

Delay-weight plasticity-based supervised learning in optical spiking neural networks

YANAN HAN,¹ SHUIYING XIANG,^{1,2,*} ZHENXING REN,¹ CHENTAO FU,¹ AIJUN WEN,¹ AND YUE HAO²

¹State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China

²State Key Discipline Laboratory of Wide Band Gap Semiconductor Technology, School of Microelectronics, Xidian University, Xi'an 710071, China

*Corresponding author: jxxy@126.com

Received 29 October 2020; revised 13 December 2020; accepted 3 February 2021; posted 5 February 2021 (Doc. ID 413742); published 26 March 2021

We propose a modified supervised learning algorithm for optical spiking neural networks, which introduces synaptic time-delay plasticity on the basis of traditional weight training. Delay learning is combined with the remote supervised method that is incorporated with photonic spike-timing-dependent plasticity. A spike sequence learning task implemented via the proposed algorithm is found to have better performance than via the traditional weight-based method. Moreover, the proposed algorithm is also applied to two benchmark data sets for classification. In a simple network structure with only a few optical neurons, the classification accuracy based on the delay-weight learning algorithm is significantly improved compared with weight-based learning. The introduction of delay adjusting improves the learning efficiency and performance of the algorithm, which is helpful for photonic neuromorphic computing and is also important specifically for understanding information processing in the biological brain. © 2021 Chinese Laser Press

<https://doi.org/10.1364/PRJ.413742>

1. INTRODUCTION

As the improvement of traditional neural networks has gradually approached an upper limit, research focuses on neural networks with more biological reality. The spiking neural networks (SNNs), normally known as the third generation of artificial neural networks (ANNs), are more biologically plausible than previous ANNs [1] and have attracted more and more attention in recent decades [2–6]. Spikes transmitted in the biological neural networks enable the network to capture the rich dynamics of neurons and to integrate different information dimensions [3]. However, the information representation and processing manner has become a controversial issue and remains very challenging.

Rate coding is widely used in traditional SNNs; however, there is biological evidence that the precise timing of spikes also conveys information in nervous systems [7–9]. The precise timing of spikes enables higher information encoding capacity and lower power consumption [10–12], which are extremely important in information processing of the human brain. However, the exact learning mechanism still remains an open problem [13]. It has been shown that in the cerebellum and the cerebellar cortex, there exist signals that act like an instructor that helps the processing of information [14,15]. Several supervised learning algorithms have been proposed upon which specific problems that are tightly related to neural processing such as spike sequence learning and pattern recognition have been

solved successfully [16–20]. Remote supervised method (ReSuMe) is one of the supervised learning algorithms originally derived from the well-known Widrow-Hoff rule [17]. Based on photonic spike-timing-dependent plasticity (STDP) and anti-STDP rules, the synaptic weights can be adjusted to train the output neuron to fire spikes at the desired time.

Time-delayed transmission is an intrinsic feature in neural networks. Biological evidence shows that the transmission velocities in the nervous system can be modulated [21,22], for example, by changing the length and thickness of dendrites and axons [23]. The adjustability of both delay and weight of a synapse is referred to as synaptic plasticity. Delay plasticity has also been found to be helpful for the neuron in changing its firing behavior and synchronization, and it helps to understand the process of learning [24,25]. Delay selection and delay shift are two basic approaches incorporated in delay learning works [26,27]. To be specific, delay selection is to strengthen the weight of an optimal synapse among multiple subconnections, and delay shift is a more biologically plausible method training neurons to fire with coincident input spikes and with constant weight. Recently, researchers found that combined adjustment of delay and weight enhances the performance of an SNN [28–32]. In 2015, a delay learning remote supervised method (DL-ReSuMe) for spiking neurons was proposed to merge the delay shift approach and weight adjustment based on ReSuMe, by which the learning accuracy and learning speed are enhanced

[30]. In 2018, Taherkhani *et al.* proposed to appropriately train both weights and delays of excitatory and inhibitory neurons in a multilayer SNN to fire multiple desired spikes. Experimental evaluation on benchmark data sets shows that higher classification accuracy than single layer and a similar multilayer SNN can be achieved by the proposed method [31]. In 2020, Zhang *et al.* investigated the synaptic delay plasticity and proposed a novel learning method, where two representative supervised learning methods, ReSuMe and the perceptron based spiking neuron learning rule (PBSNLR), were studied and found to outperform the traditional synaptic weight learning methods [32].

For the sake of emulating realistic biological behaviors, SNN hardware realizations are designed to seek ultralow power consumption [5]. Devices for the implementation of basic elements of SNN, namely, artificial spiking neurons and synapses, have been achieved via complementary metal-oxide-semiconductors (CMOS), transistors, and the emerging non-volatile memory technologies [33–38], etc. Photonic neuromorphic systems have attracted attention for being a potential candidate in applications of ultrafast processing. Despite its similarity with biological neurons [39], the semiconductor laser also exceeds its electronic counterpart in its ultrafast response and low power consumption. Numerous studies on photonic synapses and photonic neurons [40–53] have laid a solid foundation for significant progress in photonic neuromorphic computing based on both software and hardware implementations [54–59]. In 2019, an all-optical SNN with self-learning capacity was physically implemented on a nanophotonic chip, which is capable of supervised and unsupervised learning [55]. In 2020, we proposed to solve XOR in an all-optical neuromorphic system with inhibitory dynamics of a single photonic spiking neuron based on vertical-cavity surface-emitting lasers (VCSELs) with an embedding saturable absorber (VCSEL-SA) [58], and an all-optical spiking neural network based on VCSELs was also proposed for supervised learning and pattern classification [59]. However, as far as we know, delay learning has not yet been applied in photonic SNNs.

