



OPEN ACCESS

EDITED BY

Anup Das,
Drexel University, United States

REVIEWED BY

Qi Xu,
Dalian University of Technology, China
Oliver Rhodes,
The University of Manchester, United Kingdom
Peng Li,
Tianjin University, China
Tielin Zhang,
Chinese Academy of Sciences (CAS), China

*CORRESPONDENCE

Liang-Jian Deng
✉ liangjian.deng@uestc.edu.cn

†These authors have contributed equally to this work and share first authorship

RECEIVED 06 November 2022

ACCEPTED 28 April 2023

PUBLISHED 23 May 2023

CITATION

Qiu X-R, Wang Z-R, Luan Z, Zhu R-J, Wu X, Zhang M-L and Deng L-J (2023) VTSNN: a virtual temporal spiking neural network. *Front. Neurosci.* 17:1091097. doi: 10.3389/fnins.2023.1091097

COPYRIGHT

© 2023 Qiu, Wang, Luan, Zhu, Wu, Zhang and Deng. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). 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.

VTSNN: a virtual temporal spiking neural network

Xue-Rui Qiu^{1†}, Zhao-Rui Wang^{1†}, Zheng Luan^{1†}, Rui-Jie Zhu², Xiao Wu³, Ma-Lu Zhang⁴ and Liang-Jian Deng^{3*}

¹School of Optoelectronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, ²School of Public Affairs and Administration, University of Electronic Science and Technology of China, Chengdu, China, ³School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China, ⁴School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

Spiking neural networks (SNNs) have recently demonstrated outstanding performance in a variety of high-level tasks, such as image classification. However, advancements in the field of low-level assignments, such as image reconstruction, are rare. This may be due to the lack of promising image encoding techniques and corresponding neuromorphic devices designed specifically for SNN-based low-level vision problems. This paper begins by proposing a simple yet effective undistorted weighted-encoding-decoding technique, which primarily consists of an Undistorted Weighted-Encoding (UWE) and an Undistorted Weighted-Decoding (UWD). The former aims to convert a gray image into spike sequences for effective SNN learning, while the latter converts spike sequences back into images. Then, we design a new SNN training strategy, known as Independent-Temporal Backpropagation (ITBP) to avoid complex loss propagation in spatial and temporal dimensions, and experiments show that ITBP is superior to Spatio-Temporal Backpropagation (STBP). Finally, a so-called Virtual Temporal SNN (VTSNN) is formulated by incorporating the above-mentioned approaches into U-net network architecture, fully utilizing the potent multiscale representation capability. Experimental results on several commonly used datasets such as MNIST, F-MNIST, and CIFAR10 demonstrate that the proposed method produces competitive noise-removal performance extremely which is superior to the existing work. Compared to ANN with the same architecture, VTSNN has a greater chance of achieving superiority while consuming $\sim 1/274$ of the energy. Specifically, using the given encoding-decoding strategy, a simple neuromorphic circuit could be easily constructed to maximize this low-carbon strategy.

KEYWORDS

spiking neural networks, undistorted weighted-encoding/decoding, neuromorphic circuits, Independent-Temporal Backpropagation, biologically-inspired artificial intelligence

1. Introduction

Spiking Neural Networks (SNNs) are artificial neural networks of the “third generation” that closely resemble natural neural networks (Maass, 1997). Since biological motion processing depends on temporal information and gains superb performances (Saygin, 2007). Researchers attempt to use SNN to convert spatial complication to temporal complication. Since the information is transmitted in the form of spikes. It also has a lower carbon footprint (Roy et al., 2019) and superior robustness (Sironi et al., 2018). SpikeProp (Bohte et al., 2002) initially updated weights using SNN with backpropagation and supervised learning. Few studies are devoted to low-level image tasks with supporting neuromorphic chips, and the

majority of SNNs are currently focused on classification (Xing et al., 2020; Fang et al., 2021; Zheng et al., 2021).

In addition, the vast majority of SNNs designed for low-level tasks require specialized hardware such as event-based cameras (Zhang et al., 2021; Zhu et al., 2022). This requirement substantially raises the bar for usage. Pioneers in this area introduced a novel SNN, requiring no specialized hardware (Comşa et al., 2021). Their performance, however, is not ideal, and our work will improve it. Since 2002 (Bohte et al., 2002), the surrogate gradient has been commonly employed for backpropagation in SNN, then Neftci et al. (2019) introduced Backpropagation Through Time (BPTT) to this area. Besides, Deng et al. (2022) assert standard direct training by utilizing a formula to distinguish it from ANN-SNN conversion. Also, hybrid ANN-SNN conversion requires additional time steps and must be shadow trained exclusively (Eshraghian et al., 2021). Inspired by related studies (Werbos, 1990), Spatio-Temporal Backpropagation (STBP) is introduced.

Since 2002 (Bohte et al., 2002), the surrogate gradient has been commonly employed for backpropagation in SNN, then Neftci et al. (2019) introduced Backpropagation Through Time (BPTT) to this area. Besides, Deng et al. (2022) assert standard direct training by utilizing a formula to distinguish it from ANN-SNN conversion. Also, hybrid ANN-SNN conversion requires additional time steps and must be shadow trained exclusively (Eshraghian et al., 2021). Inspired by related studies (Werbos, 1990), Spatio-Temporal Backpropagation (STBP) is introduced.

Other related approaches include Temporal Spike Sequence Learning via Backpropagation (TSSL-BP) (Zhang and Li, 2020) but only appropriate for the classification task. For the low-level denoising assignment in this work, STBP performs worse (Comşa et al., 2021) than our Independent-Temporal Backpropagation (ITBP).

Rate coding, temporal coding, delta modulation, and direct coding are four common encoding methods. Among them, delta modulation and rate coding lose pixel location information (Kim et al., 2022). Direct coding can maintain location information, but it cannot be analyzed quantitatively (Jin et al., 2022). Weighted phase spiking coding (a type of temporal coding) employs the binary encoding concept (Kim et al., 2018). But it is also distorted and requires a normalization trick. Comşa et al. (2021) employed a latency coding method called time-to-first-spike (TTFS), inspired by biological vision (Hubel and Wiesel, 1962), to represent pixel brightness. TTFS cannot guarantee undistorted results, needs more time steps, and performs worse than ours. The classification task does not generate images; consequently, there are few decoding methods for low-level tasks such as reconstruction. Membrane Potential Decoding (MPD) (Kamata et al., 2022) is, to the best of our knowledge, the only appropriate decoding method. However, MPD generates floating results, necessitating the inclusion of a surrogate function. Prior to our work, there was no symmetric and undistorted SNN encoding-decoding method. Rate coding, temporal coding, delta modulation, and direct coding are four common methods of encoding. Among them, delta modulation and rate coding lose pixel location information (Kim et al., 2022). Direct coding can maintain location information, but it cannot be analyzed quantitatively (Jin et al., 2022). Weighted phase spiking coding (a type of temporal coding) employs the binary

encoding concept (Kim et al., 2018). But it is also distorted and requires a normalization trick. Comşa et al. (2021) employed a latency coding method called time-to-first-spike (TTFS), inspired by biological vision (Hubel and Wiesel, 1962), to represent pixel brightness. TTFS cannot guarantee undistorted results, needs more time steps, and performs worse than ours. The classification task does not generate images; consequently, there are few decoding methods for low-level tasks such as reconstruction. Membrane Potential Decoding (MPD) (Kamata et al., 2022) is, to the best of our knowledge, the only appropriate decoding method. However, MPD generates floating results, necessitating the inclusion of a surrogate function. Prior to our work, there was no symmetric and undistorted SNN encoding-decoding method.

This paper here presents a Virtual Temporal Spiking Neural Network (VTSNN) for image reconstruction. VTSNN is based on a modified U-net (Ronneberger et al., 2015) which is a classical architecture. There are many works that apply U-shape architecture to do image reconstruction tasks such as image denoising and achieving promising results (Yue et al., 2020; Cheng et al., 2021; Zamir et al., 2021; Wang et al., 2022). Alternatively, we propose an Undistorted Weighted-Encoding-Decoding method for converting an arbitrary image into binary data (0/1) in order to efficiently encode image data. We also demonstrate that this encoding-decoding procedure can be performed by simple neuromorphic circuits, thereby increasing its effectiveness. The schematic diagram of the circuits consists of ADC and DAC. Additionally, we propose a novel backpropagation technique called Independent-Temporal Backpropagation (ITBP) to avoid the inefficiency of Spatio-Temporal Backpropagation (STBP) (Wu et al., 2018). The main contributions of this paper can be summarized as follows:

- We propose, to the best of our knowledge, the first symmetric and undistorted encoding-decoding approach with high efficiency for fully spiking SNN-based image reconstruction tasks that can be implemented using simple neuromorphic circuits. This raises the prospect of low-level tasks being applied to neuromorphic devices.
- First, we introduce a virtual temporal SNN. This suggests that even without temporal information, SNN can be used to achieve competitive performance. A novel backpropagation for direct training, called ITBP, is also proposed for the designed encoding-decoding technique to improve effectiveness.
- Experimental results on a variety of datasets are often superior to the current SNN-based approach (Comşa et al., 2021) while superior to same-architecture ANN in some cases. In addition, VTSNN uses roughly 1/274 of the energy of ANN-based methods.

