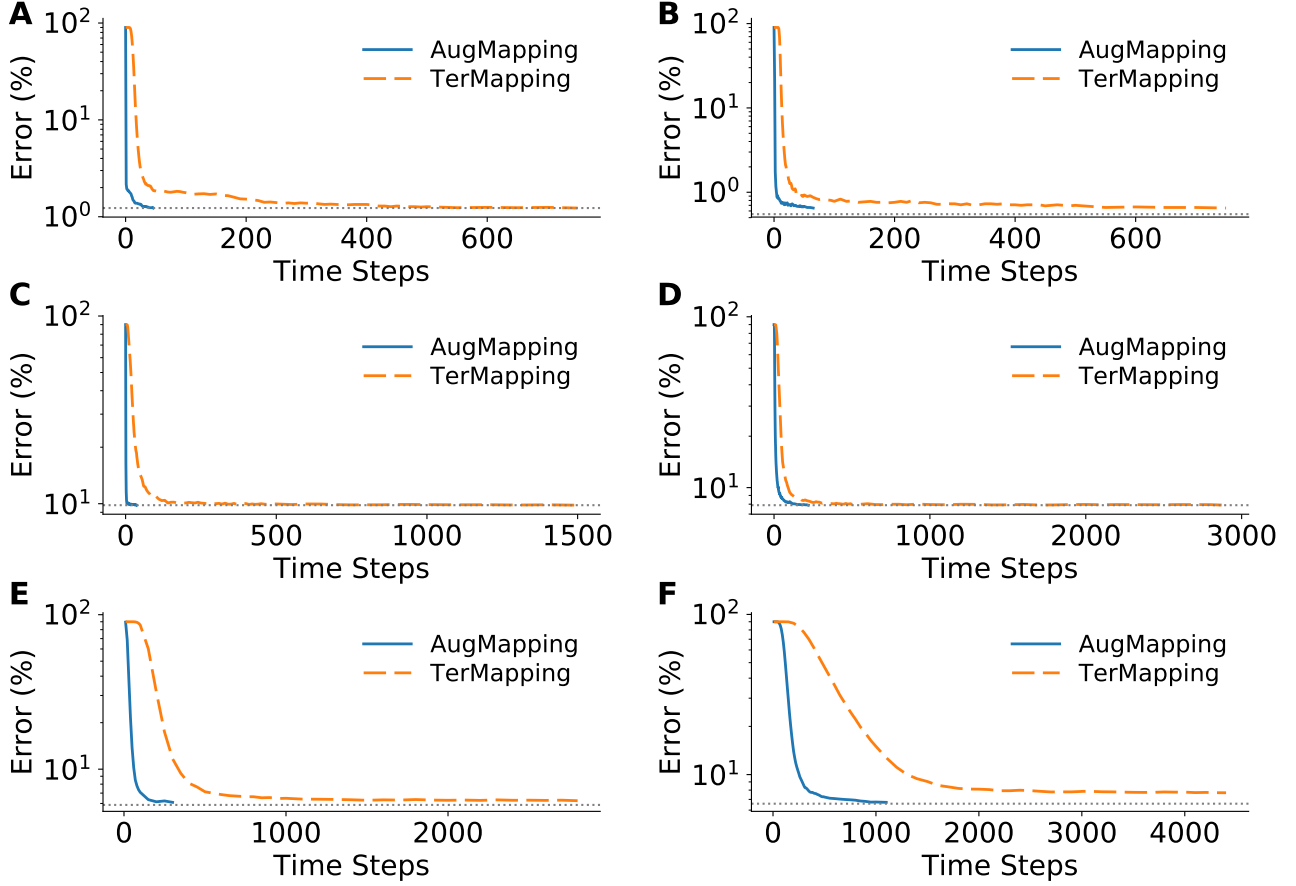


Fig. 2. Inference error versus time steps. Results for Net1 to Net6 are presented in pannel A to F accordingly. The horizontal dotted lines in each panel denote the accuracies of the corresponding ANNs.

Table III. Conversion results with our methods of TerMapping and AugMapping

Network	ANN acc.	Method	SNN acc.	Latency	# Events ( $10^6$ )
Net1	98.77%	TerMapping	98.77%	750	0.03
		AugMapping	98.77%	46	0.02
Net2	99.35%	TerMapping	99.35%	750	0.28
		AugMapping	99.35%	65	0.11
Net3	90.18%	TerMapping	90.18%	1500	0.05
		AugMapping	90.18%	37	0.03
Net4	92.11%	TerMapping	92.11%	2900	2.45
		AugMapping	92.11%	220	0.68
Net5	94.13%	TerMapping	93.75%	2800	15.36
		AugMapping	93.90%	300	12.51
Net6	93.42%	TerMapping	92.30%	4400	17.66
		AugMapping	93.29%	1100	18.10

both of our two methods can achieve the same test accuracies as the corresponding ANNs, while the AugMapping has much shorter classification latency and consumes significantly smaller number of events than the TerMapping, reflecting the efficiency in both time and energy. On the more challenging Fashion-MNIST dataset, both our methods still succeed in no-

accuracy-loss conversions with Net3 and Net4. Note that our methods significantly outperforms the recent reported result such as in [58] (around 85%), highlighting our contribution in improving the spike-based performance in accuracy.

