Perturbation of Spike Timing Benefits Neural Network Performance on Similarity Search

Ziru Wang, Jiawen Liu, Yongqiang Ma, *Student Member, IEEE*, Badong Chen, *Senior Member, IEEE*, Nanning Zheng, *Fellow, IEEE*, and Pengju Ren, *Member, IEEE*

***Abstract*— Perturbation has a positive effect, as it contributes to the stability of neural systems through adaptation and robustness. For example, deep reinforcement learning generally engages in exploratory behavior by injecting noise into the action space and network parameters. It can consistently increase the agent's exploration ability and lead to richer sets of behaviors. Evolutionary strategies also apply parameter perturbations, which makes network architecture robust and diverse. Our main concern is whether the notion of synaptic perturbation introduced in a spiking neural network (SNN) is biologically relevant or if novel frameworks and components are desired to account for the perturbation properties of artificial neural systems. In this work, we first review part of the locality-sensitive hashing (LSH) of similarity search, the FLY algorithm, as recently published in *Science*, and propose an improved architecture, time-shifted spiking LSH (TS-SLSH), with the consideration of temporal perturbations of the firing moments of spike pulses. Experiment results show promising performance of the proposed method and demonstrate its generality to various spiking neuron models. Therefore, we expect temporal perturbation to play an active role in SNN performance.**

***Index Terms*— Locality sensitive hashing (LSH), spiking neural network (SNN), temporal perturbation.**

I. Introduction

IN THE biological context, molecular noise is central in all intracellular processes due to inevitable Brownian motion of molecules and the inherent thermal stochasticity of molecular events, such as binding, chemical reactions, or conformational transitions. The amplitudes and exact firing moments of the action potential—spikes of the neural network—are ascribed to the randomness of various factors, such as a neuron's synaptic response current, membrane voltage, and refractory delay [1]. Therefore, the internal disturbance and constraints of a biological environment cause a variance of the firing time of spikes. Numerical studies have shown that the emission time is chaotic in the sense that it is nonrepetitive and sensitive to small perturbations [2], [3]. Therefore, from the perspective of computational neuroscience, perturbations are regarded as the main challenge to the precise emulation of the behavior of neuron spikes.

In a system context, resilience refers to the possibility that, after an unpredictable perturbation, a system can return to its original behavior if the deformation does not exceed some bound. It is obvious that the randomness of neuron activity is constrained, as the membrane, integration, and regulation by and within a neuron all contribute to restricting and canalizing randomness. Nevertheless, from a system perspective, the inherent disturbance improves the adaptability and robustness of a neural system.

As a consequence, biological neuron systems exhibit an impressive range of properties and behaviors that we currently struggle to model with analytical tools, whether at the microlevel or macrolevel. Stochastic threshold, random reset (stochastic channel opening and closing), and noisy integration (inputs and thermal noise) can all be reflected in the release time of spikes. In this work, a mathematical relation is established by considering the perturbation of the release time of spikes as random drift. Notice that this is a purely mathematical notion; it may not be exactly consistent with physical measurement, yet it applies to the spiking neural network (SNN) architecture with the introduction of the time-shifter unit. We demonstrate that the classification function of an SNN is robust, and the classification accuracy is beneficial to the temporal perturbations of spikes. According to our experiments, for perturbations subject to a Gaussian distribution, the variance has little effect on performance compared to the mean value. We view these as important results in light of ongoing research about the introduction of spike perturbation as a necessary component in SNN architecture.

To be specific, in a recent article in *Science* [4], researchers studied the fruit fly olfactory circuit and uncovered the computational strategies of approximate similarity (or nearest-neighbors) search of high-dimensional points using locality-sensitive hashing (LSH) [5]–[7]. They described their FLY algorithm as treating a fly's circuit as a hash function whose input is an odor, and whose output is a tag (called a hash) for the specific odor. Inspired by this work, we integrate the SNN model with LSH and improve the system architecture in which we call time-shifted spiking LSH (TS-SLSH), with the consideration of perturbations. In particular, we demonstrate how temporal perturbation, an essential component of biological information processing, can greatly promote the classification accuracy and robustness of an SNN.

Manuscript received 8 January 2019; revised 6 November 2019 and 11 July 2020; accepted 26 January 2021. Date of publication 19 February 2021; date of current version 2 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61773307 and Grant 62088102. *(Corresponding author: Pengju Ren.)*

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: pengjuren@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3056694.

Digital Object Identifier 10.1109/TNNLS.2021.3056694

We tested the performance of TS-SLSH on three graphic data sets: Mixed National Institute of Standards and Technology database (MNIST), street view house number dataset (SVHN), and scale invariant feature transform (SIFT). Experiments show the proposed algorithm provides a viable means to discover similar inputs and achieves better results and robustness than previous research.

TS-SLSH consists of a temporal encoding module followed by a random connection layer that involves a spatiotemporal spike pattern, which holds biological evidence. Similar to a support vector machine (SVM), which maps the original data to a higher dimensional feature space to solve the nonlinear classification, TS-SLSH introduces temporal features, so it potentially has stronger expression ability. We mainly focus on the following four aspects.

1) Our phase encoding method applies a nonlinear function to expand the discrimination of the intermediate gray value of pixels at the same position of the image, which helps to distinguish categories.
2) The synaptic time-shifter is applied to an individual spike before it is superimposed into spike trains to reflect the disturbance of its release time.
3) Random connection between an afferent neuron and output neuron is applied, randomly directing input into individual neurons to provide a mechanism that can produce a richer response.
4) We conducted experiments in multiple dimensions, including various neuron models, hash lengths, and connection ratios, as well as different means and variances of perturbations obeying Gaussian distributions, to faithfully study our proposed TS-SLSH.

The rest of this article is organized as follows. Related work is introduced in Section II, and the system architecture is described in Section III. We describe our experimental analysis reflecting the ability of TS-SLSH in Section IV. The hardware implementation of SNN is discussed in Section V. In Section VI, we provide our conclusions and analysis.

II. Related Work

The primary task in neural circuits is to produce neural patterns in response to external stimuli, which can help identify similar inputs through nearest-neighbors searching. It plays a fundamentally important role in data compression and information retrieval [5]. Information conveyance in neural systems is studied thoroughly in this article, with a focus on temporal perturbation of spikes for nearest-neighbors searching.

Several algorithms have been proposed recently to address the approximate similarity search in a given feature space. LSH [5], [6] is one of the most frequently used approaches in this field. Different from a traditional (non-LSH) hash function, LSH keeps a good balance of input distance between the original k-dimensional space (k is the number of pixels of an input image) and h-dimensional space (h represents the hash length) so that its inputs are not scattered stochastically. The advantage of the LSH function lies in the two adjacent inputs in the original space after transformation. LSH's local sensitivity is achieved by some specific hash functions. The study of nearest-neighbors search commonly requires the distance in the original space between two points to conform to that in the feature space. It is worth mentioning that LSH does not necessarily guarantee to find the input's nearest neighbors, but it greatly reduces the dimensionality of high-dimensional data so that similar items map to the same "buckets" with high probability. The schematic of LSH is shown in Fig. 1(a), where h output neurons are fully connected to the k input neurons (units).

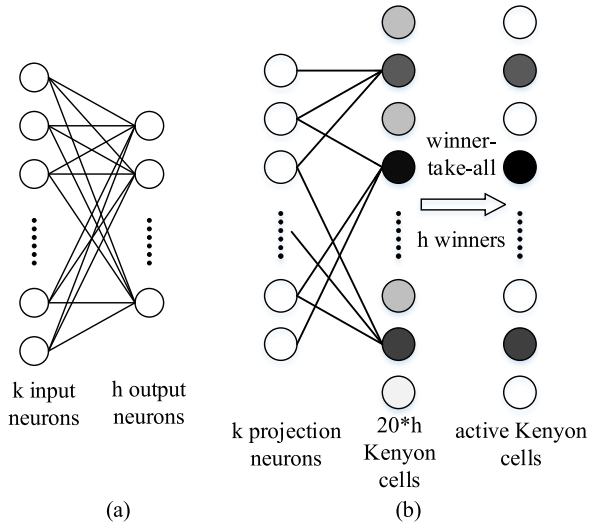


Fig. 1. Structure of traditional LSH and FLY algorithms. FLY considers a fly's olfactory system as a hash function and views the input and output as the odor and hash keys, respectively. (a) In the structure of traditional LSH, k input neurons are fully connected to h output neurons. (b) In the FLY structure, k PNs are sparsely connected to $20 \times h$ KCs with binary weights. Then, h active KCs are generated by the WTA rule [4].