2. Method

Based on our analysis, the application of SNN and its neuromorphic devices is almost limited to the classification task. Consequently, we intend to investigate SNN's capabilities for low-level image tasks, such as reconstruction. In the meantime, popular input encoding methods have numerous shortcomings, including

redundant time steps and information distortion. Moreover, studies on output decoding are quite rare. Therefore, we propose a novel symmetric and undistorted encoding-decoding method to fill the above gaps. Currently, researchers generally use STBP for the low-level SNN task (Comşa et al., 2021), which allows information to propagate in both temporal and spatial domains. Therefore, we present a new backpropagation that only permits information to propagate via the spatial domain. This backpropagation can improve the effectiveness and gain better performance. In addition, we want to use simple neuromorphic circuits to demonstrate the feasibility of our encoding-decoding method. With undistorted and symmetric encoding/decoding, simpler and more effective backpropagation, and fewer time steps, we aim to achieve competitive performance in low-level image reconstructing.

2.1. Preliminary

2.1.1. Spiking neurons

Since 1907 (Lapique, 1907), qualitative scientific study has been conducted on the membrane voltage of neurons. Compared to the many-variable and intricate H-H model (Hodgkin and Huxley, 1952), the integrate-and-fire (IF) neuron model and leaky-integrate-and-fire (LIF) neuron model have a significantly reduced computational demand and are commonly recognized as the simplest models among all popular neuron models while retaining biological interpretability (Burkitt, 2006). The spiking neuron model is characterized by the following differential equation (Gerstner et al., 2014):

$$\tau \frac{du(t)}{dt} = -u(t) + x(t) \tag{1}$$

Where $u(t)$ represents the membrane potential of the neuron at time step t , $x(t)$ represents the input from the presynaptic neurons, and τ is a time constant. What's more, spikes will fire if $u(t)$ exceeds the threshold V_{th} . The spiking neuron models can be described explicitly iteratively to improve computational traceability.

$$x_{t+1,n}^i = \sum_j w_{jn}^i o_{t+1,n-1}^j \tag{2}$$

$$u_{t+1,n}^i = u_{t,n}^i g(o_{t,n}^i) + x_{t+1,n}^i \tag{3}$$

$$o_{t+1,n}^i = h(u_{t+1,n}^i - V_{th}) \tag{4}$$

Here, t and n , respectively, represent the indices of the time step and n -th layer, and o^j is its binary output of j -th neuron. Furthermore, w^j is the synaptic weight from j -th neuron to i -th neuron, and by altering the way that w^j is linked, we can implement convolutional layers, fully connected layers, etc. To be more precise, the spiking neurons become the IF neuron if $g(x) = \tau$ and the LIF neuron if $g(x) = \tau e^{-\frac{x}{\tau}}$. Since $h(\cdot)$ represents the Heaviside function and Equation (4) is non-differentiable. The following derivatives of the surrogate function can be used for approximation.

$$\frac{\partial o_{t+1,n}^i}{\partial u_{t+1,n}^i} = \frac{1}{1 + (\pi x_{t+1,n}^i)^2} \tag{5}$$

The working schematic of spiking neurons is shown in Figure 1 (Eshraghian et al., 2021).

2.1.2. Tensor multiplication

In Section 2.4, a transform pair for tensors are used to describe the decoding process. To better understand that process, here we first give some preliminary tensor definitions. A tensor with N dimensions is defined as $\mathcal{P} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. Elements of \mathcal{P} are denoted as p_{i_1, i_2, \dots, i_N} , where $1 \leq i_n \leq I_N$. The n -mode unfolding vectors of tensor \mathcal{P} are the I_n -dimensional vectors obtained from \mathcal{P} by changing index i_n while keeping the other indices fixed. The n -mode unfolding matrix $P_{(n)} \in \mathbb{R}^{I_n \times I_2 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N}$ is defined by arranging all the n -mode vectors as the columns of the matrix (Kolda, 2006). The n -mode product of the tensor $\mathcal{P} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with the matrix $B \in \mathbb{R}^{I_n \times I_n}$, denoted by $\mathcal{P} \times_n B$, is an N -dimensional tensor $\mathcal{Q} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N}$. Hence, we have the following transform pair that will be used later in the image decoding process.

$$\mathcal{Q} = \mathcal{P} \times_n B \Leftrightarrow Q_{(n)} = BP_{(n)} \tag{6}$$

2.2. Virtual temporal SNN

In this section, we propose and describe the concept of Virtual Temporal SNN (VTSNN):

VTSNN is an abstract SNN definition that uses raw static data to generate spiking sequences (0/1) as network input, and the sequences are virtually ordered in the temporal domain.

Specifically, VTSNN holds the following fundamental:

The raw static data consists of non-temporal information and will be transformed into ordered sequences (a static encoding process), such as the operation of event-based hardware, rate coding, direct coding, etc.

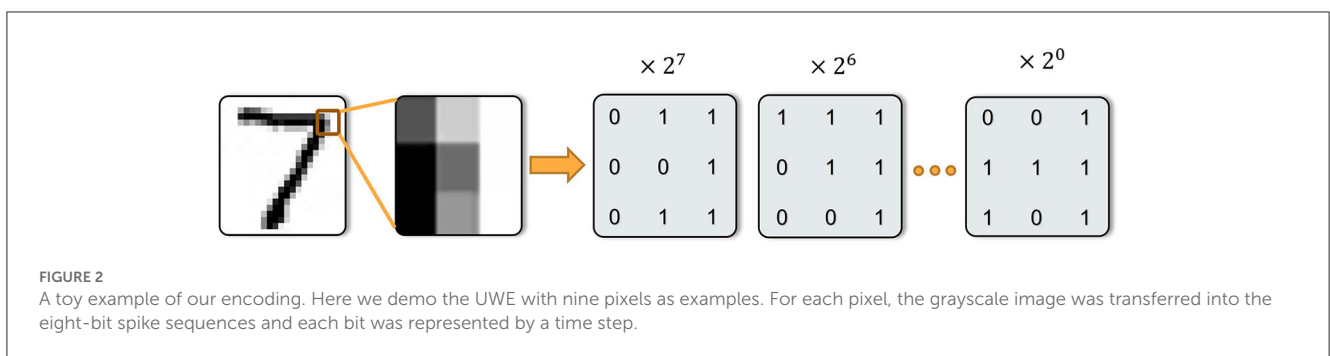
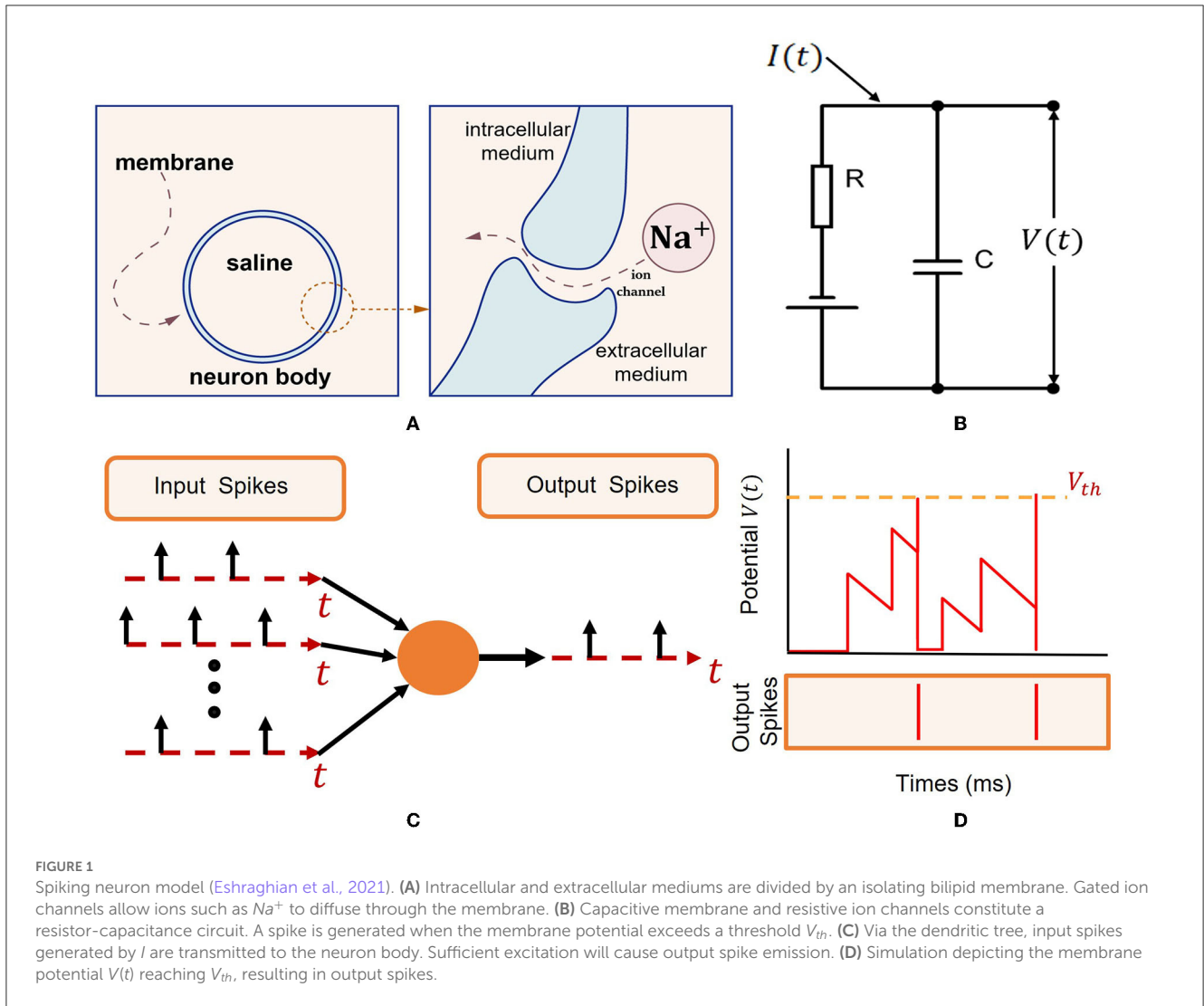
To realize the VTSNN, the crucial factors are to carefully design the corresponding encoding and decoding strategies which will be illustrated in detail.