As the task gets more challenging in CIFAR10 where significantly deep networks are adopted, reaching a lossless accuracy is getting more difficult for both of our methods within thousands of time steps. Nevertheless, our AugMapping consistently outperforms the other one with a better accuracy and lower latency. Notably, with the networks getting deeper, more events and time steps are required to achieve a nearly lossless conversion with both of our methods, being consistent with our theoretical analysis provided in Section II-D.

The inference errors of converted SNNs decrease as the number of time steps increases. The evolving details for both AugMapping and TerMapping are shown in Fig. 2. As can be seen, the AugMapping is significantly more faster than the TerMapping to reach a nearly lossless accuracy for all the networks. In most cases, the AugMapping is faster than the other one with at least one order of magnitude, while achieves a better accuracy. As the firing ability of the neuron at each time step is rather constrained in TerMapping to relieve issues of over-activation and under-activation, spiking neurons normally need a large number of time steps to precisely reflect the activation of their counterparts in ANNs. When networks



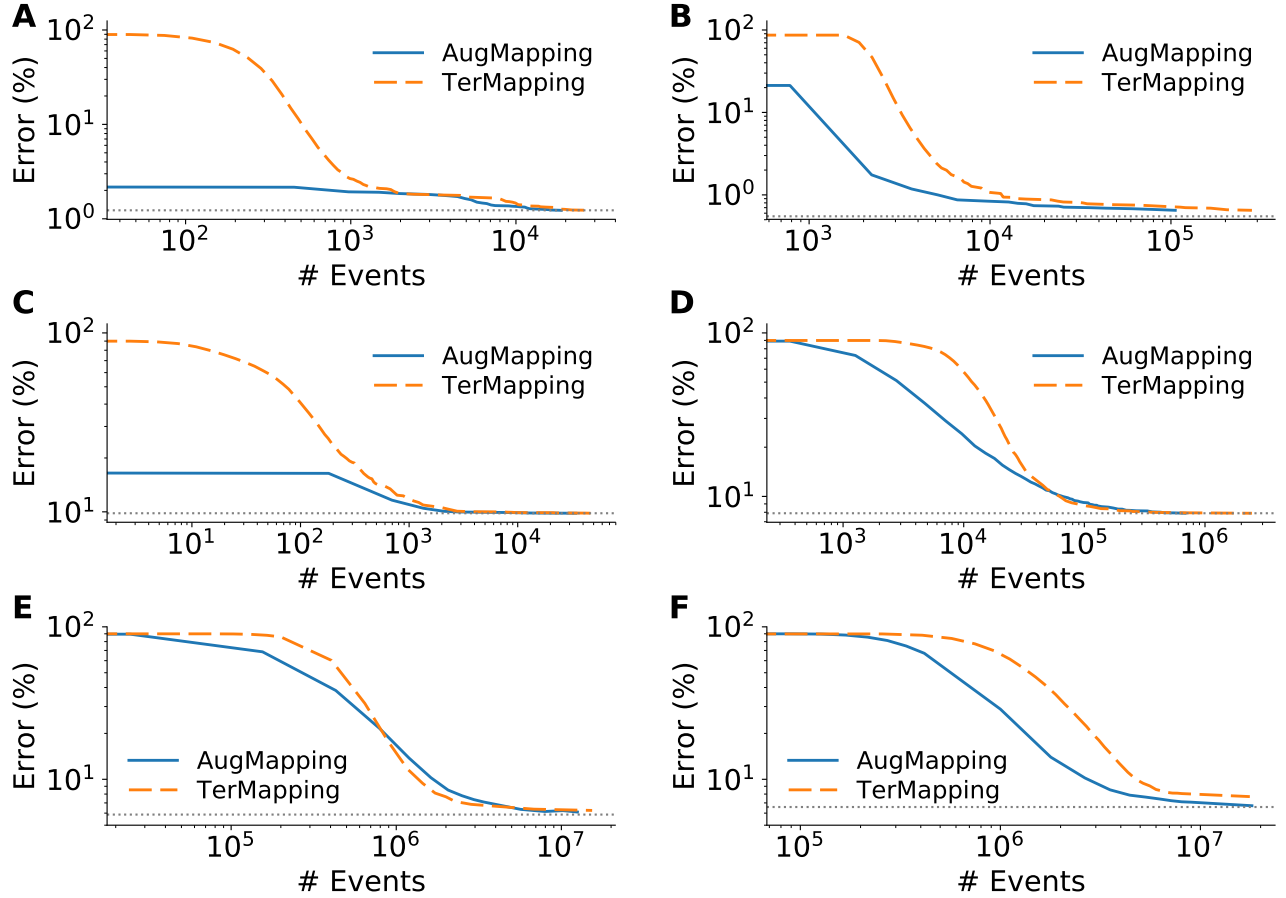


Fig. 3. Inference error versus the number of events in SNNs. **A** to **F** shows the results of Net1 to Net6 accordingly. The horizontal dotted lines in each panel represent the accuracies of the corresponding ANNs.

get deeper, significantly more time steps are required for SNNs to accumulate sufficient information through layers. In the contrary, the AugMapping only takes a small number of time steps to achieve a remarkable accuracy, owing to the enhanced capability for information representation with augmented spikes. As a result, the AugMapping is advantageous in accuracy, speed and energy-consumption.