In this work, we propose to incorporate delay learning with the traditional algorithm based on an optical SNN. First, we propose a modified algorithm that combines delay learning with ReSuMe in a photonic SNN, which adjusts synaptic weight and delay simultaneously. By implementing a spike sequence learning task, better performance and learning efficiency of the proposed algorithm than that of its weight-based counterpart are verified. Then, the proposed algorithm is also implemented for classification, where two benchmarks, the Iris data set and the breast cancer data set, are adopted. By applying the delay-weight (DW)-based algorithm, the testing accuracy for both benchmarks is significantly improved (reaching 92%).

2. SYSTEM MODEL

A. Photonic Neurons and Synapses Based on VCSEL-SA and VCSEA

The schematic diagram of DW-based supervised learning architecture is illustrated in Fig. 1, where the pre-synaptic neurons (PREs) of the input layer are fully connected to the single post-synaptic neuron (POST) in the output layer via photonic

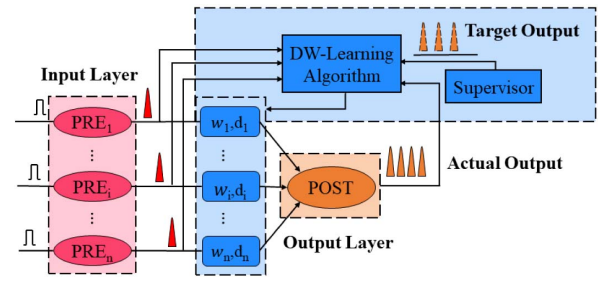


Fig. 1. Schematic diagram of DW-based learning in a single-layer photonic SNN.

synapses with adjustable weight ω and delay d . The actual output of the POST and the target output are sent to the DW learning algorithm module, based on which ω and d can be adjusted independently to train the POST to fire spikes at a desired time. To implement a spatiotemporal encoding, the PRE_i is stimulated by a pre-coded square-wave pulse whose central timing contains time information. In a possible experiment, the modulated spikes from PREs are sent into the DW module, which contains a programmable attenuator array for the modulation of weight and a programmable true delay line (TTDL) array for the adjusting of delay. The time resolution could reach 20 ps for the TTDL devices [60], which might be sufficient for the realization of DW-ReSuMe. A possible control unit can be introduced to detect the precise spiking times of the PRE and POST photonic neurons and calculate the variation of weight and delay for each synapse during a training cycle. The training process stops when accuracy meets the requirement. Note that given the technologies available, we think it may be more realistic to adopt an *ex situ* learning approach for the training process as offline training is much easier. Once trained, the SNN can be directly used for inference implemented on optical hardware [61].

The photonic neurons and synapses are the basic elements in a photonic SNN. VCSEL-SA can mimic spiking dynamics of a biological neuron [51] and VCSEL that works below threshold value can serve as a vertical-cavity semiconductor optical amplifier (VCSEA) that is able to perform the STDP function [47], which provides possibilities of large-scale integration and low power consumption in a photonic SNN. In this work, the spiking dynamics are implemented via the excitable VCSEL-SA neurons. The rate equations of a VCSEL-SA are written as follows [57]:

$$\begin{aligned} \dot{S}_{i,o} = & \Gamma_a g_a (n_a - n_{0a}) S_{i,o} + \Gamma_s g_s (n_s - n_{0s}) S_{i,o} \\ & - S_{i,o} / \tau_{ph} + \beta B_r n_a^2, \end{aligned} \quad (1)$$

$$\begin{aligned} \dot{n}_a = & -\Gamma_a g_a (n_a - n_{0a}) (S - \Phi_{pre,i} - \Phi_{post,o}) \\ & - n_a / \tau_a + I_a / (eV_a), \\ \Phi_{pre,i} = & k_{ei} \tau_{ph} \lambda_i P_{ei}(\tau_i, \Delta\tau) / (hcV_a), \\ \Phi_{post,o} = & \sum_{i=1}^n \omega_i \lambda_i \tau_{ph} P_i(t - d_i) / (hcV_a), \end{aligned} \quad (2)$$

$$\dot{n}_s = -\Gamma_s g_s (n_s - n_{0s}) S_{i,o} - n_s / \tau_s + I_s / eV_s, \quad (3)$$

where $i = 1, 2, \dots, n$ is the number of the PREs, and o denotes the POST. The subscripts a and s represent the gain and absorber regions, respectively. $S_{i,o}(t)$ stands for the photon density in the cavity of the PREs and POST. Γ_a is the gain region confinement factor. g_a is the gain region differential gain/loss. n_a (n_s) is the carrier density in the gain (absorber) region.

The term $\Phi_{pre,i}$ in Eq. (2) describes the pre-coded square-wave pulses injected as external stimulus into PREs, and $\Phi_{post,o}$ represents the weighted sum of all of the pre-synaptic spikes fed into the POST. d_i is the adjustable transmission delay from PRE $_i$ to the PSOT. k_{ei} , τ_i , and $\Delta\tau$ denote the strength, the central timing, and the temporal duration of the pulse, respectively. $\Delta\tau_i = \tau_i - \tau_{i-1}$ is the time interval between two adjacent input pulses. ω_i is the coupling weight between the PRE $_i$ and the POST that can be tuned according to the supervised training method. The output power of PREs and POST can be calculated by $P_{i,o}(t) \approx \eta_c \Gamma_a S_{i,o}(t) V_a h c / (\tau_{ph} \lambda_{i,o})$. In practice, the ω_i is calculated as an initial weight ω_0 multiplying a constant coefficient $\omega_f = \eta_c \Gamma_a h c / (\tau_{ph} \lambda_{i,o})$ to match the optical system. The remaining parameters are the same for all neurons as in Ref. [56]. The rate equations are numerically solved by using the fourth-order Runge-Kutta method.