A recent article in *Science* presented a neural algorithm, FLY, to improve the LSH model [4]. It mimics a fly's olfactory system and views the olfactory circuit as the hash function, where the input is the corresponding odor and the output is a hash key. The FLY and LSH algorithms differ in two specific ways: 1) LSH reduces the number of neurons, whereas FLY expands it and 2) LSH preserves its output, while FLY selects a winner output by the winner-take-all (WTA) rule. The structure of FLY is presented in Fig. 1(b). Projection neurons (PNs) are connected to Kenyon cells (KCs) represented by a random and binary projection matrix. Winner neurons are selected among KCs, and the h-dimensional outputs of the winners represent the feature of the k-dimensional input. Though this projection matrix is sparse, it still makes good use of inputs' information as it increases the network's dimension. However, both ignore the real neural processing in biological systems and lack bionic plausibility. For example, although FLY is inspired by a fly's olfactory system, the way by which information is transferred among neurons is still purely numeric calculation, which may lead to the loss of some biological information. Therefore, we want to combine the advantages of SNNs and FLY.

SNNs [8]–[10] have achieved excellent results. They are more promising and biologically feasible than traditional artificial neural network SNNs [11], as they increase biological realism by introducing spikes, and they incorporate time as an explicit dependence in their computations to manipulate information in the temporal domain for processing. Therefore,

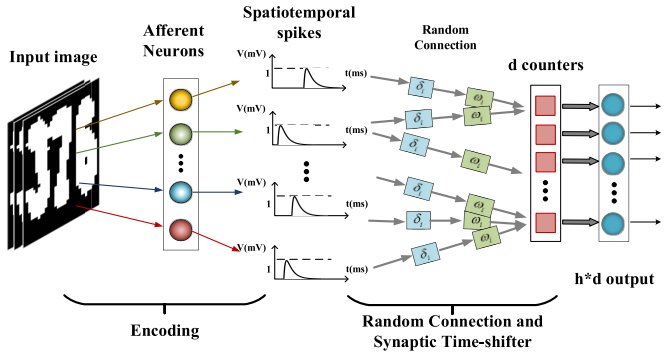


Fig. 2. Diagram of the TS-SLSH algorithm, consisting mainly of encoding, random connection, and time-shifting stages.

spikes are always considered as a potential approach in information processing in the neural system, as they can carry more information than scalars by changing fire times and amplitudes. With the encoding of time information in spiked neurons and hierarchical network architecture, SNNs may become suitable to process relevant spatiotemporal information.

We previously proposed spiking LSH (SLSH) [12] to combine the advantages of LSH and SNN. Although spiking neurons are applied in searching the nearest neighbors, problems, such as a heavy calculation burden, require further exploration.

Converting stimuli, such as sounds and images to spike signals, is generally the first and necessary step, and the utilization of encoding mechanisms is vital. The many encoding methods can be classified as either rate [13] or temporal encoding [14]. The rate encoding method is seldom used, as it transforms the stimulus to the rate of generating spikes rather than specific firing times, which has poor expression ability [12], [15], [16]. The temporal encoding method is more popular because it can convey more useful information. The observation that stimuli are transferred to spikes in the retina [17] and visual cortices [18], [19] on a millisecond timescale supports temporal encoding. Therefore, we adopt the temporal encoding method and apply the same timescale.

III. System Architecture

Fig. 2 shows the schematic of TS-SLSH, which has two stages: phase encoding that encodes external stimuli into spikes, and random connection (projection) with time-shifting. Each pixel of the input image is connected to a corresponding neuron to encode it into a precise firing time. Following encoding, a synaptic time-shifting unit introduces the notion of perturbation. Encoded neurons are randomly connected to output neurons as the biologically synaptic connections, as the research [4], [20] has shown that KCs of the fruit fly integrate inputs from glomeruli in random ways.

1) *Encoding:* Temporal encoding is applied in TS-SLSH. The encoding part of the network has k units, which is the number of pixels in the input image, which can be transferred to k encoding neurons with pixels composed of grayscale values, which are converted to decimal values to express their corresponding units' firing times within the encoding time window [21]. The encoding window of size $(3T/2)$ will be described shortly. This part has two main steps. The first is to convert the input pixels to potential oscillations. The second step involves a modulus to ensure that all firing times are in the positive time domain.
2) *Random Connection and Synaptic Time-Shifting:* This part consists of h output neurons randomly connected to k encoding afferent neurons. The connection can be represented by a sparse, random projection matrix. Time perturbations are added to each spike by a time-shifter. The various spikes are combined into a sequence, called the spike-train.

In the following, we describe the encoding method, the neuron models, random projection matrix, time-shifter, and similarity measurement of time sequences.

A. Encoding Method

Encoding plays a critical role in neural network processing. It aims to generate discrete spike events in a temporal domain. Various encoding methods can be used to convert external visual information into spike trains. The encoding method affects the amount and content of information transmitted by spikes. Action potentials in the human brain are said to be related to the phases of subthreshold membrane potential oscillations [13], [22]. Therefore, we apply phase encoding and improve it to maximize the differences between corresponding pixels among images. We convert each image to k single spikes (firing times) to convert the precise pixel values to several precise action potentials [23], [24]. The process of phase encoding is shown in Fig. 3(b).

The encoding phase has two steps.

Step 1: First is to convert an input image to potential oscillations to extract the features of external stimuli. Image pixels are expanded by rows; the oscillation of the ith pixel is described as

$$\mathrm{OSC}_i = A\cos(\omega t + \phi_i) + B \quad t + \frac{\phi_i}{\omega} \in (0, T) \tag{1}$$

where A is the amplitude of the subthreshold membrane, and B is the bias of the oscillation. Usually, $B = 0$ in the state of rest. ω is the angular velocity, and ϕ_i is the phase shift of the ith pixel, defined here as

$$\phi_i = \phi_0 + (i-1)\cdot\Delta\phi \tag{2}$$

where ϕ_0 is the initial phase shift, and $\Delta\phi = (2\pi/k)$ is the constant phase difference between adjacent pixels, where k is the number of pixels in an image, equal to the number of encoding neurons. We set the oscillation period as $T = 300$ ms.

All pixels are now transformed to membrane oscillations, as described in Fig. 3(a). If a pixel is white, its corresponding oscillation will shift left by $\pi/2$, and a black pixel will cause a right shift of $\pi/2$. Gray pixels will generate a value between $\pi/2$ and $-\pi/2$ according to the applied nonlinear curve. When these oscillations exceed the thresholds $W_{\mathrm{thr}}/2$ or $B_{\mathrm{thr}}/2$, for white and black pixels, respectively, homologous spikes are generated.