Additionally, we perform a detailed examination on the total number of spike events consumed to achieve certain levels of accuracies. As can be seen from Fig. 3, there is a trade-off between the accuracy and the number of events for both methods across all networks. To be more specific, a high accuracy requires a large number of events, indicating a sacrifice on energy efficiency. Gladly, the headache on energy consumption can be relieved by setting an acceptable level for accuracies. Notably, the AugMapping often generates a smaller number of events than that of the TerMapping in most cases, indicating its high energy-efficiency since the power consumption of SNNs is roughly proportional to the number of events [42], [49], [59].

The above results highlight the effectiveness of both our double-threshold and augmented schemes, which play an important role in improving the performances of SNNs.

### C. Early Decision

As is compared to an ANN, SNN is advantageous in a prompt response thanks to a fast and asynchronous propagation of spikes through the network [25], and as a result, decisions can be made based on early spikes. In this part, we examine the property of our methods with respect to early decisions at the cost of a certainly small loss on accuracy. Notably, a small sacrifice on accuracy will offer the opportunity for a great level of speed acceleration and spike event reduction. In our experiments, different levels of accuracy loss are used to closely evaluate the benefits on the reduction of both latency and spike events.

Fig. 4 and Fig. 5 shows the required time steps and the number of events to achieve the acceptable accuracies with both of our methods, respectively. Note that the results for the TerMapping under several conditions are not shown because it cannot reach a satisfactory accuracy within thousands of time steps.

As can be seen from the figures, with a small compromise on acceptable accuracies, both the number of time steps and spike events required for inference can be significantly reduced. Taking Net2 as an example, a criterion of 1% accuracy loss for both our AugMapping and TerMapping can lead to the reduction of both time steps and the number of events with around two orders of magnitude as compared to

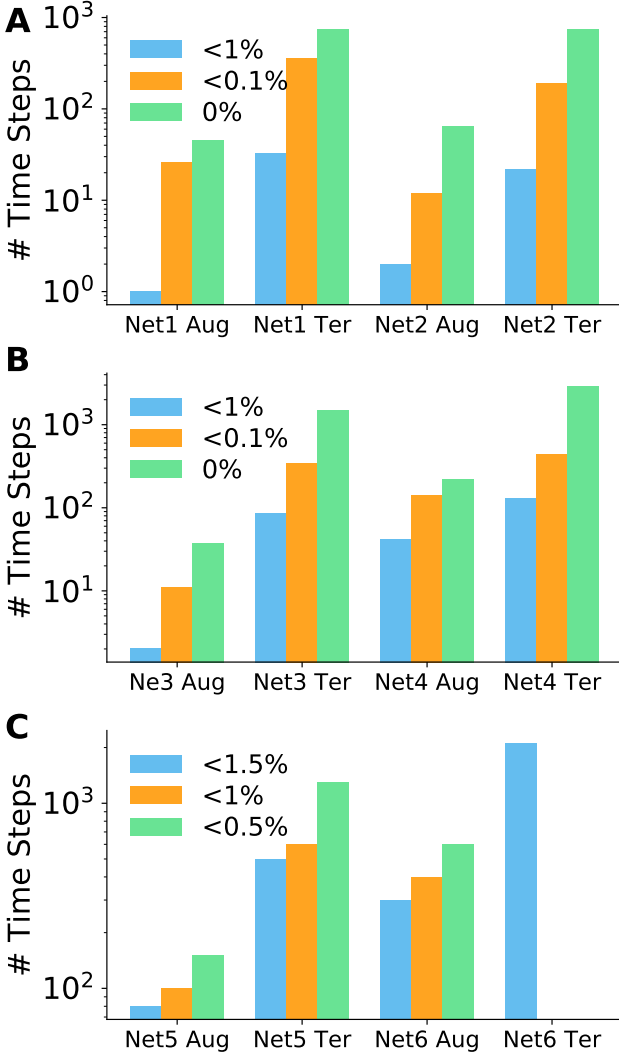


Fig. 4. The number of required time steps to reach an acceptable accuracy specified by certain loss conditions for early decisions. For example, a condition like ‘< 1%’ represents the maximally allowed accuracy loss from the ANN baseline is 1%. Some results for TerMapping on Net6 are not shown as no convergence appears within thousands of time steps.

zero-loss conversion. Although lossless accuracy would be favorable in ANN-to-SNN conversion with deep networks, early decision with a fast speed and small energy consumption could play a more important role in practical applications where efficiency in both time and energy is the main concern. Notably, the AugMapping still outperforms the TerMapping for almost every case under a given condition. Specifically, the TerMapping requires more time steps and spike events to reach a same accuracy level as the AugMapping. Additionally, for both of our methods, the bigger compromise on accuracy, the more benefit in the efficiency with respect to both time steps and spike events.