B. DW-Based Learning Algorithm

In most of the algorithms with tasks tightly related to neural processing, such as spike sequence learning and pattern recognition, only weight adjustment is considered. However, the time delay from the PREs to the POST is hardly considered, which may play a vital role in brain computing [31]. Under this consideration, the time-delay plasticity is combined with the ReSuMe algorithm. The weight and delay changes of the i -th synapse after each training epoch are

$$\Delta\omega_i = (n_d - n_o) + \sum_{t_d} \sum_{t_i \leq t_d} \Delta\omega_{STDP}(t_d - t_i) + \sum_{t_o} \sum_{t_i \leq t_o} \Delta\omega_{aSTDP}(t_o - t_i), \quad (4)$$

$$\Delta d_i = (D_{id} - D_{io}); D_{id} = t_d - t_i, D_{io} = t_o - t_i, \quad (5)$$

where n_d and n_o are spike numbers of the desired and the actual output spike sequences. t_i , t_d , and t_o denote the input, target, and output spiking time, respectively. A schematic illustration of ReSuMe is shown in Fig. 2, where the $\Delta\omega$ depends on three parts: namely, the non-Hebbian term, the difference between target spiking time t_d and input time t_i , and the difference

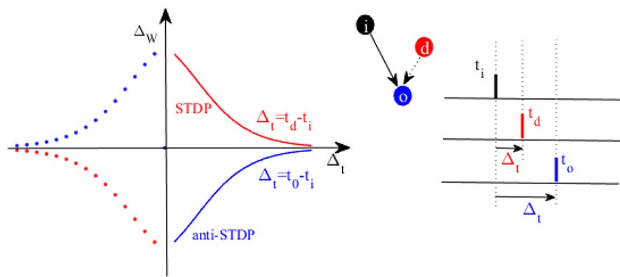


Fig. 2. Schematic illustration of the ReSuMe incorporated with optical STDP rule. i , d , and o denote the input, the target, and the output, respectively.

between actual output spiking time t_o and t_i . The first term of Eq. (4) is a non-Hebbian term that aims to adjust the average strength of the input synapses to accelerate training. The $\Delta\omega_{STDP}(t_d - t_i)$ and $\Delta\omega_{aSTDP}(t_o - t_i)$ in Eq. (4) are photonic STDP and anti-STDP learning rules denoting the synaptic potentiation (depression), which can be calculated by [43,57]

$$\Delta\omega_{STDP}(t_d - t_i) = \begin{cases} \Delta\omega_o(\Delta t), & \text{if } t_d - t_i > 0 \\ 0, & \text{if } t_d - t_i \leq 0 \end{cases}, \quad (6)$$

$$\Delta\omega_{aSTDP}(t_o - t_i) = \begin{cases} -\Delta\omega_o(\Delta t), & \text{if } t_o - t_i > 0 \\ 0, & \text{if } t_o - t_i \leq 0 \end{cases}. \quad (7)$$

$t_i \leq t_d$ and $t_i \leq t_o$ in Hebbian terms represent that this rule only modifies inputs that contribute to the neuron state before the desired or actual output firing time but neglect those inputs that fire afterward, which leads to better performance of the proposed DW-based algorithm.

The term D_{id} (D_{io}) is the distance between t_i and t_d (t_o). The delay adjustment is based on the distance between input spikes and output spikes (target spikes), in a way similar to that of synaptic weight. Note that both $\Delta\omega$ and Δd approach 0 if the POST fires at the desired times, which ensures the convergence of the proposed algorithm. In addition, the synaptic weights (delays) are adjusted with a learning rate η_f (η_d) and within a learning window T_w (T_d). The weight and delay of the i -th synapse are adjusted only if the input spike distance D_{id} is less than the learning window. Finally, the weight and delay of the i -th synapse are updated by

$$\omega_i(x+1) = \omega_i(x) + \eta_\omega \Delta\omega_i, \quad (8)$$

$$d_i(x+1) = d_i(x) + \eta_d \Delta d_i, \quad (9)$$

where the term x denotes the training cycle. In general, $\eta_d = 0.5$ contributes to better performance, while other parameters should be carefully selected according to different tasks.

A simple case for delay learning is illustrated in Fig. 3. Consider two PREs; each fires a spike at t_1 and t_2 , respectively. After delayed transmission through the synapses, the actual input spikes locate at $t_1 + d_1$ and $t_2 + d_2$ [see in Fig. 3(a)]. The principle of delay learning is to shift the actual input spikes toward the desired time, also called coherence learning, in

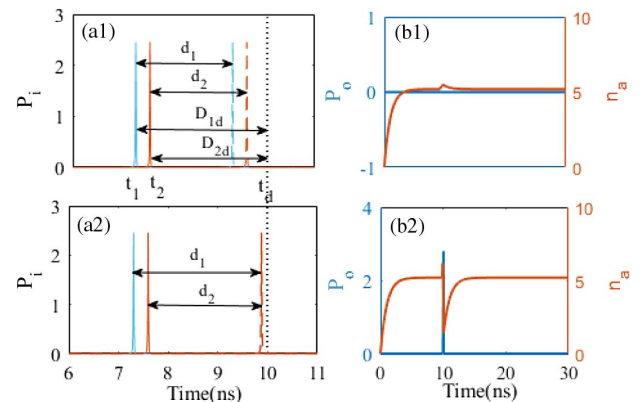


Fig. 3. (a1) and (b1) Input pattern and output pattern before delay adjustment. (a2) and (b2) After 7 training epochs.

which several coherent input spikes trigger the POST to fire a spike very shortly after the input time [Figs. 3(a2) and 3(b2)], while a single input spike cannot enable the release of spikes at the desired time [Fig. 3(b1)].