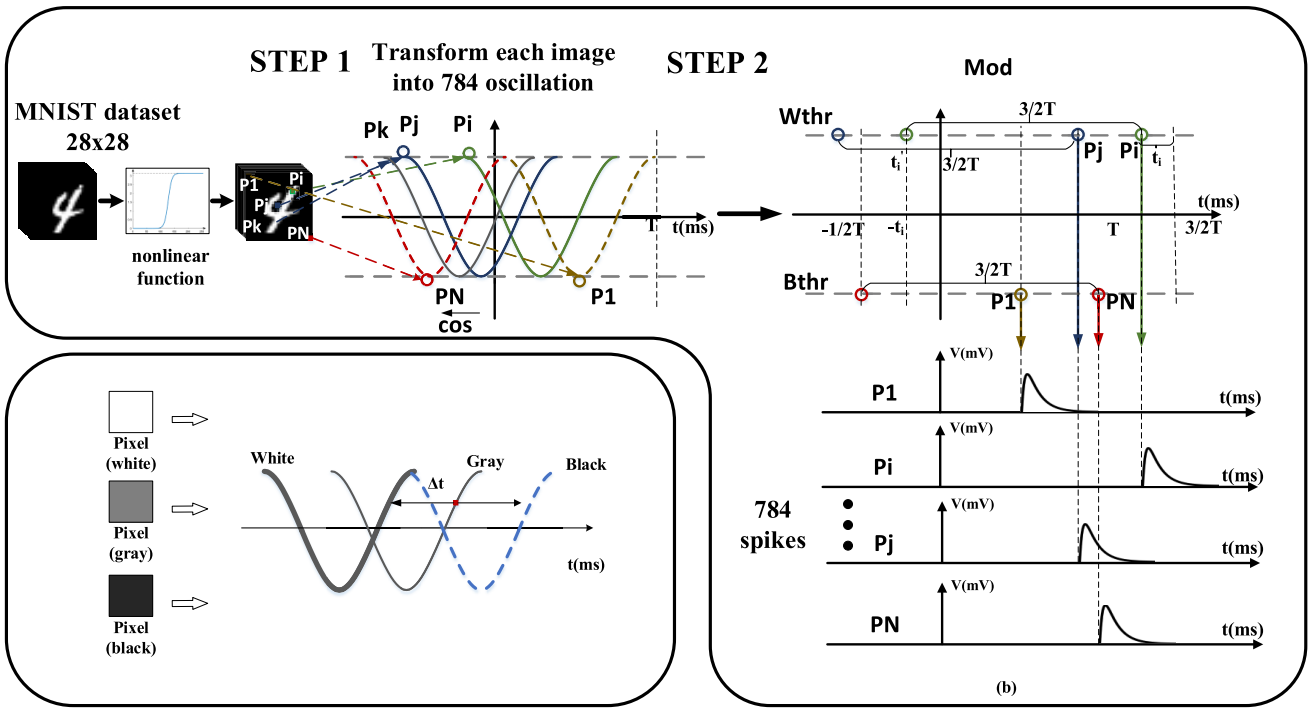


Fig. 3. Structure of the encoding method. For the MNIST data set, the 28×28 input image is encoded into k single spikes triggered at different times. (a) Phase adjustment according to the gray value of the pixel. (b) Spike-encoding procedure.

Step 2: After the first step, some firing times are negative, such as P_i, P_j, and P_N in Fig. 3(b). We use a "mod" operation to guarantee that the firing times of all these spikes are positive and fall in the time window $[0, 3/2T]$ (as proved later). After adjustments, pixels of the input image are encoded with various firing times. The second step can be described as

$$t_i = \operatorname{mod}\left(t_i', \beta\right) \tag{3}$$

where β is the time window, t_i' is the initial firing time of the ith pixel in the image, and t_i is the firing time after adjustment. The process is shown in Fig. 3(b). The relationship between the time window and oscillation period is

$$t_i \in \begin{cases} \left[0, \dfrac{T}{2}\right] & t_i' \in \left[0, \dfrac{T}{2}\right] \\ (\beta - T, \beta) & t_i' \in (-T, 0). \end{cases} \tag{4}$$

We set $\beta \geq T$ because we expect that $t_i \geq 0$.

To prevent the phenomenon that no spike is fired within a certain time interval, it must be true that

$$\beta - T \leq \frac{T}{2} \implies \beta \leq \frac{3T}{2}. \tag{5}$$

A crucial purpose of encoding is to distinguish different images as much as possible. The distance between white and black pixels at the same position among different images should be as large as possible. It can be described as

$$\hat{\beta} = \arg\max_{\beta} J\left(\beta, {t_i'}^B, {t_i'}^W\right) \tag{6}$$

with

$$J\left(\beta, {t_i'}^B, {t_i'}^W\right) = \left\| {t_i}^B - {t_i}^W \right\| \tag{7}$$

where $J(\beta, {t_i'}^B, {t_i'}^W)$ is the distance between the ith black and white pixels, and ${t_i'}^B$ and ${t_i'}^W$ are their respective initial firing times. Considering (3), (6) can be expressed as

$$\widehat{\beta} = \arg\max_{\beta} \left\| \operatorname{mod}\left(t_i'^B, \beta\right) - \operatorname{mod}\left(t_i'^W, \beta\right) \right\| \tag{8}$$

$$\text{subject to} \quad \max J\left(\beta, t_i'^B, t_i'^W\right) = \max \left\| t_i{}^B - t_i{}^W \right\| = \begin{cases} \left| t_i'^B - t_i'^W \right| & t_i'^B \times t_i'^W > 0 \\ \left| t_i{}'^B - \left(\beta + t_i'^W\right) \right| & \text{otherwise.} \end{cases} \tag{9}$$

Considering (4), we can find that $\widehat{\beta} = (3T/2)$. Therefore, we set the time windows to $[0, 3/2T]$.

B. Neuron Models

Neuromorphic research spans a range of neuron models with different levels of abstraction. By using precise firing moments, neurons show advantages in terms of time accuracy. There are several neuron models, as follows.

1) Leaky Integrate-and-Fire (LIF) Model: LIF is most widely used because of its simple and effective computation [25]–[32], as follows:

$$\tau_m \frac{dV_m}{dt} = -V_m + E + I_{\text{syn}} R_m \tag{10}$$

where V_m is the membrane potential and τ_m is the constant membrane time. $\tau_m = R_m C_m$ indicates the membrane time depending on membrane resistance R_m and capacitance C_m of the neuron. E is the neuron's resting potential. If V_m reaches the membrane threshold V_{th}, then the neuron will fire a spike, and the membrane potential V_m will be reset to V_{reset}.

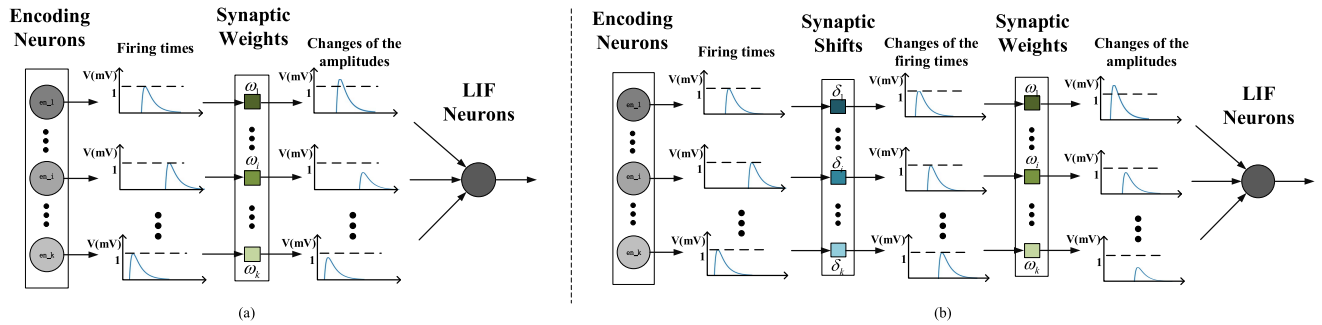


Fig. 4. Information transfer between neurons. (a) Traditional information transfer in SNN. (b) Proposed processing procedure.

The LIF model in our article is driven by exponential decaying synaptic currents generated by all the inputs from the incoming neurons. To fully utilize the temporal characteristics of SNN, we introduce the time-shifter before the weighted sum of the exponential decaying synaptic currents, whose process is described in Section III-C.

We test the flexibility and versatility of the TS-SLSH model by using different neuron models, such as Izhikevich neuron model (IM) and Fitzhugh-Nagumo (FN) model.

2) IM Model: The dynamic of the IM model is described as

$$\begin{cases} \dfrac{dV_m}{dt} = 0.04V_m^2 + 5V_m + 140 - U + I_{\text{syn}} \\ \dfrac{dU}{dt} = a(bV_m - U) \end{cases} \tag{11}$$

where V_m and I_{syn} are the membrane potential and synaptic currents, respectively, and U is the membrane recovery variable. Here, $a = 0.02$ and $b = 0.2$ are the empirical values exhibiting the spiking behavior. When V_m exceeds the threshold 30 mV, the neuron fires and V_m and U are reset to -65 and $+8$ mV, respectively.

3) FN Model: The dynamic of the FN model is described as

$$\begin{cases} \dfrac{dV_m}{dt} = V_m - \dfrac{{V_m}^3}{3} - U + I_{\text{syn}} \\ \dfrac{dU}{dt} = e(V_m + c - dU) \end{cases} \tag{12}$$

where V_m, I_{syn}, and U are described above. The parameters $c = 0.7$, $d = 0.8$, and $e = (1/9)$ are empirical values chosen to express the dynamics of the FN neuron. We study the performance impact of the proposed method on different neural models in Section IV-C.

To ensure computational efficiency, we use LIF in our experiments, unless otherwise indicated.

C. Random Connection and Synaptic Time-Shifting

Inspired by the fly's olfactory circuit and the FLY algorithm [4], [20], we randomly connect output neurons with afferent neurons to convert an input image to a spike train. This enables us to maximize the diversity of features conveyed to output neurons because a given stimulus can activate different patterns in output neurons.

The random connections in this network consist of h output neurons randomly connected to the k encoding neurons (afferent neurons). The connections between layers can be represented by a sparse, random projection matrix. The total number of synaptic weights is $\eta \times k \times h$, where η is the connection ratio, which represents the percentage of element "1" in the projection matrix. We discuss the performance influence of different values of η in Section IV-F. In the following experiments, we set $\eta = 50\%$, unless otherwise indicated.