2.3. Encoding

2.3.1. Rethinking time-to-first-spike encoding (TTFS)

Previous study (Comşa et al., 2021) has applied a TTFS encoder to encode more salient information as earlier spikes and gained good results in reconstruction tasks. This encoding method is inspired by the idea of a rapid information process with spiking data (Thorpe et al., 2001). Here r_t^{ij} is the response of a pixel of an image at time step t . Equation (7) shows the calculation of r_t^{ij} for TTFS.

$$r_t^{ij} = \begin{cases} 1 & t = (\frac{\max(x) - x^{ij}}{\max(x)})T \\ 0 & \text{otherwise} \end{cases} \tag{7}$$



In terms of r_t^{ij} , after obtaining it, spike sequences are generated using the same methods as Algorithm 1 in this paper. There are two obvious disadvantages of TTFS.

- TTFS is distorted, which means not being capable of restoring information after coding, if solely uses a function to approach it.

2.3.2. Undistorted weighted-encoding (UWE)

In this section, we propose the so-called Undistorted Weighted-Encoding (UWE) to encode the input images into spike sequences, as opposed to distorted encoders such as the time-to-first-spike (TTFS) encoder. Specifically, UWE can encode n -bit image $[0, 2^n - 1]$ theoretically and a toy example of our coding is shown in Figure 2. In what follows, Algorithm 1 illustrates the process of encoding an image.

```

Input: Undistorted Weighted-Encoding;  $n$ -bit image:
           $x$ ; Image shape:  $H, W$ ; Simulation length:  $T$ 
Output: Spiking sequences:  $\tilde{r}_t$ 
Initialization:
for  $t = T - 1, T - 2, \dots, 0$  do
  for  $i = 0, 1, \dots, H - 1; j = 0, 1, \dots, W - 1$  do
     $a_t^{ij} = x^{ij}$ .
     $r_t^{ij} = \lfloor \frac{a_t^{ij}}{2^t} \rfloor$ .
    if  $t \neq 0$  then
       $a_{t-1}^{ij} = a_t^{ij} \bmod 2^t$ .
    else
      break.
    end
    Append  $r_t^{ij}$  to  $r_t \in \mathbb{R}^{1 \times HW} = \{r_t^{H-1, W-1}, \dots, r_t^{0,1}, r_t^{0,0}\}$ .
    Reshape it to  $r_t \in \mathbb{R}^{H \times W}$ .
  end
  Append  $r_t$  to the spiking sequences:
   $\tilde{r}_t = \{r_{T-1}, \dots, r_1, r_0\}$ .
end

```

Algorithm 1. UWE algorithm for n -bit image.

For simplicity, we set $n = 8^1$ in our work, since the inputs are 8-bit image $[0, 255]$. Thus, we use 8-bit UWE in this work. Especially, in Algorithm 1, x^{ij} is a pixel in the image x . After input encoding process, x is transferred into the spiking sequences $\tilde{r}_t \in \mathbb{R}^{T \times H \times W}$ and $r_t \in \mathbb{R}^{H \times W}$ accumulates information for each time step. Additionally, UWE is capable to be easily integrated with a neuromorphic chip which is introduced in the discussion part. This means n -bit image can be transferred into spiking sequences by the neuromorphic SAR ADC circuits (discussed in Section 4.4) without any floating arithmetic.

2.4. Decoding

2.4.1. Rethinking membrane potential decoding (MPD)

The decoding method that (Kamata et al., 2022) uses for reconstruction tasks is categorized as MPD. Actually, MPD is similar to our Undistorted Weighted-Decoding (UWD) to some extent. This method, like UWD, applies a weight series to encode. However, the weight values of MPD are from 2 to 0.8 and it calls a float artificial neuron (\tanh function) before returning outputs. This means the n -bit decoding matrix A in UWD is adjusted to $\Theta = \{\theta^{T-1}, \theta^{T-2}, \dots, \theta^0\}$ and $\theta = 0.8$, then a \tanh function is used to get the real-valued reconstructed image \hat{Y} . The mechanism of UWD will be introduced in the next section. Furthermore, there is a noticeable disadvantage: MPD will induct floating arithmetic, which is unfriendly to neuromorphic chips.

2.4.2. Undistorted weighted-decoding

To overcome the disadvantages of existing decoders, we also present an Undistorted Weighted-Decoding (UWD) to decode the output spiking sequences $\hat{o}_t \in \mathbb{R}^{H \times W}$ ($t = T - 1, T - 2, \dots, 0$) into the final image \hat{Y} . This decoding process is actually a symmetric process of UWE, which means UWD will transform the spiking sequences into a n -bit image. According to the preliminary, we use the output spiking sequences \hat{o}_t ($t = T - 1, T - 2, \dots, 0$) to build a tensor $\hat{O} \in \mathbb{R}^{T \times H \times W}$. Then, we define a n -bit decoding matrix $A \in \mathbb{R}^{1 \times T} = \{2^{T-1}, 2^{T-2}, \dots, 2^0\}$. Similar to UWE, we also set $T = 8$ in the decoding process. In Section 2.1.2, we have already introduced tensor multiplication. The final decoding process can be described by the following formula:

$$\hat{Y} = \hat{O} \times_n A \Leftrightarrow \hat{Y}_{(3)} = A\hat{O}_{(3)} \tag{8}$$