3. RESULTS AND DISCUSSION

The learning capability of a single neuron is highly related to the property of a neural network. In Fig. 4, we compare the performance of the weight-based ReSuMe and the DW-ReSuMe in recognizing a single spike, where $N_i = N_o = 1$, $d = 2$ ns, and $t_d = 8$ ns. To quantify convergence rate, the value of spike sequence distance (SSD) (defined in Ref. [57]) is adopted. The learning window T_ω is 3 ns, while T_d is not constrained. The input spiking time t_i varies from 4 ns to 7 ns. The time range within which an input pattern can be successfully learned, denoted as valid input window, suggests the learning ability of a neuron. The SSD at different training epochs is presented in Fig. 4 for the two algorithms. The valid input window for ReSuMe and DW-ReSuMe is 0.3 ns (4.4–4.7) and 2.4 ns (4.4–6.8) at the 50th training epoch, respectively, as shown in Figs. 4(a) and 4(b). It is shown that with DW-ReSuMe, the simple network is able to learn a single input spike with wider time range and with higher precision. Not surprisingly, with the increase of learning epochs, the valid input window also becomes a little wider. Note that weight learning is crucial when the single input is too weak to trigger an output spike in the POST. Here, we emphasize the importance of initial weight ω_0 and weight learning rate η_ω on the performance of DW-based learning. When ω_0 is too small, a larger learning rate of weight is required to generate an output spike before the delay learning has shifted the input spiking time right at that of the target. As can be seen in Fig. 4(c), the efficient input window is widened by increasing the learning rate from $0.1 \times \omega_0$ to $0.2 \times \omega_0$. In addition, for large ω_0 , smaller η_ω is required to avoid missing the optimal solution, as indicated in Fig. 4(d). The initial weight ω_0 ranges from a unit value of

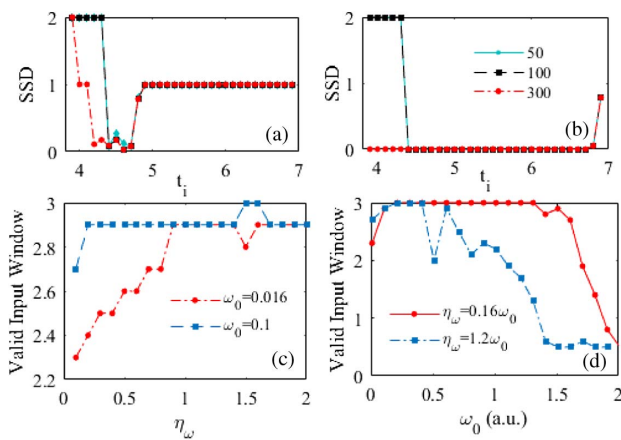


Fig. 4. Comparison of the learning capability of a single neuron based on (a) weight-based ReSuMe and (b) DW-ReSuMe. The value of SSD after the 50th, 100th, and 300th training epoch is presented for different t_i . (c) The valid input window as a function of η_ω for different ω_0 based on DW-ReSuMe. (d) The valid input window as a function of ω_0 for different η_ω based on DW-ReSuMe. $n = 1$, $t_d = 8$ ns.

0.1 to 2, and smaller η_ω contributes to a larger efficient input window. We suggest relatively larger weight learning rate η_ω and smaller initial weight ω_0 for spike sequence learning. For a classification problem, ω_0 should be large enough to trigger a spike of the PREs, while smaller η_ω is necessary to obtain higher accuracy.

However, note that if the desired output contains more than one spike, the delay learning window T_d has to be limited within the range of the minimum inter-spike interval (ISI) to maintain stable performance. Since some of the input spikes are shifted toward a certain target spike, there is an additional consideration in the DW-ReSuMe that the injection power of the output neuron should be limited to protect devices of photonic neurons and to ensure spiking dynamics.

A. Spike Sequence Learning

Then a spike sequence learning task is implemented via a single-layer photonic SNN. Both DW-ReSuMe and ReSuMe are considered for comparison. The PRE includes 60 input neurons stimulated via pre-coded rectangular pulses with a time interval $\Delta\tau_i$ of 0.1 ns, each connected to the POST with a photonic synapse. In Fig. 5 we show the learning process of the two algorithms. The black line in Fig. 5(a1) describes the carrier density n_a of the POST after training of 300 epochs based on DW-ReSuMe. When n_a exceeds a threshold value (marked in red dotted line), a spike is emitted. We can see that after training, the POST is able to fire accurately at the desired time (denoted by blue dotted line). The training process is further illustrated in Fig. 5(b1), where the SSD converges quickly from 2 to 0 within about 20 training epochs.