Currently, the best reported result for converting a significantly deep network of VGG16 [56] (denoted as Net6 in our study) achieves an accuracy of 91.41% with a consumption of 793 time steps and  $9.342 \times 10^6$  events [50]. As is shown in Table III, our methods result in much higher accuracies as compared to the state-of-the-art one, indicating compromises

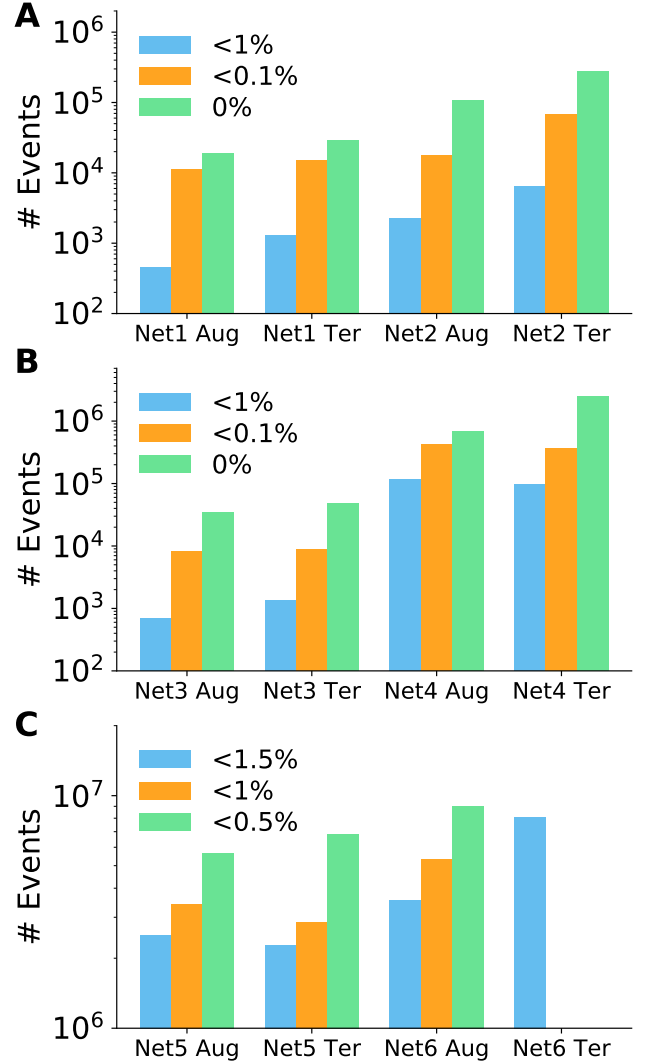


Fig. 5. Results of early decision with respect to the required number of events for a certain satisfactory accuracy.

can be made to further improve efficiency. If we lower the accuracy to 92.76% which is still significantly better than the current best, our AugMapping only requires 500 time steps and  $7.15 \times 10^6$  spike events. Hence, our method outperforms the state-of-the-art with respect to all metrics of accuracy, energy efficiency and speed, highlighting our contribution to improve the spike-based performance.

In order to better quantify the improvements of AugMapping over TerMapping for early decisions, we measure their relative performances on latency and the number of spike events under a same condition. As is shown in Table IV, AugMapping has a significantly better performance than TerMapping in terms of latency reduction, with a maximum improvement reaching  $42.5\times$  for Net3 under the condition of <1% accuracy loss. Considering all the other cases, the minimum improvement on the latency is around  $3.1\times$ . For the reduction on the number of spike events, AugMapping still outperforms the other one for most cases. Notably, if a perfect accuracy without loss is required, AugMapping achieves a minimum improvement of  $1.2\times$  on the reduction



of spike events. The above results thus highlight the advanced performance of AugMapping as is compared to TerMapping.

Table IV. Relative performance comparison between AugMapping and TerMapping for early decisions under a certain tolerance on accuracy loss, e.g. ‘<1%’. Relative reductions on latency and the number of spike events are recorded.

	Latency Reduction			Spike Events Reduction		
	<1%	<0.1%	0%	<1%	<0.1%	0%
Net1	33×	13.8×	16.3×	2.8×	1.3×	1.5×
Net2	11×	15.8×	11.5×	2.9×	3.9×	2.7×
Net3	42.5×	30.9×	40.5×	1.9×	1.1×	1.4×
Net4	3.1×	3.1×	13.2×	0.8×	0.8×	3.6×
	<1.5%	<1%	<0.5%	<1.5%	<1%	<0.5%
Net5	6.3×	6×	8.7×	0.9×	0.8×	1.2×
Net6	7×	-	-	2.3×	-	-

#### D. Evaluation of Approximation

In this part, we investigate the correlation between the output activations in ANNs and the firing rates in SNNs for both AugMapping and TerMapping, such that the conversion efficacy can be better presented. We choose Net5 as an example due to its relatively deep structure and the challenge of the task. The firing rates of the output layer are recorded during inference, which are further compared to the output activations of their counterparts in ANN.