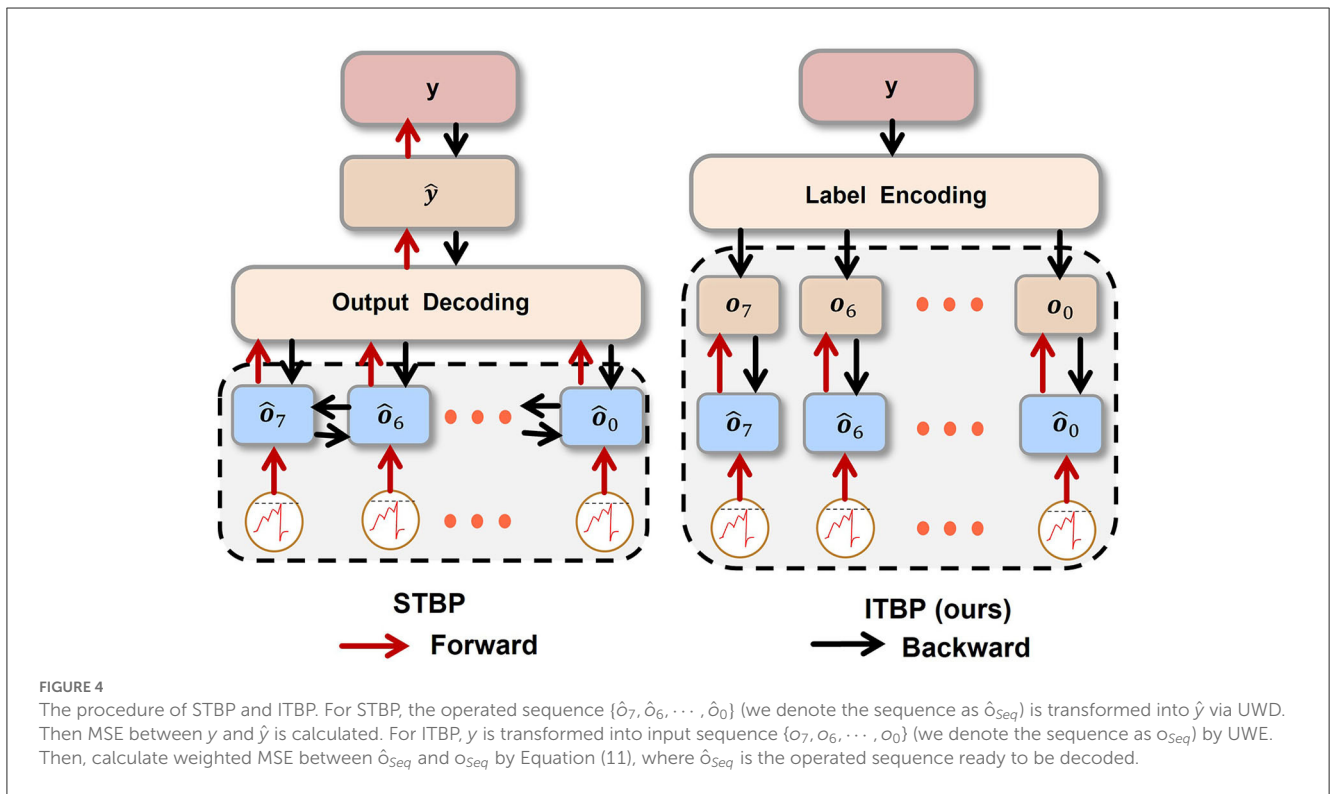
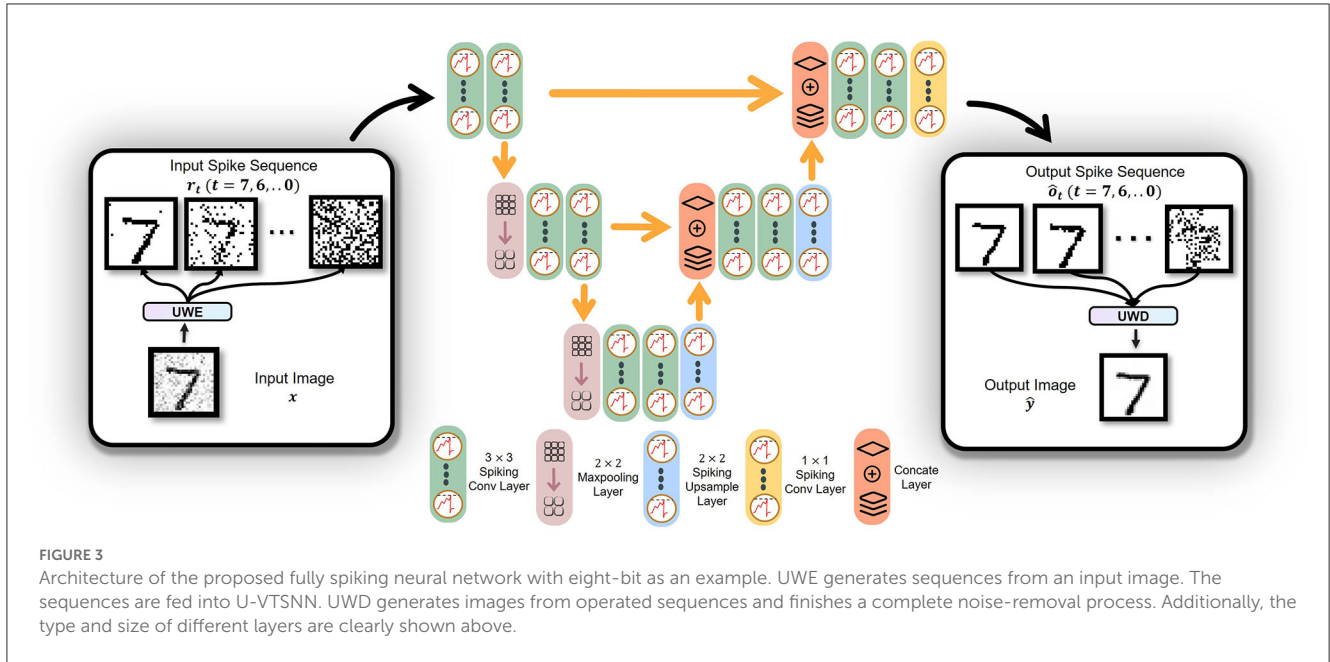
Where $\hat{O}_{(3)} \in \mathbb{R}^{T \times HW}$ is the 3-mode unfolding matrix of \hat{O} while $\hat{Y}_{(3)} \in \mathbb{R}^{1 \times HW}$ is the 3-mode unfolding matrix of $\hat{Y} \in \mathbb{R}^{H \times W \times 1}$. Hence, from the knowledge of tensor and transform pair effectively introduced in the preliminary part (Equation 6), we can get the final output image \hat{Y} . As the parallel inverse process of UWE, the decoding method can be realized by the neuromorphic chip we discussed later as well. The neuromorphic DAC circuits (discussed in Section 4.4) can convert spiking sequences to a real-valued reconstructed image without the use of floating-point arithmetic.

2.5. Spiking neural network architecture

As an abstract and flexible concept, VTSNN can be applied to various types of network architectures. In this work, our VTSNN is embedded in a shallow U-net architecture, named U-VTSNN. Because light U-net can extract features from images relatively efficiently. Additionally, unlike current SNNs for low-level image tasks whose data flow may contain floating numbers (Zhu et al., 2022), the U-VTSNN is a fully spiking neural network where all modules are built with SNN and all synapse operations are completed by spiking neurons (Kamata et al., 2022). In addition, U-VTSNN is a fully convolutional network while the biases of all convolutional layers are set to 0.

At the beginning of our image noise removal task, the image is transformed into spike sequences, which means a $1 \times H \times W$ tensor is fed into VTSNN and transformed as the size of $T \times H \times W$ via UWE, followed by U-VTSNN. The details of the internal blocks are clearly shown in Figure 3. After all intermediate operations, the last block of U-VTSNN will output spiking sequences. Thus, for decoding, UWD will use the output spike sequences to generate the noise-removed image. Based on our experiments, U-VTSNN is suitable for diverse popular datasets, and its computational efficiency is vastly superior to that of the same ANN architecture (over 274 times).

1 In our work, SNN simulation length $T = n$, which means each bit is represented by a time step.



2.6. Loss function and backpropagation

2.6.1. Rethinking spatio-temporal backpropagation (STBP)

A previous study has applied for training high-performance SNN (Wu et al., 2018; Jin et al., 2022). Noticeably, while examining the stability of a classification task, some researchers applied STBP for image generation (Comşa et al., 2021). The standard backpropagation only considers the spatial information, which can

easily be underfitted and STBP overcomes that shortage. In order to compare STBP with our Independent-Temporal Backpropagation (ITBP) in the noise-removal task, the loss function corresponding to STBP is shown below.

$$\mathcal{L}_{STBP} = \frac{1}{N} \|y - \hat{y}\|_F^2 \quad (9)$$

According to this loss function expression, the process of updating parameters is presented. To fairly compare, we show

how \mathcal{L}_{STBP} updates w_n^j in spatio-temporal domain. Other cases of updating parameters of STBP can be seen in Wu et al. (2018)'s work.

$$\frac{\partial \mathcal{L}_{STBP}}{\partial w_n^j} = \sum_{t=1}^T \frac{\partial \mathcal{L}_{STBP}}{\partial u_{t,n}^i} o_{t,n-1}^j \quad (10)$$

As Equation (12) shown, while STBP updates w_{n+1}^j , $o_{t,n+1}^i$, $o_{t-1,n+1}^i$, and $o_{t,n}^j$ are all connected with w_{n+1}^j . In Figure 4, unlike STBP, error backpropagation of ITBP will not go through the decoder. Hence, ITBP is more efficient than STBP. Because error backpropagation of ITBP is in latent space but representational space for STBP. In other words, there is a risk of overfitting. Since there is a clear difference between ITBP and standard backpropagation: during the training process, ITBP only encodes the labels and does not decode network outputs; while ITBP does not encode the labels and does decode network outputs during the testing process. STBP has overcome standard backpropagation in noise removal task (Comşa et al., 2021). Later in the experiment, ITBP performed even better than STBP in a similar task.

2.6.2. Independent-Temporal Backpropagation (ITBP)

To show the Independent-Temporal Backpropagation (ITBP) training framework, we create the loss function \mathcal{L}_{ITBP} where the weighted mean square error is used as the error index. The expression of it is described below:

$$\mathcal{L}_{ITBP} = \frac{1}{N} \sum_{t=0}^{T-1} 2^t \|o_t - \hat{o}_t\|_F^2 \quad (11)$$

Where N is the number of training examples and $\|\cdot\|_F$ represents the Frobenius norm, T is the total time step and we set $T = 8$ for our UWE and UWD. From the equation above, we regard \mathcal{L}_{ITBP} as a function of w (weight). To obtain the derivative of \mathcal{L}_{ITBP} to w is necessary for the gradient descent. To obtain the final $\frac{\partial \mathcal{L}_{ITBP}}{\partial w_n^j}$, the critical step is to obtain the $\frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n}^i}$ and $\frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i}$ at time t . Now, we show the insight of getting the complete gradient descent. First, from Equations (2) to (4), the output of spiking neurons $o_{t,n+1}^i$ can be represented below:

$$o_{t,n+1}^i = h[u_{t-1,n+1}^i g(o_{t-1,n+1}^i) + \sum_j w_{n+1}^j o_{t,n}^j - V_{th}] \quad (12)$$

Where w_{n+1}^j is the synaptic weight which links the output of $n + 1$ layer spiking neuron $o_{t,n+1}^i$ with the one of n layer $o_{t,n}^j$. According to Equations (2) to (4), we can calculate $\frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n}^i}$ and $\frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i}$ as follows.

$$\frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i} = \frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n}^i} \frac{\partial o_{t,n}^i}{\partial u_{t,n}^i} \quad (13)$$

$$\frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n}^i} = \frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n+1}^j} \frac{\partial o_{t,n+1}^j}{\partial o_{t,n}^i} \quad (14)$$

$$\begin{aligned} \frac{\partial o_{t,n+1}^i}{\partial o_{t,n}^j} &= \frac{\partial o_{t,n+1}^i}{\partial u_{t,n+1}^i} \frac{\partial u_{t,n+1}^i}{\partial o_{t,n}^j} \\ &= \sum_j \frac{\partial o_{t,n+1}^i}{\partial u_{t,n+1}^i} w_{n+1}^j \end{aligned} \quad (15)$$