The evolution of synaptic weights and delays is shown in Fig. 6. The initial weights and delays of all neurons are identical, as indicated by Figs. 6(a) and 6(b), respectively. After training, some of the weights and delays are potentiated, while others are depressed or hardly changed. It is interesting to note that for some synapses, the weights and (or) delays change obviously at the first six training epochs. Such fluctuations during training are mainly caused by the combined effect of delay learning and weight learning. That is, when initially the output spiking time t_o is far before the target t_d , the update amount of both weight and delay is relatively large as the distance between

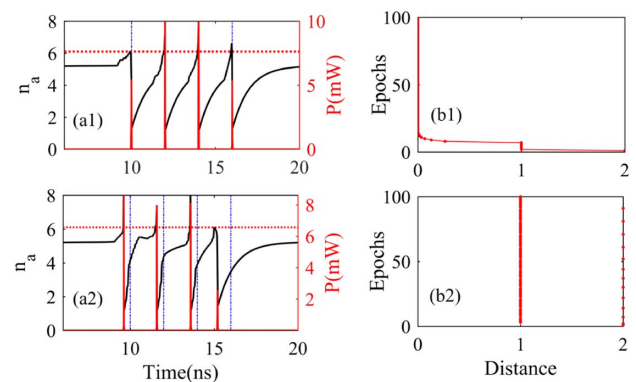


Fig. 5. (a1) Carrier density of the POST after training and (b1) the evolution of output spikes based on the DW-ReSuMe; (a2) and (b2) those based on ReSuMe. The black solid line is n_a and the red solid line represents P .

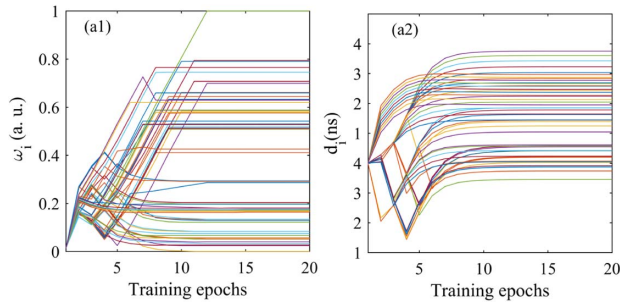


Fig. 6. Evolution of (a1) synaptic weights ω_i and (a2) delays d_i during the first 20 training epochs.

t_o and t_d is large. In this case, the combined adjustment of weight and delay may make t_o lag far behind t_d but closer to t_d than in the previous epoch. Note that the weight-based ReSuMe is not able to solve this problem in an SNN with 60 PREs, as shown in Figs. 5(a2) and 5(b2). The results show that combined with delay learning, DW-ReSuMe is more powerful than the weight-based algorithm.

Moreover, we found that in the cases of target spikes that have arbitrary different ISIs, the DW-ReSuMe also performs better than the traditional weight-based algorithm. Here, two cases are considered (refer to Ref. [57]). Figure 7 illustrates the learning process and evolution of SSD during 300 learning epochs. The desired spike sequence is [10 ns, 12 ns, 14 ns, 18 ns, 20 ns, 22 ns, 24 ns, 26 ns, 29 ns] in Figs. 7(a1) and 7(a2) and [10 ns, 11 ns, 13 ns, 14.5 ns, 17 ns, 21 ns, 23 ns, 25.5 ns, 27 ns] in Figs. 7(b1) and 7(b2). In both cases, with the increase of the learning epoch, the POST gradually learns to produce spikes at the desired time, and the values of SSD gradually decrease to approximate 0 after about 20 learning epochs and stay steady. Compared with our previous work [57], the network based on the DW-ReSuMe algorithm has better learning ability than that based on the weight-based ReSuMe.

B. Fisher's Iris Data Set

The Fisher's Iris flower data set (Fisher, 1936) is a classic benchmark of pattern recognition that contains three classes of Iris

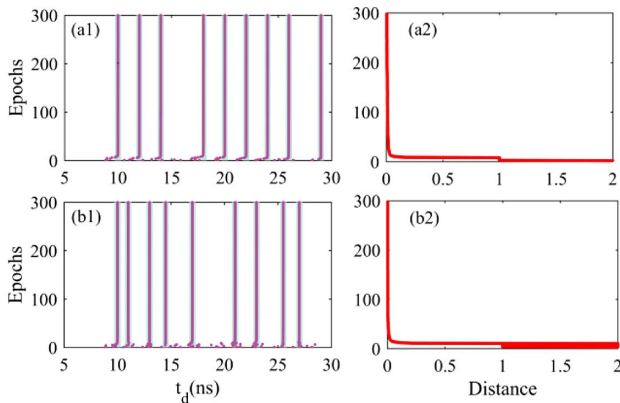


Fig. 7. Learning spike sequences with ununiformed ISI. (a1) and (b1) The evolution of output spikes for spike sequence [10 ns, 12 ns, 14 ns, 18 ns, 20 ns, 22 ns, 24 ns, 26 ns, 29 ns] and [10 ns, 11 ns, 13 ns, 14.5 ns, 17 ns, 21 ns, 23 ns, 25.5 ns, 27 ns], respectively. (a2) and (b2) The evolution for the corresponding distance.

flowers with a total of 150 case entries [31]. One of the classes is linearly separable from the other two, while the other two classes are linearly inseparable. Four measurements are used to describe and differentiate the three classes, and each measurement is directly encoded into single spike firing at different times and is linearly rescaled into the interval of [5 ns, 10 ns]. The network comprises 4 PREs and a single POST. The encoding spikes are fed into the 4 PREs of the SNN via synapses with adjustable delay and weight. The initial delays are 2 ns for all of the four input neurons, and the initial weight ω_0 for each synapse is randomly selected as a constant coefficient (which is 0.1 here) multiplied by a random number from [0,1]. In this case, nearly all of the PRE neurons are capable of generating just a single spike in response of each input pattern. The weight learning rate η_ω is 0.01 and decays by half every 20 training epochs to enhance the convergence of learning. The output of the network is represented by the precise spiking time of the POST, which fires a desired spike at different times for different classes. Here, the target spikes for the three classes are 8 ns, 9 ns, and 10 ns, respectively. If the output spike locates within 40% the interval of target spikes around a certain t_d , the input entry is classified into the corresponding class.