In order to quantify the similarity between two vectors such as  $\mathbf{x}$  and  $\mathbf{y}$ , we use

$$S(\mathbf{x}, \mathbf{y}) = \cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \quad (14)$$

where  $\theta$  represents the angle between the two vectors. Eq. (14) is used to measure the similarity between the firing rates of SNN and the non-spiking output vectors of ANN.

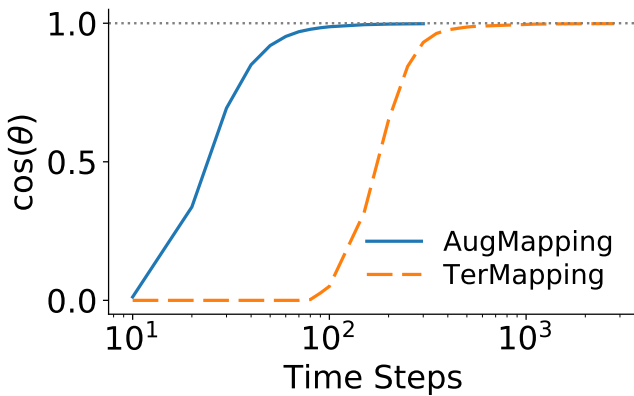


Fig. 6. Evaluation of the similarity between the firing rates in SNN and the output activations in ANN for both AugMapping and TerMapping. The horizontal dotted line indicates the perfect match.

Fig. 6 shows the similarities between the outputs of SNN and ANN during inference for both two of our conversion methods. As can be seen from the figure, both of the two methods can approach a perfect match to the one in ANN with the number of time steps for inference increasing, while

the AugMapping is still much faster than the other one. This reveals the reason underlying the high performance of a conversion scheme. As the latency increases, neurons continuously integrate information such that the firing rates gradually approximate the activations of their counterparts in ANN, thus resulting in a high and nearly lossless accuracy. This result is also consistent with our theoretical analysis.

#### E. Role of Double-Threshold Firing

Our double-threshold scheme is designed to convert both positive and negative activations from ANNs to SNNs. In this part, we will continue to examine its role on recognition performance. The deep Net5 and Net6 on the challenging CIFAR10 benchmark are selected in this experiment.

In order to show the importance of negative activations, we first conduct an experiment on the non-spiking Net5 and Net6 by suppressing all the neurons that are negatively activated. The results show that both Net5 and Net6 suffer a severe decrease, with accuracies down to 35.55% and 12.36%, respectively. This great loss in accuracy suggests that negative outputs in ANNs play a non-trivial role in transferring important information for a remarkable recognition performance. Therefore, it is important and valuable to represent negative activations in an SNN.

Next, we provide insights into the role of the double-threshold scheme in converted SNNs. Both Net5 and Net6 can be successfully converted into corresponding SNNs with nearly lossless accuracies by either AugMapping or TerMapping. In order to assess the role of our double-threshold scheme, we remove it from the converted SNNs. The SNN results for Net5 only achieve 35.4% and 32.9% with the AugMapping and the TerMapping, respectively; the accuracies for converted Net6 are even lower than a chance level. The significant degradation on accuracy caused by removing the double-threshold firing suggests its importance to realize conversion of lossless accuracy, and highlights its efficacy in representing both positive and negative activations of ANNs.

#### F. Influence of Boundary Constraint on Augmented Spikes

Our augmented spikes are capable of addressing the over-activation issue encountered by typical conversion methods [43]–[46] thanks to their advanced form to represent more information with a spike coefficient in a time step. Here, we examine the effects of spike coefficients by adding a constraint to limit their capability for information representation in a time step. We define a parameter  $M_{\text{aug}}$  controlling the maximum number of standard spikes that an augmented one can bundle in one time step. Therefore, the firing status of our augmented neurons is bounded as:

$$o_i^l(t) = \begin{cases} \min(M_{\text{aug}}, o_i^l(t)), & o_i^l(t) > 0 \\ \max(-M_{\text{aug}}, o_i^l(t)), & o_i^l(t) \leq 0 \end{cases} \quad (15)$$

In our experiment, we first run the converted SNNs without any constraint until their best accuracies are obtained, and the corresponding latency is recorded and then used in the inference with a modified SNN where the boundary constraint on augmented spikes is applied.

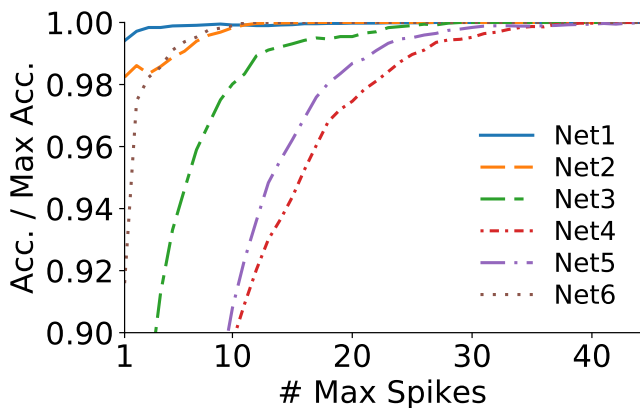


Fig. 7. Effects of boundary constraints on augmented spikes. X-axis represents the maximally permitted number of standard spikes packed by one augmented one, while Y-axis denotes the fraction of the accuracies between the constrained and the original networks configured by AugMapping.