By Equation (5), the following derivatives of surrogate function Equation (16) can be used for approximation.

$$\frac{\partial o_{t,n+1}^i}{\partial u_{t,n+1}^i} = \frac{1}{1 + (\pi x_{t+1,n}^i)^2} \quad (16)$$

Here, $\frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i}$ is the intermediate variable on the step of updating parameters w_n^j , from Equations (14) to (16), we can solve Equation (13) as follows.

$$\frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i} = \left[\frac{1}{1 + (\pi x_{t+1,n}^i)^2} \right]^2 \frac{\partial \mathcal{L}_{ITBP}}{\partial o_{t,n+1}^j} \sum_j w_{n+1}^j \quad (17)$$

Hence, the way we update parameters will be shown below.

$$\begin{aligned} \frac{\partial \mathcal{L}_{ITBP}}{\partial w_{t,n}^j} &= \frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i} \frac{\partial u_{t,n}^i}{\partial x_{t,n}^i} \frac{\partial x_{t,n}^i}{\partial w_{t,n}^j} \\ &= \frac{\partial \mathcal{L}_{ITBP}}{\partial u_{t,n}^i} o_{t,n-1}^j \end{aligned} \quad (18)$$

To state how we update weights within one epoch clearly, Algorithm 2 is shown. For Independent-Temporal Backpropagation (ITBP) in this paper, it is a non-cross-path backpropagation. That means it only propagates spatially not temporally. In other words, it is a single-modal spatial representation which means single-modality simplifies and enhances ITBP's efficiency. Last but not least, ITBP only propagates spike sequences of coded labels.

3. Results

To demonstrate the superiority of our work and compare it to existing studies fairly, we choose widely used standard datasets for our experiments. Hence, we implemented VTSNN in PyTorch (Paszke et al., 2019), and evaluated it using MNIST (LeCun et al., 1998), F-MNIST (Xiao et al., 2017), and CIFAR10 (Krizhevsky et al., 2009). For MNIST and F-MNIST, we used 60,000 images for training and 10,000 images for evaluation which is the same as Comşa et al. (2021) in the noise-removal task. The input images were resized to 28×28 . To expand the applicability of VTSNN, we also conducted experiments on CIFAR10. For CIFAR10, we used 50,000 images for training and 10,000 images for evaluation. The input images were resized to 32×32 . Moreover, all noisy images used for training and testing contain Gaussian noise at each pixel, with η representing the noise variation in the image scale from 0 to 1. Moreover, our training details are as follows. On NVIDIA GeForce GTX 2080, the models are implemented using PyTorch. In addition, each layer's bias is set to False. The optimizer is Adam Optimizer, which updates the weight parameters of the network with the loss value for better gradient descent, and its

```

Input: Undistorted Weighted Encoding (UWE);
        Undistorted Weighted Decoding (UWD);
        VTSNN model; Simulation length:  $T$ ;
 $n$ -bit input image:  $x$ ;  $n$ -bit label image:  $y$ ;
        Iteration of train numbers:  $I_{train}$ ;
        Iteration of validation numbers:  $I_{val}$ ;
        Initialization;
for  $i = 0, 1, \dots, I_{train}$  iteration do
    for  $t = 0, 1, \dots, T-1$  do
        Apply UWE to encode input  $x$ , label  $y$  to  $r_t$ ,
         $o_t$ .
        Input  $r_t$  to VTSNN model, get spiking
        sequences  $\hat{o}_t$ .
    end
    Calculate loss function:  $\mathcal{L}_{ITBP}(o_t, \hat{o}_t)$  by Equation
    (11)
    Backpropagation and update model parameters by
    Equation (18)
end
for  $i = 0, 1, \dots, I_{val}$  iteration do
    for  $t = 0, 1, \dots, T-1$  do
        Apply UWE to only encode input  $x$  to  $r_t$ , .
        Input  $r_t$  to VTSNN model, get spiking
        sequences  $\hat{o}_t$ .
        Append  $\hat{o}_t$  to  $\hat{O} = \{\hat{o}_0, \hat{o}_1, \dots, \hat{o}_{T-1}\}$ .
    end
    Use UWD to decode  $\hat{O}$  to  $\hat{y}$ ; Calculate PSNR
    between  $y$  and  $\hat{y}$ .
end
    
```

Algorithm 2. ITBP for one epoch.

initial learning rate is set to 0.001. Moreover, our batch size is 50 for both training and testing.

3.1. Comparison with existing works

The performance of two VTSNN variants is compared with some models in Table 1. And a digit from MNIST dataset is reconstructed by our model is show in Figure 5. We train and test two variants based on the PyTorch framework, resulting in enhanced performance across all tasks. And, we compare the performance between ours and the methods proposed by Comşa et al. (2021) which is the only SNN-based image reconstruction attempt yet. On neuromorphically-encoded MNIST, the boost values of PSNR on four various noise levels are {4.26, 5.75, 7.03, 7.06} with only eight time steps. On neuromorphically-encoded F-MNIST, the boost value of PSNR on four various noise levels are {3.66, 4.09, 3.692, 3.93} with also eight time steps. Moreover, the value of PSNR on four various noise levels are {18.27, 14.08, 14.76, 13.16} on neuromorphically-encoded CIFAR10 with also eight time steps, which is quite competitive. Furthermore, our method can achieve higher performance in image reconstruction tasks by neuromorphic encoding/decoding circuits. Even compared with

TABLE 1 Comparison of PSNR on existing works for various noise levels and different datasets (Bold: the best).

Method	MNIST				F-MNIST				CIFAR10			
	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$
SATC-16	17.06	16.60	15.24	14.49	17.54	16.76	16.04	15.61	-	-	-	-
SATC-32	19.11	17.40	16.06	15.10	18.01	17.17	16.72	15.90	-	-	-	-
VTSNN-LIF	22.99	21.74	19.04	19.05	20.85	20.82	19.86	18.71	15.85	10.53	10.96	9.47
VTSNN-IF	23.57	23.15	23.09	22.56	21.67	21.26	20.64	19.83	18.27	14.08	14.76	13.16

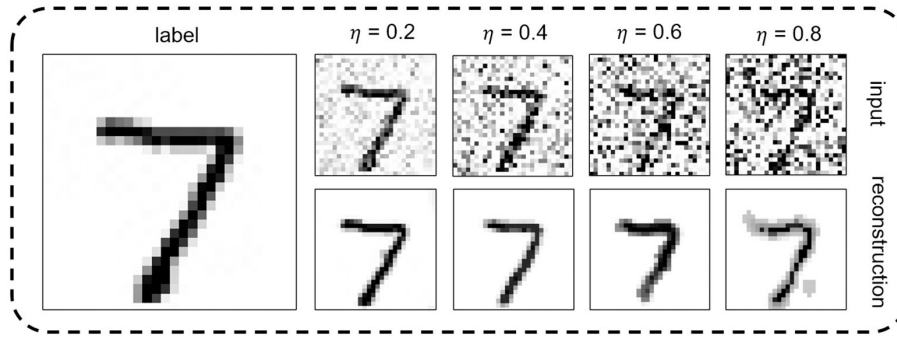


FIGURE 5 A digit from MNIST set is reconstructed by the proposed VTSNN incorporated into the commonly used U-net architecture and IF neuron, at different noise levels.

TABLE 2 Comparison of PSNR on MNIST at various noise level $\eta = 0.2$ for different encoding and decoding (Bold: the best) (Wu et al., 2018; Coşça et al., 2021; Kamata et al., 2022).

Encoding		Decoding		Backpropagation		PSNR
UWE	TTFS	UWD	MPD	STBP	ITBP	
✓		✓		✓		9.60
✓			✓	✓		8.00
	✓	✓		✓		8.54
	✓		✓	✓		9.76
✓		✓			✓	23.57
✓			✓		✓	11.91
	✓	✓			✓	11.81
	✓		✓		✓	20.70
Same architecture ANN						23.05

ANN-based work, on MNIST, at $\eta=0.2$, our VTSNN-IF performs superior to it. And the results are shown in Table 2.

3.2. Ablation study

3.2.1. Comparison between LIF neuron and IF neuron

Currently, research uses leaky-integrate-and-fire (LIF) neurons for SNN, believing its more complex differential equation (Gerstner et al., 2014) can boost performance. Our experiment disproves this bias. To conduct our experiment, we use commonly used datasets (MNIST, FMNIST, and CIFAR10). In our method, the parameters of the IF model are set as $V_{reset} = \text{None}$, $V_{th} = 0.077$ in 1×1 convolution layer (an experience parameter corresponds to best performance), and $V_{th} = 1.0$ in all the other convolution layers. In terms of LIF neurons, $\tau = 1.1$, and all the other parameters are set identically to IF neurons. In the majority of instances, as shown in Table 1, IF neurons usually do better than LIF neurons at this task, regardless of the noise level or dataset.

3.2.2. Comparison among different coding methods

TTFS and MPD are discussed relatively in depth in the rethinking part (Sections 2.3.1 and 2.4.1) and introduction. Since they have been used to generate images (Kamata et al., 2022). They are the two most comparable methods for our UWE and UWD. Table 2 displays all results. UWE is always superior to TTFS when conducting a univariate experiment, and UWD is always superior to MPD too. In addition, the UWE-UWD combination performs exceptionally well for STBP.