In our scheme, according to Ref. [31], 50% of the IRIS data are used for training and the rest for testing. The training and testing processes are implemented through program simulation based on MATLAB. For each training epoch, all of the entries in the training data set are injected into the input neurons of the SNN successively. Based on the learning algorithm, the $\Delta\omega_i$ (Δd_i) of the i -th input synapse for each entry is calculated and summed as ΔW_i (ΔD_i). After each training epoch, the mean value of ΔW_i (ΔD_i) is used for weight (delay) update, namely, the actual update amount for weight (delay) is $\Delta\omega_i = \Delta W_i / N_{\text{train}}$ ($\Delta d_i = \Delta D_i / N_{\text{train}}$), where $N_{\text{train}} = 75$ is the size of the training data set. After each training epoch, the updated weights and delays are also used for testing. The accuracy is defined as the number of correctly classified entries divided by the total entry number. Both training accuracy and testing accuracy with different learning algorithms are presented in Fig. 8. Based on the DW-ReSuMe, the accuracy arises rapidly at first and gradually approaches 96% for the training data set

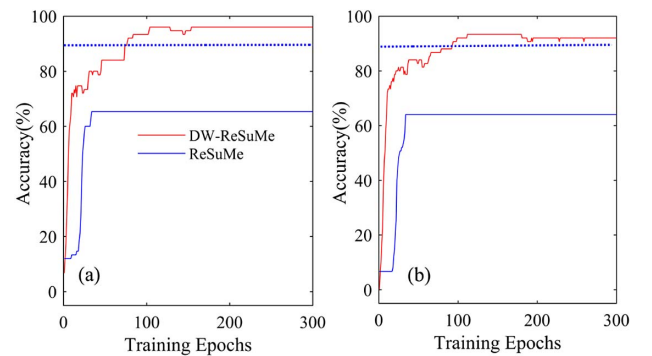


Fig. 8. (a) Training accuracy and (b) testing accuracy varying with training epochs for weight-based ReSuMe (blue solid line) and DW-ReSuMe (red solid line). $T_d = 1$ ns, $T_\omega = 4$ ns. The blue dotted line indicates an accuracy of 90%.

and 92% for the testing data set, as indicated by the red solid line.

For a detailed illustration of the classification results, a scattered plot of the target spiking time t_d and actual output spiking time t_o is shown in Figs. 9(a) and 9(b) for the training data set and testing data set, respectively. However, based on the same SNN architecture, the accuracy of the weight-based ReSuMe (blue solid line) only reaches 65%, with the same initial weights and delays, but a higher learning rate ($\eta_\omega = 0.1$) is required to make the algorithm work efficiently. The results indicate that the performance can be greatly enhanced by introducing time-delay learning.

Moreover, we also investigate the effect of learning window on the classification accuracy. The testing accuracy as a function of weight learning window T_ω and delay learning window T_d is shown in Figs. 10(a) and 10(b), respectively. We can see that a relatively larger T_ω is required for achieving higher accuracy. However, T_d should be selected as the minimum ISI of target spikes, which is 1 ns. Note that the accuracy can reach more than 85% even without the adjustment of weight. The results indicate that delay learning is an extremely efficient learning algorithm in a photonic SNN with temporal encoding, which suggests that delay learning may be an essential mechanism in the biological spiking neuron systems.

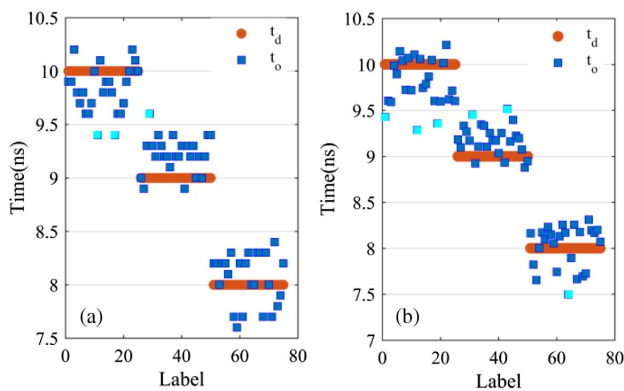


Fig. 9. Illustration of classification results for (a) training data set and (b) testing data set. The orange cycles denote target spiking time, the blue squares represent the actual spiking time, and misclassified samples are highlighted in bright blue.

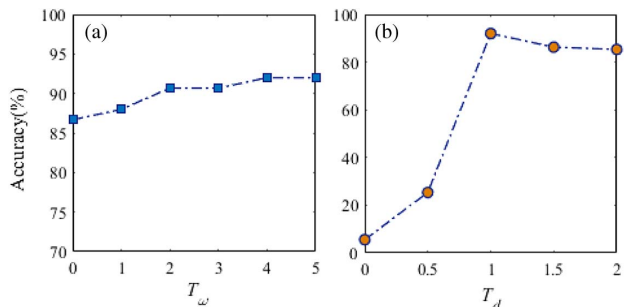


Fig. 10. Testing accuracy as a function of (a) weight learning window T_ω and (b) delay learning window T_d .