The more the output neurons, the more complex the network. To be fair, all experiments were based on the same hash length (number of output neurons). Encoding neurons convert the input to k spikes firing at different times. Output neurons receive signals from encoding neurons through synapses and then generate a spike train within the time window. The comparison among input images is based on the average distance of the spiking trains.

In previous research, the preceding neuron fired spikes, which passed into the next neuron with the influence of synaptic efficacy as reflected in different amplitudes. When the membrane voltage crossed a firing threshold value, the neuron generated action potentials (or spikes). The process is shown in Fig. 4(a).

In this article, apart from the changes of the amplitudes of spikes caused by the impact of synaptic weights, we consider the perturbations of spikes by adding a time-shifter (changing the firing times) in the process of transferring information between neurons. The process is shown in Fig. 4(b). Encoding neurons generate spiking patterns in response to external stimuli. Every encoding neuron fires only once in the encoding window. The firing time of each spike drifts according to the impact of the synaptic time-shifter. Spiking neurons then receive all incoming spikes, and their membrane potential is the weighted sum of postsynaptic potentials after shifting. If a neuron's potential reaches the threshold, then the neuron will issue a spike at that time. The process is shown as

$$I_{\text{syn}}(t) = \sum_i w_i I^i{}_{\text{post}}(t - \xi_i) \tag{13}$$

where I_{syn} is the input synaptic current of spiking neurons and w_i is the synaptic efficacy. The larger w_i is the stronger the connectivity between neurons will be. The weights of these synapses obey a normal distribution. It is worth noting that the synaptic efficacy should not be less than zero because the spiking neurons always respond after the stimulus. Therefore,

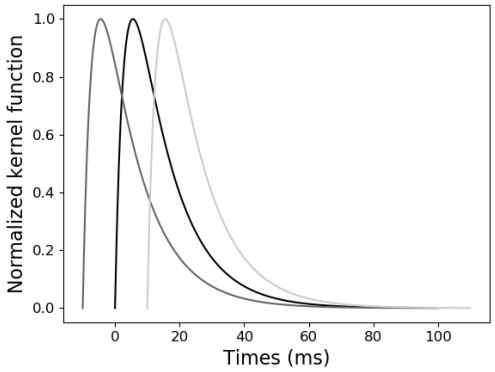


Fig. 5. Normalized kernel function.

we kept the weights greater than zero. ξ_i is the synaptic time-shifter of the ith afferent neuron. $\xi_i < 0$ represents that the spike will fire slightly later (the firing time moves forward). Conversely, $\xi_i > 0$ means that the firing time of a spike may move backward. ξ_i follows the Gaussian distribution with an average value of 0. Notice that the disturbance of the input spike sequence of a spiking neuron will directly affect its membrane potential. However, if the amount of change is not sufficient to change the magnitude relationship between the membrane potential value and the threshold, then the input interference will not affect the output spike sequence. The unweighted postsynaptic current of the ith afferent neuron is

$$I^{i}{}_{\text{post}}(t) = \sum_{t^j} K\left(t - t^j\right) H\left(t - t^j\right) \tag{14}$$

where t_j is the jth firing time of the ith afferent neuron, $H(t)$ is the Heaviside function, and K is the double exponential causal kernel function

$$K\left(t - t_j\right) = V_0\left(e^{\frac{-\left(t-t_j\right)}{\tau_p}} - e^{\frac{-\left(t-t_j\right)}{\tau_q}}\right) \tag{15}$$

where τ_p and τ_q are the respective decaying time constants of membrane integration and synaptic currents, and V_0 is normalized so that the maximum value of the kernel is 1. Here, we set $\tau_p = 4$ and $\tau_q = 12$ for MNIST [33], [34]. The kernel function is shown in Fig. 5.

Ivkovi *et al.* [35] showed that the weighted node degree of the brain network follows a normal distribution. Therefore, we set the synaptic shifters and synaptic weights to obey a normal distribution. The amplitude of each spike rises or falls due to the effect of synaptic weights. To get the best result, we tested the relevant precision as the mean value of the normal distribution changes. We study the performance influence of different distributions of ξ_i in Section IV-E.

D. Various Disturbance Factors

In the biological context, there exist various disturbances at the cellular level, including the inherent thermal stochasticity of molecular events, and variations of the activity of the ionotropic receptors [36] [37]. These effects are eventually reflected in the variation of the exact firing moments of the spike. However, most work either strives to emulate such phenomena at a sufficiently detailed level, resulting in small-scale SNNs, or completely ignores these factors. We try to emphasize the importance of the temporal perturbation in building an SNN model by introducing synaptic time-shifting units and providing corresponding SNN structures and a method to measure the sequences' similarity.

To be aware of the positive character of perturbation of the spike train is not new. We formalize the turbulence using a time-shifter unit and apply it to the SNN architecture, which, we believe, is the first attempt to emphasize the important role of temporal perturbation and introduce it as a necessary component in SNN architecture design from a practical perspective.

According to our investigation, perturbations that follow different distributions have the same effect on SNN performance, as discussed in Section IV-A. We choose a Gaussian distribution and prove that various disturbance factors can be reflected in a Gaussian perturbation ξ. Assuming that one of the input spikes j has a Gaussian perturbation, it contributes to the kernel function

$$K\left(t - \xi_j - t_j\right) = V_0\left(e^{\frac{-\left(t-\xi_j-t_j\right)}{\tau_p}} - e^{\frac{-\left(t-\xi_j-t_j\right)}{\tau_q}}\right) \tag{16}$$

$$= V_0\left(e^{\frac{-\left(t-t_j\right)}{\tau_p}} e^{\frac{\xi_j}{\tau_p}} - e^{\frac{-\left(t-t_j\right)}{\tau_q}} e^{\frac{\xi_j}{\tau_q}}\right). \tag{17}$$

Then, we have

$$\triangle K = K\left(t - \xi_j - t_j\right) - K\left(t - t_j\right) \tag{18}$$

$$= V_0\left(\left(e^{\frac{-\left(t-t_j\right)}{\tau_m}} e^{\frac{\xi_j}{\tau_m}} - e^{\frac{-\left(t-t_j\right)}{\tau_s}} e^{\frac{\xi_j}{\tau_s}}\right)\right). \tag{19}$$

According to the property of the exponential power of e, $e^{(\xi_j/\tau_m)}$, and $e^{(\xi_j/\tau_s)}$ also follow a Gaussian distribution, as does $\triangle K$, i.e., if $\triangle K$ has a Gaussian distribution, then we can find corresponding input spikes with Gaussian distributions that have the same effect. When $\xi_i > 0$, then $e^{(\xi_i/\tau_s)} > e^{(xi_i/\tau_m)} > 1$, and the curve of the kernel function will move left, and when $\xi_i < 0$, then $e^{(\xi_i/\tau_s)} < e^{(xi_i/\tau_m)} < 1$, and the curve of kernel function will move right. The corresponding curves are shown in dark and light gray, respectively, in Fig. 5.

E. Measurement Method of Time Sequence Similarity

After outputting h spike trains from output neurons, the next step is to measure the similarity among different spiking trains. The distance between time sequences is traditionally measured by directly calculating the Euclidean distance between the vectors of firing times. However, only the numerical value of the precise timing seems to have been used. We adopt the Van Rossum distance [38] to measure spike trains of different images. The distance of two spike trains can be described as

$$D = \int_0^\infty dt \left[\sum_{t^m} K\left(t - t^m\right) H\left(t - t^m\right) - \sum_{t^n} K\left(t - t^n\right) H\left(t - t^n\right)\right]^2 \tag{20}$$

where t^m and t^n are the respective firing times of the two spike trains, D is the distance of the two spike trains, and K is described in (15). This distance can be considered to

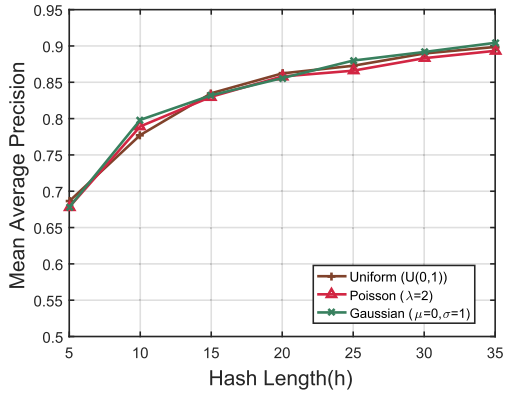


Fig. 6. Performance of TS-SLSH on the MNIST data set with time disturbances following different distributions.

be postsynaptic potentials in neurons, so the scheme can potentially utilize more information.