3.2.3. Comparison between STBP and ITBP

Experiments demonstrate that ITBP is superior to STBP in terms of the PSNR evaluation metrics. \mathcal{L}_{STBP} and \mathcal{L}_{ITBP} are applied respectively with the same U-VTSNN architecture. Table 2 displays the outcomes of these two backpropagation techniques on the MNIST dataset with $\eta = 0.2$. All these results well proved the superiority of our ITBP.

4. Discussion

4.1. Classification for UWE

In addition to image reconstruction, VTSNN is capable of performing various tasks (Xu et al., 2021; Ran et al., 2022), such as medical detection (Ghosh-Dastidar and Adeli, 2009) and speech recognition (Mansouri-Benssassi and Ye, 2019). As mentioned in the Introduction, classification is a common assignment for SNN. To demonstrate the classification, we employ a VTSNN-based LeNet (VTLLeNet) (LeCun et al., 1998) in which all activation functions are replaced by spiking neurons and UWE is used for encoding. Then, we employ VTLLeNet to classify the MNIST dataset. Furthermore, varying levels of noise ($\eta = 0.2, 0.4, 0.6, 0.8$) are applied to the images in MNIST. Here, we are not attempting to attain optimal outcomes, but rather to test the stability of our UWE classification work. The results are presented as a line chart in Figure 6.

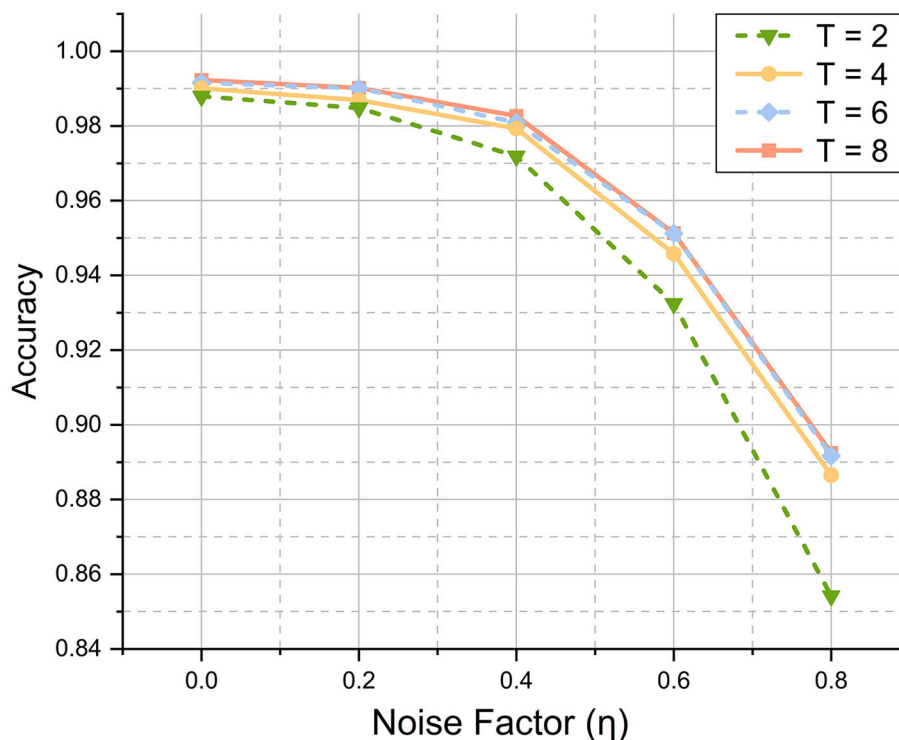


FIGURE 6 Results of classification task in MNIST dataset at different noise factors. For any T , while the noise level goes up the accuracy will decrease. However, even the worst case ($T = 2, \eta = 0.8$) will achieve a quite good result (85.2%). And the best case ($T = 8, \eta = 0.0$) can perform quite competitively (99.2%).

4.2. Energy consumption

In this section, we use the same network structure (Rathi and Roy, 2021; Zhu et al., 2022). Ideally, in the absence of spikes, no computations and active energy are used (Davies et al., 2018; Zhu et al., 2022). For the sake of fairness, we exclude convolutional computations for both and hold the above ideal conjecture. We traverse MNIST and count ANN activation function operations and SNN spikes. In these experiments, all spiking neurons are replaced with an ANN activation function (e.g., ReLU), and its total operations are counted.² ANN then needs 18.39 M Flops³, while VTSNN needs 2.51 M FLOPS. In other words, #OP_{ANN}=18.39 M, #OP_{SNN}=2.51 M.

Following the practice (Zhu et al., 2022), in 45 nm CMOS, each ANN operation consumes 4.6 and 0.9 pJ for each spike (Horowitz, 2014). Thus, 32-bit ANN costs 18.39 M × 4.6 pJ = 8.46 × 10⁻⁵ J, or 273.77 times as much as 32-bit VTSNN. Moreover, details of how energy consumption is calculated can be found in Table 3. This method of calculation is generally accepted in the SNN field and we learned from Zhu et al. (2022). The ideal results are extremely encouraging and demonstrate SNN’s immense potential. To realize

TABLE 3 Comparison of energy based on the counting of operations between ANN and SNN.

	ANN	SNN
Total params	0.12 M	0.12 M
(a) Spike rate	0	0.1366
^a (b) #OP _{ANN}	18.39 M	0
^b (c) #OP _{SNN}	0	2.51 M
^c Energy(10 ⁻⁷ J)	845.94	3.09
^d ANN/SNN Energy	273.77	

^a#OP_{ANN} is the total number of ANN operations if all spiking neurons are replaced with an ANN activation function (e.g., ReLU).

^b#OP_{SNN} = SpikeRate × #OP_{ANN}.

^cEnergy = #OP_{ANN} × 4.6pJ + #OP_{SNN} × 0.9pJ × SpikeRate.

^dEach operation in ANN (SNN) consumes 4.6 pJ (0.9 pJ). ANN/SNN Energy can be calculated by $\frac{(b) \times 4.6}{(a) \times (c) \times 0.9}$.

these awe-inspiring effects, however, future research into hardware is required. The neuromorphic circuits in this paper may be a good harbinger.

4.3. Regularity of threshold voltage

Experiments show V_{th} impacts outputs. To determine the regularity of that relation, we find the optimal V_{th} by attempts. Studies show a doubtful conjecture that increasing V_{th} increases

2 The total number of ANN operations are counted by the torchstat package (Swallow, 2018).

3 In torchstat, the count operations are calculated by the formula (Molchanov et al., 2016) #OP_{ANN} = 2HWC_{in}C_{out}K² where H and W is the output feature map size; C_{in} is input channel; K is kernel size; C_{out} is the output channel.

spiking rate frequency, which improves performance (Niu et al., 2022). At each epoch, we count output spiking rate frequencies corresponding to different V_{th} . Thus, we contradict that simple correlation. Figure 7 depicts the ebb and flow of performance regarding various V_{th} . Recent studies about astrocytes harbor find during daytime and nighttime the threshold for the cell is different (Koronowski and Sassone-Corsi, 2021). This biological property inspired us. Other scholars in the SNN field also state that the dynamic membrane potential threshold, as one of the essential properties of a biological neuron is a spontaneous regulation mechanism that maintains neuronal homeostasis, i.e., the constant overall spiking firing rate of a neuron (Ding et al., 2022). Our discussion is motivated by the above biological research, and we hope to pique the interest of more academics to investigate the regularity of threshold voltage's insight.

4.4. Neuromorphic circuits

To enhance the efficacy of UWE and UWD, a simple neuromorphic circuit can be introduced. The UWE and UWD systems rely fundamentally on a binary encoding-decoding strategy. In particular, binary data is hardware-friendly, inspiring