C. Breast Cancer Data Set

The breast cancer data set contains 608 case entries, and is divided into benign and malignant cases, each has nine measurements [62]. Five significant measurements are selected and pre-encoded into rectangular pulses, which trigger the input neurons to fire spikes at specific time within the range of 6.5 to 11 ns. The network contains 5 PREs and 1 POST. The initial delays are 0 ns for all of the 5 PREs, and the initial weights are randomly selected using the same method as for the Iris data set. The weight learning rate η_ω is 0.0001. The SNN is trained to fire a spike at 9 ns for the first case and at 13 ns for the second case. Four hundred eight entries are used to train the network, with the rest for testing. The accuracy on the training data set and testing data set reaches 93% and 92%, respectively, as shown in Figs. 11(a) and 11(b). The training data set is trained for over 100 epochs, and the accuracy converges quickly within 20 training epochs. However, with weight-based ReSuMe, the classification accuracy is less than 30% with the same operating parameters (not shown here). It is worth noting that in this case, the ReSuMe algorithm does not work efficiently, limited by the simple network structure and encoding schemes. The accuracy mainly depends on the initial synaptic delay d_0 . In Fig. 11, we also present the training results based on ReSuMe (blue solid line), with an initial delay of 0.5 ns. The weight learning rate here is 0.01.

Moreover, in consideration of the effect of initial delay d_0 on the training performance of ReSuMe and DW-ReSuMe, we show in Fig. 12 the accuracy of the two data sets after 60 training epochs as a function of different initial delays based on the two algorithms. Both training and testing results are considered. Figures 12(a1) and 12(a2) show the training and testing accuracy of the Iris data set, respectively, and the results of the breast cancer data set are shown in Figs. 12(b1) and 12(b2). Obviously, the proposed DW-based algorithm is less sensitive to varying initial delays. Note that in the Iris data set, the learning accuracy of DW-ReSuMe is small when d_0 is less than 1.5 ns, which is mainly constrained by the relatively small delay learning window (1 ns).

Finally, note that in Fig. 11 there is a downward fluctuation when the accuracy is close to the stable state. The fluctuations may be related to learning rate. A larger learning rate

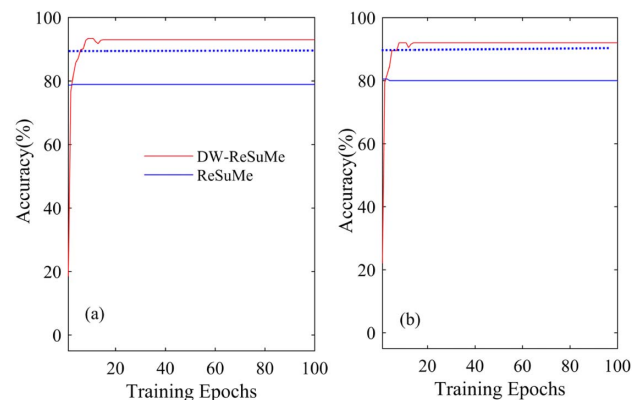


Fig. 11. (a) Training accuracy and (b) testing accuracy varying with training epochs based on DW-ReSuMe (red solid line) and ReSuMe (blue solid line), respectively. $T_d = 4$ ns, $T_\omega = 5$ ns.

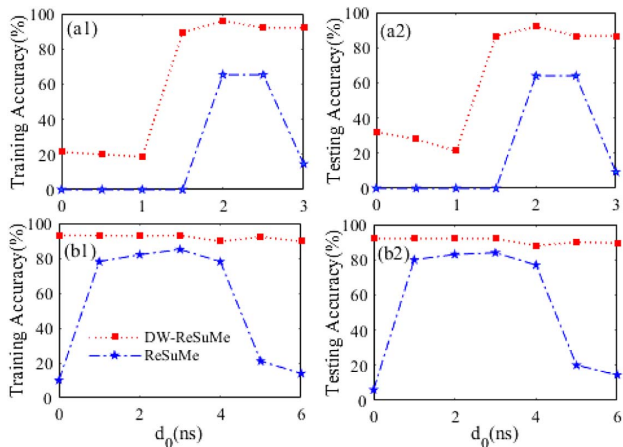


Fig. 12. (a1) Training accuracy and (a2) testing accuracy for the Iris data set after 60 training epochs with different initial delay d_0 . (b1) and (b2) The results for the breast cancer data set.

usually generates more obvious fluctuations, as illustrated in Figs. 13(a1) and 13(a2). The training accuracy fluctuates more obviously when the weight learning rate η_ω is larger. However, at about the 10th training cycle, the obvious downward fluctuation is not affected by weight learning rate. We can reasonably assume that the fluctuation is caused by the relatively larger constant delay learning rate ($\eta_d = 0.5$). As this value contributes to better training performance in different tasks, we consider a decaying η_d that is reduced by half after about five training epochs. The results are presented in Figs. 13(b1) and 13(b2), from which we can see that the fluctuation disappears. However, not surprisingly, the accuracy grows much more slowly.

DW-based learning has shown excellent performance in single-layer networks. However, the real neural networks are usually hierarchical, and synaptic weights and delays can be modulated based on different biological mechanisms, which form the foundations of complex brain functions. It is quite

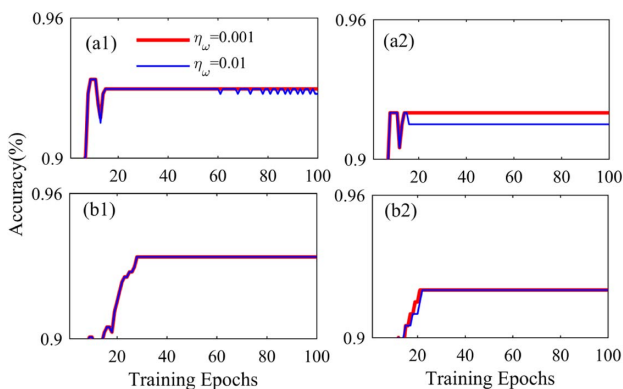


Fig. 13. Learning accuracy of the breast cancer data set based on DW-ReSuMe with different cases of η_d . The left column corresponds to the training accuracy with (a1) constant η_d and with (b1) decaying η_d . (a2) and (b2) The right column shows the corresponding results of testing accuracy. $T_d = 4$ ns, $T_\omega = 5$ ns.

interesting and challenging to consider how to effectively introduce delay adjustment into deep learning networks.