To reduce the time complexity of the distance measurement of temporal sequences, we further decrease the dimension of the output signals by segmenting the spiking train into different time intervals and counting the number of spikes in each segment. In this way, we obtain an output matrix of size $m \times h$, where m is the number of segments. From our experiments in Section IV-G, we find that using the matrix for similarity measurement obtains almost exactly the same performance as directly comparing the SNN distance when m is greater than 20, and it greatly improves the effectiveness of our algorithm.

IV. Experimental Results and Analysis

We describe our experiments to evaluate the effectiveness and performance of the proposed TS-SLSH model. We first describe the three data sets used in this article. Then, we compare the capabilities of the TS-SLSH, SLSH, LSH, and FLY models. We study the generality of TS-SLSH and SLSH for different neuron types, hash lengths, connection ratios, and noise levels, as well as different means and variances of perturbations following Gaussian distributions.

We mainly used the MNIST, SVHN, and SIFT data sets, selecting 10 000 images from each. MNIST includes ten classes of handwritten digital characters from 0 to 9. SVHN is a real-world data set obtained from house numbers of street-view photos. SIFT contains the 128-dimensional feature representations of some images from ImageNet.

A. Influence of Different Distributions

We conducted experiments with time distributions following different distributions (Gaussian, Poisson, and uniform), and the results were almost the same, as shown in Fig. 6. Hence, we chose a Gaussian distribution for the rest of our experiments.

B. Comparison of TS-SLSH, SLSH, LSH, and FLY

We compared the abilities of TS-SLSH, SLSH, LSH, and FLY to discover the most similar images in the three data sets. We converted the inputs to feature vectors (firing times) of dimension k ($k = 784$ for MNIST, $k = 1024$ for SVHN, and $k = 128$ for SIFT) using the encoding method in Section III-A. We randomly selected 100 images from these 10 000 inputs as samples. For each sample, we found the top 2% nearest neighbors from the whole data set in the input space with the Euclidean distance. After that, we found its top 2% nearest outputs after the TS-SLSH process, as measured by time sequence similarity, as mentioned in Section III-E. Finally, we used the average precision of these 100 samples to evaluate an algorithm's performance. We also tested the individual nearest-neighbor search performance of SLSH, LSH, and FLY. For each algorithm, we averaged the precision over ten times to reduce the variance problem caused by random seeds. A $20 \times h$ expansion was used for FLY. Other parameters are shown in Section IV-F. Here, we set $E = V_{\text{reset}} = 0$, $R_m = 1$, $V_{\text{th}} = 0.3$, and $\tau_m = 28$ for MNIST and SIFT and $\tau_m = 14$ for SVHN. We adjusted the parameters according to Qiang *et al.* [34]. We set the time-shifter to follow a normal distribution. Other experiments also used these hyperparameters, unless otherwise indicated. The parameters are described in Section III-B, and the structures are shown in Fig. 2.

The results are presented in Fig. 7(c). The x-axis represents the hash length (h), and the y-axis represents the average precision [39]. We find that, when the hash length expanded, the mean average precision of these algorithms increased, and TS-SLSH had much higher precision than the others. For example, when $h = 5$, for the MNIST data set, TS-SLSH had a 67.53% mean average precision, 3.24 times that of LSH (20.83%) and 1.45 times that of FLY (46.68%). For the SVHN data set, TS-SLSH had 61.53% mean average precision, 2.9 times that of LSH (21.22%) and 1.41 times that of FLY (43.56%). For the SIFT data set, TS-SLSH had 43.49% mean average precision, 1.94 times that of LSH (22.42%) and 1.05 times that of FLY (41.25%). Compared with the second best algorithm, when $h = 5$, the precision of TS-SLSH was 2%, 1%, and 10% better than for MNIST, SVHN, and SIFT, respectively. When $h = 35$, the precision of TS-SLSH was 13%, 19%, and 43% better, respectively. It can be observed that SLSH and TS-SLSH perform much better with a small hash length, which is potentially useful in real-world applications. Since the FLY model involves only simple numerical operations, when the hash length is small, the information of the input signal is far from fully expressed. As the hash length (output dimension) increases, the performance of FLY can be improved. However, for the TS-SLSH and SLSH models, due to the introduction of temporal information, even when the hash length is small, i.e., the output dimension (number of neurons) is small, each neuron can still take precise control of the release time of spikes and, thus, can fully represent features of the input stimulus. Thus, when the hash length is small (5 or 10), our model can have 21% better accuracy than FLY.

It can also be seen from the results that the gap between the precision of TS-SLSH and SLSH increased with the hash length. It is generally believed that the addition of perturbations (especially Gaussian noise) to a system, such as a neural network with many parameters and various complex nonlinearities, has no negative performance effect. Network performance can yield much by simple

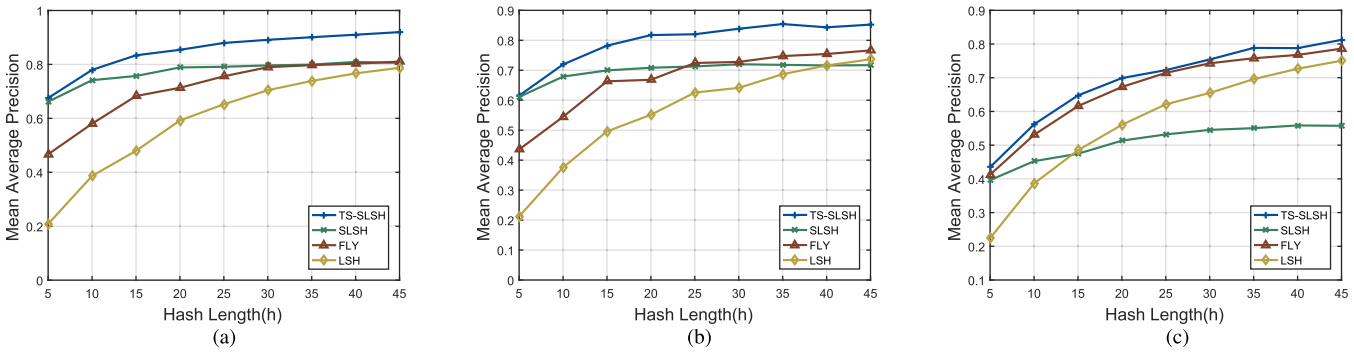


Fig. 7. Comparison of LSH, FLY, SLSH, and TS-SLSH using LIF neuron model. As the hash length increases, the network parameters increase exponentially, and the structure becomes more complicated. The introduction of the disturbance effectively increases SNN's capability of exploration and expression and, to some extent, relieves the overfitting phenomenon caused by complex parameters and network structure. TS-SLSH performs better when the hash length is larger and the network structure is more complicated. (a) MNIST. (b) SVHN. (c) SIFT.

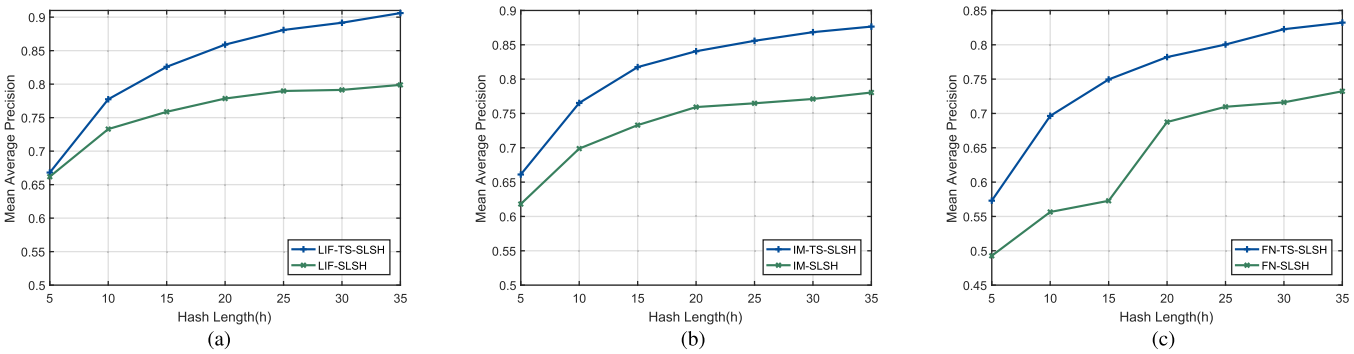


Fig. 8. Comparison of TS-SLSH and SLSH on spiking neuron models LIF, Levenberg-Marquardt learning (LM), and FN. (a) LIF neuron model. (b) IM neuron model. (c) FN neuron model.

reparameterization [40]. In biology, inside the neuron, the disturbance of parameters, such as membrane voltage and synaptic response current, will reflect in the perturbation of the timing of the pulse signal. From our investigation, introducing temporal disturbances (noise) to an SNN model can improve its expressiveness and robustness. Similarly, for SNN models based on supervised learning, the necessary disturbance can ensure that the network does not prematurely converge to a local optimal solution and can avoid the overfitting phenomenon to a certain extent.