Fig. 7 shows the effects of  $M_{\text{aug}}$  on the accuracy performance. When  $M_{\text{aug}}$  is set to 1, augmented spikes are degraded into the typical spikes since each augmented spike can only represent a status of appearance or not in a time step. As a result, recognition accuracies are decreased for all six networks, and deeper structures suffer much severer loss in accuracy. As  $M_{\text{aug}}$  increases, accuracies are gradually approaching their corresponding best, since augmented spikes are allowed to carry sufficient information with less constraints.

Notably, as can be seen from Fig. 7, with a boundary constraint on  $M_{\text{aug}}$  up to 40, our methods can successfully reach a level very close to the cases without constraints. This indicates a few bits are sufficiently enough for a good performance, being beneficial to hardware implementations.

#### G. Comparisons with Other Methods

In this part, we compare our AugMapping with other state-of-the-art conversion results on MNIST and CIFAR10, in terms of accuracy, latency, and the number of spike events as detailed in Table V. Fashion-MNIST is not included here as there are few conversion results reported for benchmarking.

The approach of phase-based weighted spikes [49] requires a shorter inference latency to reach the lossless accuracy as is compared to ours, but with a cost of significantly larger number of events, indicating a relatively lower energy efficiency. A recent work with burst spikes [50] is developed to improve the energy efficiency of converted SNNs by utilizing bursts of binary spikes, yielding a smaller number of events than [49] but still larger than ours. More importantly, our latencies outperform the previous works thanks to our augmented scheme where more information can be transmitted with one event at each time step.

Taking the accuracy as a metric for comparison, our AugMapping has a better performance than all of the other baselines for both MNIST and CIFAR10, regardless of the network type. For instance, under the same condition with the large Net6 structure and the more challenging CIFAR10 dataset, our AugMapping obtains an accuracy of 93.29% that is significantly better than the other methods. With a

different network variant of Net5, the accuracy can be further improved. The remarkable performance of our methods can be attributed to both double-threshold firing and augmented spikes, enabling spiking neurons to represent both positive and negative activations with more informative spikes, such that more advanced techniques in ANNs can be adopted for improvements. Notably, both the latency and energy efficiency of our AugMapping can be further improved with a compromise on acceptable accuracies.

In summary, our AugMapping is faster, more accurate and energy-efficient as is compared to the other state-of-the-art baselines, making it of great merit for applied implementations of deep SNNs.

## IV. DISCUSSIONS

Despite of the advantages of SNNs such as energy efficiency [15], [18], [19], their accuracies resulted from direct training mechanisms still lag far behind those of ANNs [25]–[27]. Differently, converting a pre-trained ANN to an SNN provides a straightforward and yet effective mechanism to narrow the accuracy gap between SNNs and ANNs [41], [42]. The accuracy of the converted SNN can be improved by either enhancing the one of its corresponding ANN or by reducing performance loss due to conversion. A standard SNN typically has a single firing threshold, being limited to represent positive activations only. As a result, the advanced activation functions like LeakReLU in ANNs [47], [52] cannot be fully exploited. In order to overcome this challenge, we introduce a double-threshold firing scheme, where both positive and negative thresholds are used for the neuron to elicit polarized spikes. In return, our methods can take advantages of superior ANNs to achieve better performance.

Our double-threshold firing scheme is firstly used to extend a typical threshold-balancing method of DataNorm [43], and thus TerMapping is developed. The TerMapping inherits the advantages of DataNorm and can successfully achieve nearly lossless conversion (see Table III). Differently, TerMapping is more accurate than DataNorm thanks to the double-threshold firing scheme providing approaches for mapping both positive and negative activations (see Table II). However, drawbacks from DataNorm are also brought to TerMapping, such as complicated procedures for configuring proper network parameters and inefficiency in both time and energy (see Fig. 2 and Fig. 3). Notably, our double-threshold scheme can be easily generalized to other conversion-based methods, and it could be applied to a broad range of ANNs with both positive and negative activations.

In order to further improve the efficiency of the conversion approach, we introduce a new scheme of augmented spike that employs spike coefficients to carry the number of typical all-or-nothing spikes occurring at a time step. Based on this, a new conversion method called AugMapping is developed with a clear and simple rule to assign network parameters for SNNs. Importantly, our AugMapping can not only achieve nearly lossless conversion but also consumes significantly smaller number of time steps and spike events as compared to TerMapping (see Table III and Table IV), highlighting

Table V. Comparison with the other state-of-the-art conversion methods. The digits in parentheses denote the corresponding results with a compromise on accuracy loss.