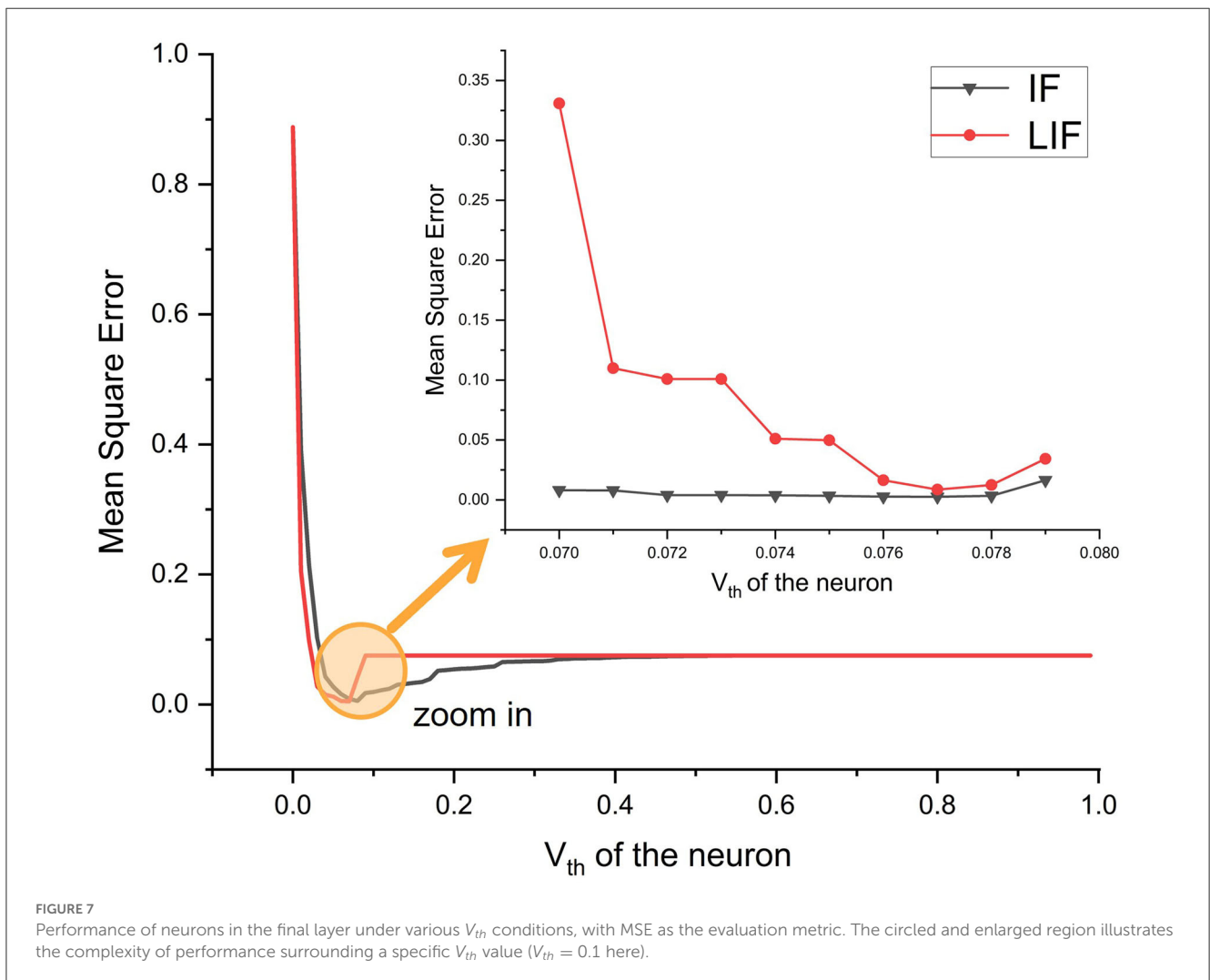
us to investigate ADC and DAC. The non-floating nature of the circuits embodies the spirit of neuromorphic chips and the successful avoidance of calculation through direct electronic responses.

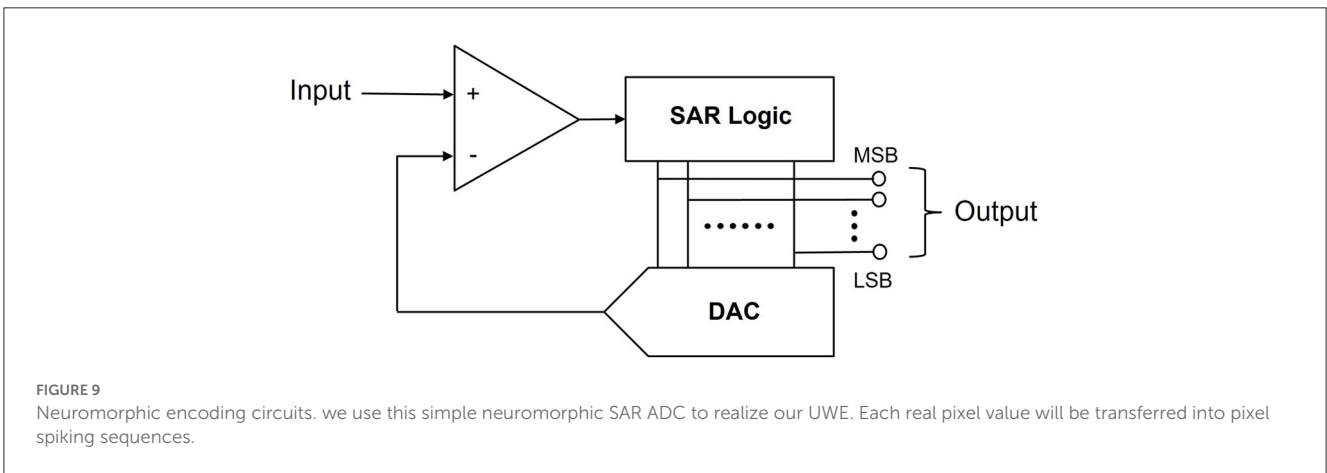
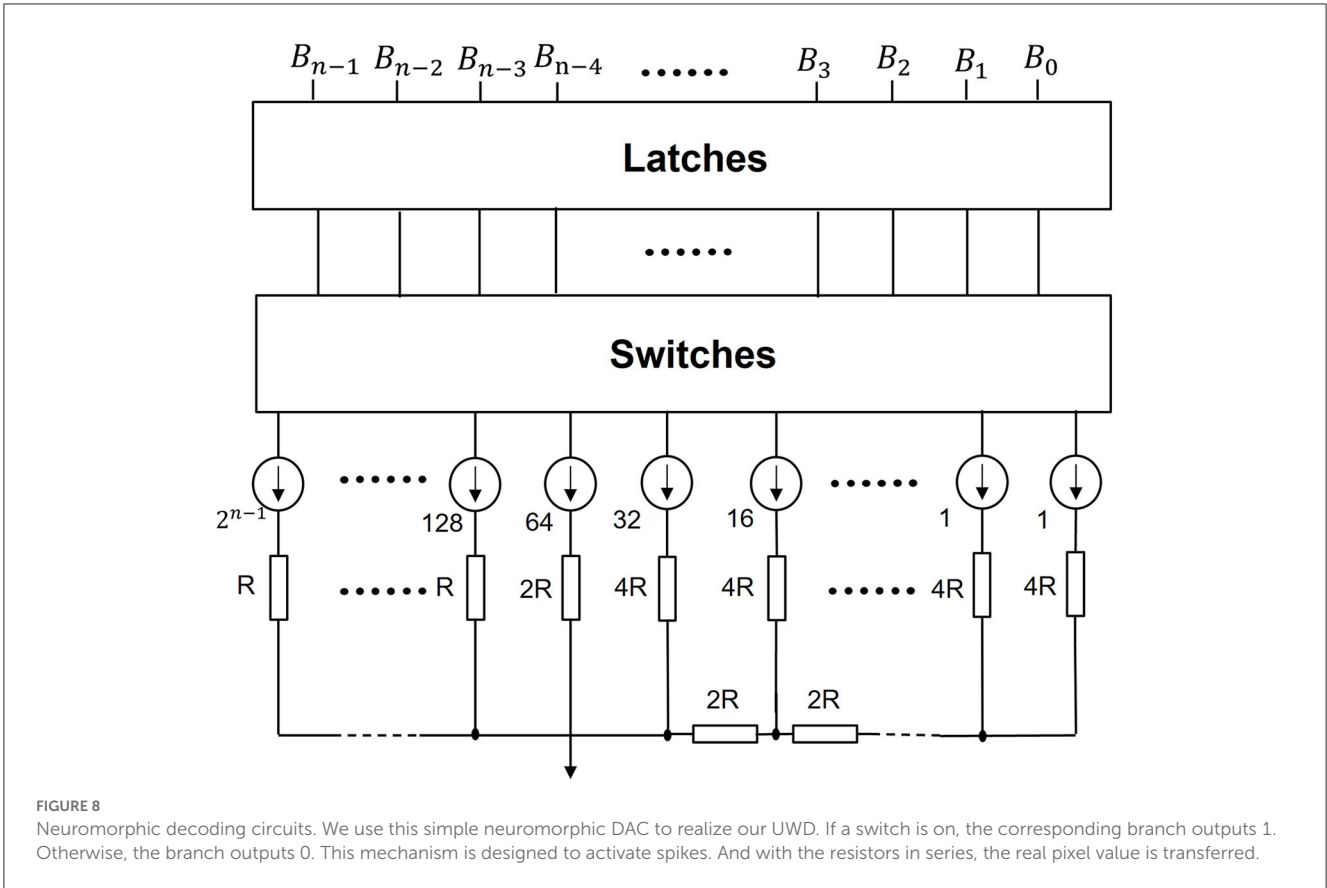
As shown in Figure 8, UWD can be enabled by a simple DAC. Here, $\{B_0, B_1, \dots, B_{n-1}\}$ refers to spiking sequences of a pixel from networks. Whether a spike occurs depends on whether switches are on or off. The output of this neuromorphic chip is the real value of that pixel. Furthermore, the resistance network corresponds to n -bit decoding matrix A in Section 2.4.

Similarly, Figure 9 shows how to realize UWE without sample-hold circuits. A comparator is linked to SAR logic and the DAC model here is the circuits in Figure 8. Finally, MSB is the abbreviation of Most Significant Bit (n -bit) while LSB refers to Least Significant Bit (1-bit). This means MSB to LSB constitutes a binary sequence.

4.5. Limitation

The majority of direct training SNNs are currently trained with rather basic data. In addition, all of the current SNN-based image reconstruction research use very simple images (Comşa et al.,





2021; Kamata et al., 2022). Similarly, our work here is unable to circumvent this difficulty. The reconstructed high-resolution images created by VTSNN are not optimal and seem blurry to the human eyes. In conclusion, SNN is still far behind ANN in image reconstruction tasks involving high-resolution images. However, SNN’s potential cannot be ignored.