From a biological point of view, the time-shifting on the release time of spikes is very common, and the neurons should have the ability to handle the input with perturbation within a certain range. In addition, SNN models manipulate the voltage changes between the neurons to perform information representation in order to fully express the characteristics of the input so that the differences between samples with different labels can be distinguished. The exploration ability of SNN is increased by the introduction of synaptic perturbation. Since the perturbations of initialized parameters of SNN are randomly generated, some inevitable system-level bias will exist. Instead of adding perturbation to the input signals, we introduce a temporal disturbance to the SNN model. Experiments illustrate that changing the parameter space appropriately will increase the exploration and robustness of the SNN model.

C. Generality to Different Neuron Models

To evaluate the model's generality to different spiking neuron models, we tested the performance of TS-SLSH and SLSH on neuron models LIF, IM, and FN, which are described in Section III-B. For fairness, these three kinds of neurons were connected to the same encoding neurons and were made to respond to the same stimulus.

We used 10 000 images of the MNIST data set for this experiment, and the results are presented in Fig. 8. The x-axis is the hash length (h), and the y-axis is the mean average precision. As the hash length (number of output neurons) grew, the mean average precision of TS-SLSH and SLSH increased. Although the structure of LIF is relatively simple, its performance is not inferior to the other two. The gaps among them increased gradually, despite the fluctuation in the process of SLSH when the FN neuron model was applied. On average, TS-SLSH gained 4.42% over SLSH with a hash length of 5 and 9.90% over SLSH when the hash length was 35. A shortcoming of SNNs is that many hyperparameters require manual adjustment. For different neuron models or learning algorithms, the introduction of a new method usually requires a certain degree of hyperparameter adjustment. Experimental results show the generality of our method, which can be applied to various neuron models without affecting the functionality of other parts of the SNN.

D. Robustness of TS-SLSH, SLSH, FLY, and LSH

In this experiment, we added salt-and-pepper noise to the three data sets. The noise level increased from 0% to 12%. We compared the performance of TS-SLSH, SLSH, FLY, and LSH on the data sets with hash length $h = 15$. Fig. 9 presents the testing results. We can observe that the mean average precision of these three algorithms declined as the

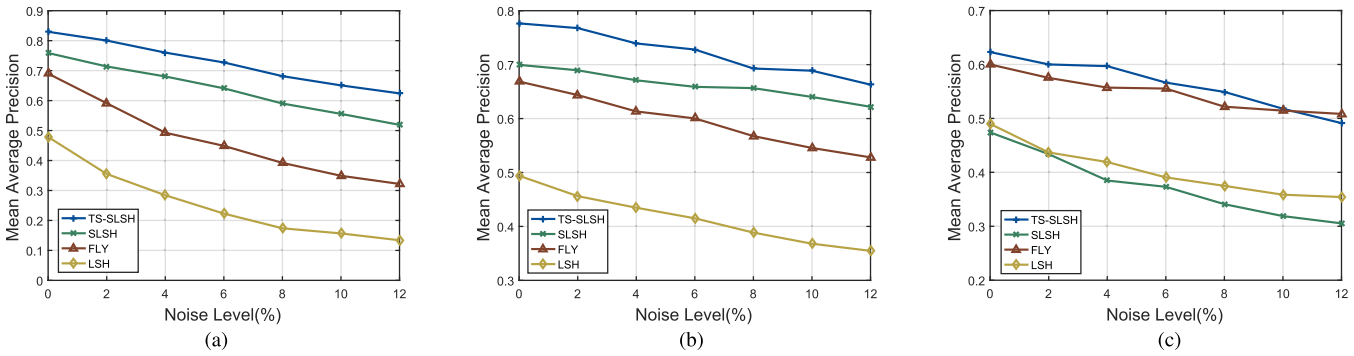


Fig. 9. Robustness of TS-SLSH, SLSH, FLY, and LSH. In all plots, the x-axis represents the noise level, the y-axis is the mean average precision, and the hash length is 15. On all three data sets, TS-SLSH outperforms the other algorithms. (a) MNIST. (b) SVHN. (c) SIFT.

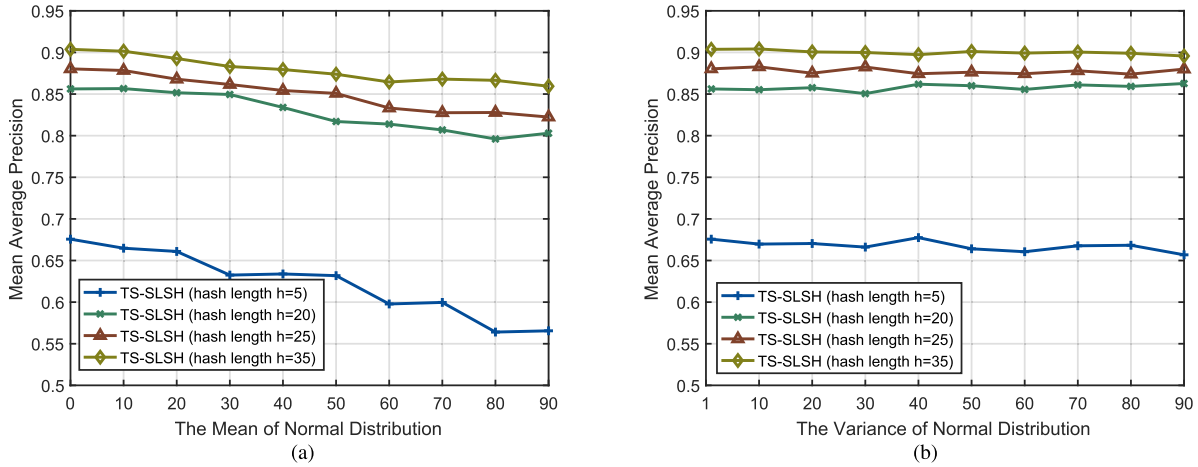


Fig. 10. Change of precision with different mean values of time-shifter for TS-SLSH on MNIST data set with different hash length. (a) Different mean values. (b) Different variance values.

noise level increased. The precision of TS-SLSH was still relatively higher than that of other algorithms, despite a slight decrease. For instance, for SVHN with a noise level of 12%, TS-SLSH had 66.29% precision, 86.9% and 6.7% more than LSH (35.47%) and FLY (62.18%), respectively. For TS-SLSH, the average accuracy dropped 20.56%, 11.36%, and 13.18%, for MNIST, SVHN, and SIFT, respectively. For SLSH, the accuracy dropped 23.97%, 7.77%, and 17.87%; for FLY, the accuracy dropped 36.81%, 14.05%, and 9.2%; and for LSH, the accuracy dropped 34.47%, 13.92%, and 13.57%, when the noise level increased from 0% to 12%.

E. Change of Precision With Different Mean Values and Variances of Time-Shifter

In Section III-C, we argued that the time-shifter obeys normal distribution. To study the effect of parameters on performance, we tested the relevant precision as the mean value and variance of the normal distribution changed. The results are shown in Fig 10.

In the figures, the x-axes are the mean value and variance of the normal distribution, and the y-axis represents the mean average precision. We can observe that when the hash lengths are 20, 25, and 35, as the mean value increases, the precision drops gradually despite a slight fluctuation around 5.5%, while as the variance value increases, the precision remains at the same level. The response of the system to a perturbation essentially depends on the time scale of its own dynamics. Therefore, SNN is resilient to small perturbations of spikes. When the mean value of time perturbation is less than 10 ms—3.3% of the time window (300 ms), the average precision is at the same level. However, the performance decreases with larger fluctuations, and the classification performance is resilient with different variances. These results are consistent with biological observation, where the internal disturbances of neurons are constrained. We also conducted experiments for different test sets. Fig. 11 shows almost no obvious difference for TS-SLSH with data sets of 100 and 1000 on MNIST.