Dataset	Method	Topology	ANN acc.	SNN acc.	Latency	# Events (10 <sup>6</sup> )
MNIST	Weighted Spikes [49]	Net1	98.6%	98.6%	24	8
	<b>AugMapping</b>	Net1	98.77%	<b>98.77% (98.65%)</b>	<b>46 (23)</b>	<b>0.02 (0.01)</b>
	Weighted Spikes [49]	Net2	99.20%	99.20%	16	3
	Burst Spikes [50]	Net2	99.25%	99.25%	87	0.251
	<b>AugMapping</b>	Net2	99.35%	<b>99.35% (99.26%)</b>	<b>65 (12)</b>	<b>0.11 (0.02)</b>
CIFAR10	Cao's Method [42]	64c5-p2-64c5-p2-64c3-64-10	79.09%	77.43%	400	20
	Rueckauer's Method [44]	32c3-32c3-2s-64c3-64c3-2s-512-10	87.86%	87.82%	280	-
	Weighted Spikes [49]	32c3-32c3-p2-64c3-64c3-p2-512-10	89.1%	89.2%	117	400
	SpikeNorm [45]	Net6	91.7%	91.45%	-	-
	Burst Spikes [50]	Net6	91.41%	91.1%	1500	49.830
	<b>AugMapping</b>	Net6	93.42%	<b>93.29% (92.76%)</b>	<b>1100 (500)</b>	<b>18.10 (7.15)</b>
	<b>AugMapping</b>	Net5	94.13%	<b>93.90% (92.91%)</b>	<b>300 (100)</b>	<b>12.51 (3.40)</b>

its advantage of high efficiency in energy. Additionally, the augmented scheme endows spikes with the advanced ability to carry more information at one time step, thus significantly reducing the latency for information accumulation. As a result, our AugMapping is also efficient in time as a low latency is required. Notably, our augmented scheme could be also favorable for hardware implementations since a few bits could be sufficiently enough for a high performance (see Fig. 7).

Three realistic datasets with various networks are used to investigate the effectiveness of our methods. The better performance of our methods over the state-of-the-art baselines (see Table V) highlight the potential merit of our approaches. In addition to the image recognition tasks, our methods could be easily generalized to other challenging problems such as object detection, for which our advantageous performance could be favorable and beneficial.

## V. CONCLUSION

In this work, we first introduced a double-threshold scheme for SNNs to fully benefit from advanced ANNs in the ANN-to-SNN conversion. Then, we developed TerMapping by extending the DataNorm method with our double-threshold scheme. Moreover, another new scheme of augmented spikes is introduced to represent more information at one time step. Accordingly, a new AugMapping was developed for ANN-to-SNN conversion, but importantly with a simple and clear rule to configure spiking neurons in contrast to the complicated threshold-balancing approaches in other related works. We investigated the performance of our methods with various networks based on three challenging datasets. Experimental results show that our double-threshold scheme benefits the improvement of accuracy for SNNs. Moreover, our advanced AugMapping is more advantageous for constructing accurate, fast and efficient deep SNNs than the state-of-the-art baselines, which could be greatly valuable for neuromorphic computing.

## REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [3] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.
- [4] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.
- [5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [6] W. Severa, C. M. Vineyard, R. Dellana, S. J. Verzi, and J. B. Aïme, "Training deep neural networks for binary communication with the whetstone method," *Nature Machine Intelligence*, vol. 1, no. 2, pp. 86–94, 2019.
- [7] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 2015, pp. 1135–1143.
- [8] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: training deep neural networks with binary weights during propagations," in *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 2015, pp. 3123–3131.
- [9] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*, 2016, pp. 525–542.
- [10] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," in *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.
- [11] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [12] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of Neural Science*. McGraw-hill New York, NY, USA, 2000, vol. 4.
- [13] R. A. Nawrocki, R. M. Voyles, and S. E. Shaheen, "A mini review of neuromorphic architectures and implementations," *IEEE Transactions on Electron Devices*, vol. 63, no. 10, pp. 3819–3829, 2016.
- [14] Q. Yu, H. Li, and K. C. Tan, "Spike timing or rate? neurons learn to make decisions for both through threshold-driven plasticity," *IEEE transactions on cybernetics*, vol. 49, no. 6, pp. 2178–2189, 2018.
- [15] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.
- [16] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. P. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.
- [17] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