5. Conclusions

We have developed a novel spiking neuron network called VTSNN, where we adopt SNN with a virtual temporal dimension

and a new backpropagation method. Besides, we raise Undistorted Weighted-Encoding to transfer the image into spiking information, which can be easily realized by a neuromorphic circuit to improve efficiency, as well as the symmetric process of Undistorted Weighted-Decoding. The experiments proved that VTSNN sometimes performs similarly to or better than ANN, for the same architecture and VTSNN is superior to all other comparable SNN models. Future research should focus on the development of hardware and the applicability of high-resolution images. There remain some constraints. The relationship between image low-level task performance and V_{th} is unclear. The proposed encoding-decoding circuits are not yet constructed physically.

Data availability statement

The study's original contributions are given in the publication. And our code is available at this <https://github.com/bollossom/VTS>NN%20>. For more information, please contact the relevant authors.

Author contributions

X-RQ, Z-RW, and ZL designed and did the experiments, wrote the code, wrote the first draft of the manuscript, and contributed equally. R-JZ provided consultation on SNN knowledge, optimized the code, and helped with literature research. XW polished the draft manuscript and reviewed the code. M-LZ contributed to the concept and design. L-JD directed the projects and provided overall guidance. All authors contributed to the article and approved the submitted version.

Funding

This research was supported by NSFC (12271083 and 12171072), Natural Science Foundation of Sichuan Province (2022NSFSC0501).

References

- Bohte, S. M., Kok, J. N., and La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48, 17–37. doi: 10.1016/S0925-2312(01)00658-0
- Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biol. Cybern.* 95, 1–19. doi: 10.1007/s00422-006-0068-6
- Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., and Liu, S. (2021). "Nbnnet: noise basis learning for image denoising with subspace projection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4896–4906. doi: 10.1109/CVPR46437.2021.00486
- Comşa, I. M., Versari, L., Fischbacher, T., and Alakuijala, J. (2021). Spiking autoencoders with temporal coding. *Front. Neurosci.* 15, 936. doi: 10.3389/fnins.2021.712667
- Davies, M., Srinivasan, N., Lin, T.-H., China, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359
- Deng, S., Li, Y., Zhang, S., and Gu, S. (2022). Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv*, 2202.11946.
- Ding, J., Dong, B., Heide, F., Ding, Y., Zhou, Y., Yin, B., et al. (2022). Biologically inspired dynamic thresholds for spiking neural networks. *arXiv preprint arXiv*, 2206.04426.
- Eshraghian, J. K., Ward, M., Nefci, E., Wang, X., Lenz, G., Dwivedi, G., et al. (2021). Training spiking neural networks using lessons from deep learning. *arXiv*, 2109.12894.
- Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021). "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC: IEEE), 2661–2671. doi: 10.1109/ICCV48922.2021.00266
- Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press.
- Ghosh-Dastidar, S. and Adeli, H. (2009). A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Netw.* 22, 1419–1431. doi: 10.1016/j.neunet.2009.04.003
- Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500. doi: 10.1113/jphysiol.1952.sp004764
- Horowitz, M. (2014). "1.1 Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (San Francisco, CA: IEEE), 10–14. doi: 10.1109/ISSCC.2014.6757323
- Hubel, D. H., and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106. doi: 10.1113/jphysiol.1962.sp006837
- Jin, C., Zhu, R. J., Wu, X., and Deng, L. J. (2022). Sit: a bionic and non-linear neuron for spiking neural network. *arXiv*, 2203.16117.
- Kamata, H., Mukuta, Y., and Harada, T. (2022). "Fully spiking variational autoencoder," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Vancouver, BC: AAAI Press), 7059–7067. doi: 10.1609/aaai.v36i6.20665
- Kim, J., Kim, H., Huh, S., Lee, J., and Choi, K. (2018). Deep neural networks with weighted spikes. *Neurocomputing* 311, 373–386. doi: 10.1016/j.neucom.2018.05.087
- Kim, Y., Park, H., Moitra, A., Bhattacharjee, A., Venkatesha, Y., and Panda, P. (2022). "Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks?" in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Toronto, ON: IEEE), 71–75. doi: 10.1109/ICASSP43922.2022.9747906
- Kolda, T. G. (2006). *Multilinear Operators for Higher-Order Decompositions*. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM; Livermore, CA. doi: 10.2172/923081
- Koronowski, K. B., and Sassone-Corsi, P. (2021). Communicating clocks shape circadian homeostasis. *Science* 371, eabd0951. doi: 10.1126/science.abd0951
- Krizhevsky, A., (2009). Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, Toronto, ON, Canada.
- Lapique, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol.* 9, 620–635.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Acknowledgments

Prof. Hong-Zhi Zhao, National Key Laboratory of Science and Technology on Communications, UESTC, provided valuable feedback about binary encoding, for which we are grateful. Mr. Jia-Le Yü, College of Architecture and Urban Planning, Tongji University, assisted with the creation of figures. Mr. Bin Kang from the National Exemplary School of Microelectronics, UESTC, gave feedback about circuits.

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.

- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7
- Mansouri-Bensassani, E., and Ye, J. (2019). “Speech emotion recognition with early visual cross-modal enhancement using spiking neural networks,” in *2019 International Joint Conference on Neural Networks (IJCNN)* (Budapest), 1–8. doi: 10.1109/IJCNN.2019.8852473
- Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. (2016). Pruning convolutional neural networks for resource efficient inference. *arXiv: 1611.06440*.
- 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
- Niu, L.-Y., Wei, Y., Long, J.-Y., and Liu, W.-B. (2022). High-accuracy spiking neural network for objective recognition based on proportional attenuating neuron. *Neural Process. Lett.* 54, 1055–1073. doi: 10.1007/s11063-021-10669-6
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). “Pytorch: an imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems (NeurIPS)* (Vancouver, BC), 32.
- Ran, X., Xu, M., Mei, L., Xu, Q., and Liu, Q. (2022). Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation. *Neural Netw.* 145, 199–208. doi: 10.1016/j.neunet.2021.10.020
- Rathi, N., and Roy, K. (2021). “Diet-SNN: a low-latency spiking neural network with direct input encoding and leakage and threshold optimization,” in *IEEE Transactions on Neural Networks and Learning Systems* (Glasgow: IEEE), 1–9. doi: 10.1109/TNNLS.2021.3111897
- Ronneberger, O., Fischer, P., and Brox, T. (2015). “U-net: convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)* (Munich), 234–241. doi: 10.1007/978-3-319-24574-4_28
- Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2
- Saygin, A. P. (2007). Superior temporal and premotor brain areas necessary for biological motion perception. *Brain* 130, 2452–2461. doi: 10.1093/brain/awm162
- Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., and Benosman, R. (2018). “Hats: histograms of averaged time surfaces for robust event-based object classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT: IEEE), 1731–1740. doi: 10.1109/CVPR.2018.00186
- Swallow, A. (2018). *torchstat*. GitHub. Available online at: <https://github.com/Swallow/torchstat.git> (accessed July 7, 2022).
- Thorpe, S., Delorme, A., and Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Netw.* 14, 715–725. doi: 10.1016/S0893-6080(01)00083-1
- Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., and Li, H. (2022). “Uformer: a general u-shaped transformer for image restoration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Orlando, FL: IEEE), 17683–17693. doi: 10.1109/CVPR52688.2022.01716
- Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 1550–1560. doi: 10.1109/5.58337
- Wu, Y., Lei, D., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12, 331. doi: 10.3389/fnins.2018.00331
- Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv: 1708.07747*.
- Xing, Y., Di Caterina, G., and Soraghan, J. (2020). A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Front. Neurosci.* 14, 1143. doi: 10.3389/fnins.2020.590164
- Xu, Q., Shen, J., Ran, X., Tang, H., Pan, G., and Liu, J. K. (2021). Robust transcoding sensory information with neural spikes. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 1935–1946. doi: 10.1109/TNNLS.2021.3107449
- Yue, Z., Zhao, Q., Zhang, L., and Meng, D. (2020). “Dual adversarial network: toward real-world noise removal and noise generation,” in *European Conference on Computer Vision* (Springer), 41–58. doi: 10.1007/978-3-030-58607-2_3
- Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., et al. (2021). “Multi-stage progressive image restoration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Montreal, QC: IEEE), 14821–14831. doi: 10.1109/CVPR46437.2021.01458
- Zhang, W., and Li, P. (2020). Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Adv. Neural Inform. Process. Syst.* 33, 12022–12033.
- Zhang, X., Liao, W., Yu, L., Yang, W., and Xia, G.-S. (2021). “Event-based synthetic aperture imaging with a hybrid network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Montreal, QC: IEEE), 14235–14244. doi: 10.1109/CVPR46437.2021.01401
- Zheng, H., Wu, Y., Deng, L., Hu, Y., and Li, G. (2021). “Going deeper with directly-trained larger spiking neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence* (Vancouver, BC: AAAI Press), 11062–11070. doi: 10.1609/aaai.v35i12.17320
- Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., and Tian, Y. (2022). “Event-based video reconstruction via potential-assisted spiking neural network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Orlando, FL: IEEE), 3594–3604. doi: 10.1109/CVPR52688.2022.00358