F. Performance of TS-SLSH With Different Values of Connection Ratio η

In Section III-C, we applied a sparse and random projection matrix to connect different layers. To choose the most appropriate connection ratio, we also tested the performance influence of TS-SLSH under different values of η, especially for hash lengths of 5 and 10. We used 10 000 images of MNIST for this experiment. The results are presented in Fig. 12.

The x-axis represents the connection ratio, and the y-axis is the mean average precision. We can observe that, when η

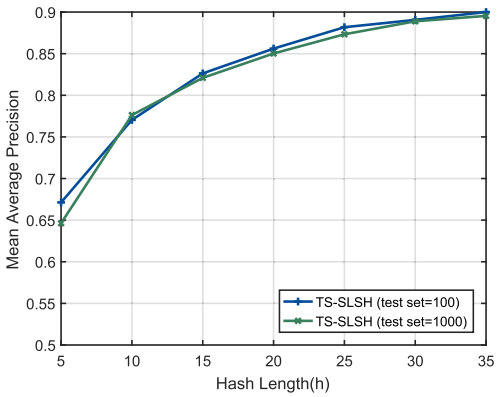


Fig. 11. Performance of TS-SLSH with different test sets on MNIST.

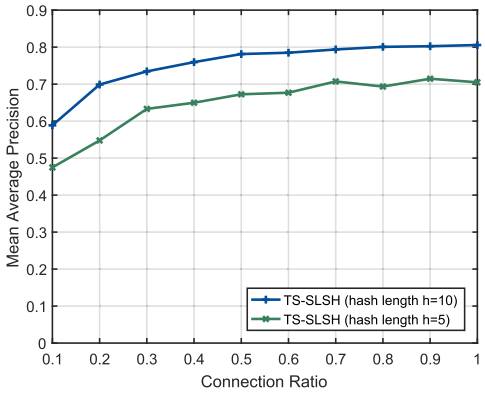


Fig. 12. Performance of TS-SLSH with different values of connection ratio.

is less than 0.5, the precision remains at a higher level even if there is a fluctuation. As the connection ratio drops from 1.0 to 0.5, the precision drops by 2.44% and 3.24%, respectively, for hash length $h = 10$.

Fig. 13(a) and (b) shows the distribution of the connection ratio of neurons for different hash lengths when the sparse ratio is 50% for the connect matrix. The x-axis is the number of output spiking neurons, equal to the hash length, and the y-axis is the connection ratio of each neuron. In FLY, every neuron is connected to the same number of neurons from the previous layer. According to our experiments, this condition can be relaxed. The standard deviation of the connection ratio is 0.0255 and 0.0197 for hash lengths of 5 and 10, respectively.

It is worth noting that, in FLY and TS-SLSH, neurons are connected by a sparse binary random connection matrix. For the former, every KC accepts and integrates spikes from around six PNs. However, in TS-SLSH, the number of connections is more flexible; each output neuron can accept any random number of input neurons. We believe that the random connection scheme of TS-SLSH can promote a model's ability to accommodate the diversity of learning samples and, thus, be able to distinguish between different samples.

G. Spike Train Similarity Measurement

Using the Van Rossum distance as the similarity metric can obtain good accuracy, but it will require a longer execution time. Although, for a random sparse connection, compared with the full connection method, a large part of the calculation is reduced, the time complexity of similarity measurement brought by the method of using the kernel function and inner product is relatively high, which does not meet the purpose of designing an efficient algorithm. Therefore, a sequence similarity measurement method based on time partitioning and spike counting is used and compared with different similarity measurement methods to verify its efficiency.

Experiments have proved that the coding method adopted by TS-SLSH is more suitable for the similarity measurement method based on the time partition and spike counting. To be specific, the spike train is divided into a fixed number of segments, and the number of spikes in each segment is recorded to express it as a vector of fixed length. The smaller the time window, the longer the vector obtained after segmentation. The smaller time window can represent the information more accurately, at the cost of increased computational complexity. Conversely, when the time window is large, the dimensionality reduction will result in the inability to fully express the information contained in the spike train. Therefore, it is important to choose an appropriate time window size. Notably, the temporal disturbance may cause changes in the partition results. For fine-grained time partitions, this change will be more apparent. For coarse-grained time partitions, it can tolerate small fluctuations but may lose some details of time information. Experiments have found that, for the three data sets, when using 20 time partitions, the algorithm achieves the best balance of accuracy and calculation time.

V. Hardware-Friendly SNN Design

From the point of view of software simulation and hardware implementation of SNNs, the precise time manipulation of spike signals requires a sophisticated design and relatively large overhead. For software simulation, time stamps must be sent along with the spikes to ensure accurate time control, which consumes additional communication bandwidth and storage overhead.

Several neuromorphic hardware implementations have been proposed recently, such as TrueNorth [41], SpiNNaker [42], and Loihi [43]. These digital architectures approximate the continuous-time dynamics of spikes with a fixed-size discrete model and apply synchronization mechanisms to maintain a consistent understanding of time. This allows spikes to be generated, routed, and consumed in an event-driven manner.

Many core architectures usually contain thousands of neuromorphic cores, where the parallelizable work of individual neuron state updates can be calculated concurrently. A dedicated on-chip network (OCN) is applied to transmit spiking messages between neuron units in the form of packets. Spikes are generated from any core (source neuron) and can be directed to any number of destinations (sink neurons). Considering the complexity of connections between neurons, the neuron-to-neuron fan-out degree is around 10 000. It is challenging to guarantee that spikes from the same neuron arrive at the destination at the same time. Otherwise, timestamps alongside the spikes will be required to ensure precise time control. Therefore, the performance of OCN is

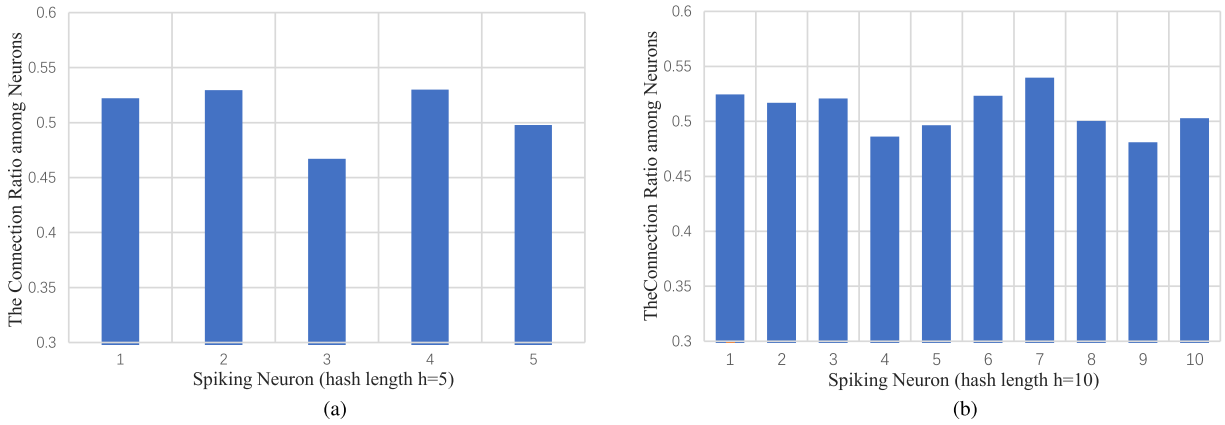


Fig. 13. Connection ratio of each neuron connected with the encoding neurons when the hash length is (a) 5 and (b) 10.

crucial to SNNs. Furthermore, maintaining a global synchronization mechanism arouses overhead, such as in terms of clock trees, power consumption, and chip area.

However, biological neural networks are fundamentally asynchronous, which is reflected in the absence of an explicit synchronization scheme. It significantly simplifies the back-end design timing closure and allows SNN to be easily scaled up to large sizes, where synchronization has always been considered the biggest obstacle. Therefore, from the perspective of software simulation and hardware implementation, the efficiency of the proposed design is proved.

VI. Conclusion

This article presents a forward spiking neuron architecture, TS-SLSH, which integrates time-shifter units and introduces temporal perturbations of spikes in the system and applies random connections between afferent neurons and output neurons. In addition, TS-SLSH can maximize the difference between corresponding pixels of any two images, which presents higher recognition effectiveness according to our experimental results. We also use convolution with a double exponential causal kernel to measure the similarity of spike trains.

Most people believe that accurate spike timing can help to form strict mathematical expressions, and a spike's disturbance is regarded as a drawback that must be overcome. However, biochemical reactions underlying all biological activities are probabilistic in nature. In this study, we observe that time-shifting and reflection of time perturbation in the spike transfer among neurons are inherent characteristics that help diversify the input samples. Experiments demonstrate that temporal perturbation, an essential component of biological information processing, can greatly promote the classification accuracy and robustness of a spike neural network. We view this work as an important result in light of ongoing research in neuromorphic communities with the introduction of spike perturbation as a necessary component.