- [18] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
- [19] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.
- [20] R. Güttig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nature neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.
- [21] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 10, pp. 1539–1552, 2013.
- [22] Q. Yu, R. Yan, H. Tang, K. C. Tan, and H. Li, "A spiking neural network system for robust sequence recognition," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 3, pp. 621–635, 2015.
- [23] A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 11, pp. 5394–5407, 2018.
- [24] C. Hong, X. Wei, J. Wang, B. Deng, H. Yu, and Y. Che, "Training spiking neural networks for cognitive tasks: A versatile framework compatible with various temporal codes," *IEEE transactions on neural networks and learning systems*, 2019.
- [25] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, 2018.
- [26] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. S. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.
- [27] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Networks*, vol. 122, pp. 253–272, 2020.
- [28] G. qiang Bi and M. ming Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *The Journal of Neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.
- [29] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.
- [30] M. Mozafari, S. R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, and M. Ganjtabesh, "First-spike-based visual categorization using reward-modulated stdp," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6178–6190, 2018.
- [31] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "StdP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.
- [32] C. Lee, G. Srinivasan, P. Panda, and K. Roy, "Deep spiking convolutional neural network trained with unsupervised spike timing dependent plasticity," *IEEE Transactions on Cognitive and Developmental Systems*, 2018.
- [33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 696–699, 1988.
- [34] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1–4, pp. 17–37, 2002.
- [35] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 3227–3235, 2017.
- [36] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, 2018.
- [37] Y. Jin, W. Zhang, and P. Li, "Hybrid macro/micro level backpropagation for training deep spiking neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7005–7015.
- [38] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems*, 2018, pp. 1412–1421.
- [39] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, 2019.
- [40] P. Gu, R. Xiao, G. Pan, and H. Tang, "Stca: spatio-temporal credit assignment with delayed feedback in deep spiking neural networks," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 1366–1372.
- [41] J. A. Pérez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2706–2719, 2013.
- [42] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.
- [43] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.
- [44] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017.
- [45] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, 2019.
- [46] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-yolo: Spiking neural network for energy-efficient object detection," in *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.
- [47] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
- [48] R. Chen, H. Ma, S. Xie, P. Guo, P. Li, and D. Wang, "Fast and efficient deep sparse multi-strength spiking neural networks with dynamic pruning," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.
- [49] J. Kim, H. Kim, S. Huh, J. Lee, and K. Choi, "Deep neural networks with weighted spikes," *Neurocomputing*, vol. 311, pp. 373–386, 2018.
- [50] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," in *Proceedings of the 56th Annual Design Automation Conference 2019*. ACM, 2019, p. 53.
- [51] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [52] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models."
- [53] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [54] H. Xiao, K. Rasul, and R. Vollgraf, (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
- [55] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
- [58] Y. Hao, X. Huang, M. Dong, and B. Xu, "A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule," *Neural Networks*, vol. 121, pp. 387–395, 2020.
- [59] S. Krithivasan, S. Sen, S. Venkataramani, and A. Raghunathan, "Dynamic spike bundling for energy-efficient spiking neural networks," in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.



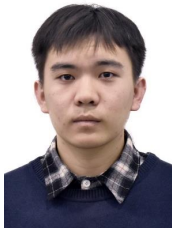
**Qiang Yu** (M'12) received the B.Eng. degree in electrical engineering and automation from the Harbin Institute of Technology, Harbin, China, in 2010, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2014.

He is an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. Before that, he was a Post-Doctoral Research Fellow with the Max-Planck-Institute for Experimental Medicine, Göttingen, Germany, from 2014 to 2016, and a Research Scientist in the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore, from 2016. He is a recipient of the 2016 IEEE Outstanding TNNLS Paper Award. His current research interests include learning algorithms in spiking neural networks, neural coding, cognitive computations and machine learning.



**Jianwu Dang** (M'12) graduated from Tsinghua Univ., China, in 1982, and got his M.S. degree at the same university in 1984. He worked for Tianjin Univ. as a lecture from 1984 to 1988. He was awarded the PhD degree from Shizuoka Univ., Japan in 1992. He worked for ATR Human Information Processing Labs., Japan, as a senior researcher from 1992 to 2001. He joined the University of Waterloo, Canada, as a visiting scholar for one year from 1998. Since 2001, he has worked for Japan Advanced Institute of Science and Technology (JAIST) as a

professor. He joined the Institute of Communication Parlee (ICP), Center of National Research Scientific, France, as a research scientist the first class from 2002 to 2003. Since 2009, he has joined Tianjin University, Tianjin, China. His research interests are in all the fields of speech science including brain science, and speech signal processing. He built MRI-based bio-physiological models for speech and swallowing, and endeavors to apply these models on clinics.



**Chenxiang Ma** received the B.Eng. degree from the China University of Petroleum, Qingdao, China, in 2019. He is currently pursuing the master's degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His current research interests include learning algorithms in spiking neural network and deep learning.



**Shiming Song** received the bachelor's degree from Sichuan University, Chengdu, China, in 2018. He is currently pursuing the master's degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His current research interests include spike-based learning, neural encoding and machine learning.



**Kay Chen TAN** (SM'08-F'14) received the B.Eng. (First Class Hons.) degree in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and five books.

Dr. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, was the Editor-in-Chief of the IEEE Computational Intelligence Magazine from 2010 to 2013, and currently serves as the Editorial Board Member of over 20 journals. He is an elected member of the IEEE CIS AdCom from 2017 to 2019 and is an IEEE CIS Distinguished Lecturer from 2015 to 2017.



**Gaoyan Zhang** received the B.S. degree in communication engineering from Henan Normal University, Xinxiang, China, in 2008 and the Ph.D. degree in cognitive neuroscience from Beijing Normal University, Beijing, China in 2014. She is an Associate Professor with College of Intelligence and Computing, Tianjin University, Tianjin, China since 2014. From 2017 to 2018, she was a visiting scholar in National Institutes of Health, USA. Her research interest includes brain signal and information processing, especially fMRI and EEG data analysis and

modeling.