References

[1] C. S. Calude and G. Longo, "Classical, quantum and biological randomness as relative unpredictability," *Natural Comput.*, vol. 15, no. 2, pp. 263–278, Jun. 2016.

[2] E. S. Kuebler, E. Bonnema, J. McCorriston, and J.-P. Thivierge, "Stimulus discrimination in networks of spiking neurons," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.

[3] J.-P. Thivierge and P. Cisek, "Nonperiodic synchronization in heterogeneous networks of spiking neurons," *J. Neurosci.*, vol. 28, no. 32, pp. 7968–7978, Aug. 2008.

[4] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, Nov. 2017.

[5] A. Andoni, P. Indyk, and R. Krauthgamer, "Earth mover distance over high-dimensional spaces," in *Proc. 19th ACM-SIAM Symp. Discrete Algorithms*, 2008, pp. 343–352.

[6] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases (VLDB)*, Edinburgh, U.K., Sep. 1999, pp. 518–529.

[7] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2006, pp. 459–468.

[8] A. M. Andrew, "Spiking neuron models: Single neurons, populations, plasticity," *Kybernetes*, vol. 32, no. 7/8, pp. 277–280, Oct. 2003.

[9] M. A. Khosruf, *Spiking Neural Networks*. New York, NY, USA: Springer, 2004.

[10] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Trans. Soc. Comput. Simul. Int.*, vol. 14, no. 4, pp. 1659–1671, 1997.

[11] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, 2009.

[12] Z. Wang, Y. Ma, Z. Dong, N. Zheng, and P. Ren, "Spiking locality-sensitive hash: Spiking computation with phase encoding method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.

[13] R. Kempter, W. Gerstner, and J. L. V. Hemmen, "Spike-based compared to rate-based Hebbian learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 11, 1998, pp. 125–131.

[14] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends Neurosci.*, vol. 33, no. 3, pp. 111–120, 2010.

[15] A. Borst and F. E. Theunissen, "Information theory and neural coding," *Nature Neurosci.*, vol. 2, no. 11, pp. 947–957, 1999.

[16] J. J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representation," *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.

[17] V. J. Uzzell and E. J. Chichilnisky, "Precision of spike trains in primate retinal ganglion cells," *J. Neurophysiol.*, vol. 92, no. 2, pp. 780–789, 2004.

[18] B. Y. Wang and M. P. Gong, "Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey," *Neural Comput.*, vol. 8, no. 6, pp. 1185–1202, 1996.

[19] Z. F. Mainen and T. J. Sejnowski, "Reliability of spike timing in neocortical neurons," *Science*, vol. 268, no. 5216, pp. 1503–1506, 1995.

[20] S. J. C. Caron, V. Ruta, L. F. Abbott, and R. Axel, "Random convergence of olfactory inputs in the drosophila mushroom body," *Nature*, vol. 497, no. 7447, pp. 113–117, May 2013.

[21] D. A. Butts *et al.*, "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.

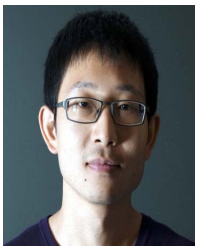
[22] K. Koepsell, "Retinal oscillations carry visual information to cortex," *Frontiers Syst. Neurosci.*, vol. 3, p. 4, Apr. 2009.
[23] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.*, vol. 25, no. 2, pp. 450–472, Feb. 2013.
[24] Z. Nadasdy, "Information encoding and reconstruction from the phase of action potentials," *Frontiers Syst. Neurosci.*, vol. 3, p. 6, Jul. 2009.
[25] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
[26] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing-based decisions," *Nature Neurosci.*, vol. 9, no. 3, pp. 420–428, 2006.
[27] J. M. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, no. 11, pp. 2881–2912, 2007.
[28] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: A spiking neural network training algorithm for classification problems," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1817–1830, Nov. 2010.
[29] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Competitive STDP-based spike pattern learning," *Neural Comput.*, vol. 21, no. 5, pp. 1259–1276, 2009.
[30] S. M. Bohte, J. N. Kok, and H. La Poutré "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, nos. 1–4, pp. 17–37, 2000.
[31] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010.
[32] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Fast and adaptive network of spiking neurons for multi-view visual pattern recognition," *Neurocomputing*, vol. 71, nos. 13–15, pp. 2563–2575, Aug. 2008.
[33] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS ONE*, vol. 8, no. 11, Nov. 2013, Art. no. e78318.
[34] Q. Yu, H. Tang, K. C. Tan, and H. Yu, "A brain-inspired spiking neural network model with temporal encoding and learning," *Neurocomputing*, vol. 138, pp. 3–13, Aug. 2014.
[35] M. Ivković, A. Kuceyeski, and A. Raj, "Statistics of weighted brain networks reveal hierarchical organization and Gaussian degree distribution," *PLoS ONE*, vol. 7, no. 6, Jun. 2012, Art. no. e35029.
[36] S. H. Wu, "Contribution of AMPA, NMDA, and GABAA receptors to temporal pattern of postsynaptic responses in the inferior colliculus of the rat," *J. Neurosci.*, vol. 24, no. 19, pp. 4625–4634, May 2004.
[37] Y. Ben-Ari, R. Khazipov, X. Leinekugel, O. Caillard, and J.-L. Gaiarsa, "GABAA, NMDA and AMPA receptors: A developmentally regulated 'ménage à trois,"' *Trends Neurosci.*, vol. 20, no. 11, pp. 523–529, 1997.
[38] M. C. W. V. Rossum, "A novel spike distance," *Neural Comput.*, vol. 13, no. 4, pp. 751–763, Apr. 2001.
[39] Y. Lin, R. Jin, D. Cai, S. Yan, and X. Li, "Compressed hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 446–451.
[40] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017, *arXiv:1703.03864*. [Online]. Available: http://arxiv.org/abs/1703.03864
[41] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.
[42] X. Jin, M. Lujan, L. A. Plana, S. Davies, S. Temple, and S. B. Furber, "Modeling spiking neural networks on SpiNNaker," *Comput. Sci. Eng.*, vol. 12, no. 5, pp. 91–97, Sep. 2010.
[43] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

Ziru Wang received the B.S. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2016, and the M.S. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2019.

Her current research interests include reinforcement learning and spiking neural networks.

Jiawen Liu received the B.S. degree in software engineering from the Taiyuan University of Technology, Taiyuan, China, in 2017, and the M.S. degree in software engineering from Xi'an Jiaotong University, Xi'an, China, in 2020.

Her current research interests include spiking neural networks and image recognition.



Yongqiang Ma (Student Member, IEEE) received the master's degree in software engineering from Xi'an Jiaotong University, Xi'an, China, in 2015, where he is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence and Robotics.

His current research interests include neuromorphic computing, neural network, and cognitive computing.

Badong Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in control theory and engineering from Chongqing University, Chongqing, China, in 1997 and 2003, respectively, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2008.

He was a Post-Doctoral Researcher with Tsinghua University, from 2008 to 2010, and a Post-Doctoral Associate with the Computational NeuroEngineering Laboratory, University of Florida, Gainesville, FL, USA, from 2010 to 2012. He visited the Nanyang Technological University, Singapore, as a Visiting Research Scientist in 2015 for one month. He is currently a Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China. He has published two books, four chapters, and more than 200 papers in various journals and conference proceedings. His current research interests include signal processing, information theory, machine learning, and their applications to cognitive science, and neural engineering.

Dr. Chen is a Technical Committee Member of IEEE SPS Machine Learning for Signal Processing and IEEE CIS Cognitive and Developmental Systems, and an Associate Editor of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the *Journal of the Franklin Institute*, and has been on the Editorial Board of *Entropy*.

Nanning Zheng (Fellow, IEEE) received the graduate degree in electrical engineering from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975, and the M.E. degree in information and control engineering from Xi'an Jiaotong University, in 1981, and the Ph.D. degree in electrical engineering from Keio University, Tokyo, Japan, in 1985.

He is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His current research interests include computer vision, pattern recognition, computational intelligence, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999.

Pengju Ren (Member, IEEE) received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2012.

From 2009 to 2011, he was a Visiting Ph.D. Student with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He is a Professor with the College of Artificial Intelligence, Xi'an Jiaotong University. His current research interests include on-chip network and computer architecture.