4. CONCLUSION

This paper proposed a supervised DW learning method in an optical SNN. Based on precise timing of spikes, delay learning trains coherent inputs in the input layer of an SNN via shifting the synaptic delays according to the desired and actual output timing. The proposed DW-ReSuMe is applied to spike sequence learning and two classification benchmarks, the Iris data set and breast cancer data set, successfully. The performance of the SNN is significantly improved compared with its weight-based counterpart. By introducing time-delay learning in a photonic SNN, fewer optical neurons are required to solve different tasks, which is significant to photonic neuromorphic computing. The results also suggest that synaptic delay and weight may be a combined learning mechanism in real biological neural networks, which deserves deeper investigation.

Funding. National Outstanding Youth Science Fund Project of National Natural Science Foundation of China (62022062); National Natural Science Foundation of China (61674119, 61974177).

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.* **19**, 295–308 (2009).
2. R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, Jr., M. Zirpe, T. Natschläger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. P. Davison, S. El Boustani, and A. Destexhe, "Simulation of networks of spiking neurons: a review of tools and strategies," *J. Comput. Neurosci.* **23**, 349–398 (2007).
3. N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," *Neural Netw.* **41**, 188–201 (2013).
4. K. Roy, J. Akhilesh, and P. Priyadarshini, "Towards spike-based machine intelligence with neuromorphic computing," *Nature* **575**, 607–617 (2019).
5. W. Q. Zhang, B. Gao, J. S. Tang, P. Yao, and H. Q. Wu, "Neuro-inspired computing chips," *Nat. Electron.* **3**, 371–382 (2020).
6. T. Clarence, M. Arlija, and N. Kasabov, "Spiking neural networks: background, recent development and the NeuCube architecture," *Neural Process. Lett.* **3**, 1675–1701 (2020).
7. A. Cariani, "Temporal codes and computations for sensory representation and scene analysis," *IEEE Trans. Neural Netw.* **15**, 1100–1111 (2004).
8. S. M. Bohte, "The evidence for neural information processing with precise spike-times: a survey," *Natural Comput.* **3**, 195–206 (2004).
9. A. Mohammed and S. Schliebs, "Training spiking neural networks to associate spatio-temporal input-output spike patterns," *Neurocomputing* **107**, 3–10 (2013).
10. S. B. Laughlin, R. R. de Ruyter van Steveninck, and J. C. Anderson, "The metabolic cost of neural information," *Nat. Neurosci.* **1**, 36–41 (1998).
11. S. B. Laughlin, "Energy as a constraint on the coding and processing of sensory information," *Curr. Opin. Neurobiol.* **11**, 475–480 (2001).
12. H. Paugam-Moisy and S. Bohte, "Computing with spiking neuron networks," in *Handbook of Natural Computing* (Springer, 2012), pp. 335–376.

13. J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.* **25**, 450–472 (2013).
14. F. Ponulak and A. Kasinski, "Introduction to spiking neural networks: information processing, learning and applications," *Acta Neurobiol. Experim.* **71**, 409–433 (2011).
15. H. Jörntell and C. Hansel, "Synaptic memories upside down: bidirectional plasticity at cerebellar parallel fiber-Purkinje cell synapses," *Neuron* **52**, 227–238 (2006).
16. R. Gütiğ and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nat. Neurosci.* **9**, 420–428 (2006).
17. F. Ponulak and A. Kasinski, "Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting," *Neural Comput.* **22**, 467–510 (2010).
18. I. Sporea and A. Grüning, "Supervised learning in multilayer spiking neural networks," *Neural Comput.* **25**, 473–509 (2013).
19. S. R. Kulkarni and B. Rajendran, "Spiking neural networks for handwritten digit recognition—supervised learning and network optimization," *Neural Netw.* **103**, 118–127 (2018).
20. C. Hong, X. Wei, J. Wang, B. Deng, H. Yu, and Y. Che, "Training spiking neural networks for cognitive tasks: a versatile framework compatible with various temporal codes," *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 1285–1296 (2020).
21. S. Boudkazi, L. Fronzaroli-Molinieres, and D. Debanne, "Presynaptic action potential waveform determines cortical synaptic latency," *J. Physiol.* **589**, 1117–1131 (2011).
22. J. W. Lin and D. S. Faber, "Modulation of synaptic delay during synaptic plasticity," *Trends Neurosci.* **25**, 449–455 (2002).
23. C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, "Dynamics of self-organized delay adaptation," *Phys. Rev. Lett.* **82**, 1594–1597 (1999).
24. X. B. Gong, Y. B. Wang, and B. Ying, "Delay-induced firing behavior and transitions in adaptive neuronal networks with two types of synapses," *Sci. China Chem.* **56**, 222–229 (2012).
25. M. Dhamala, V. K. Jirsa, and M. Ding, "Enhancement of neural synchrony by time delay," *Phys. Rev. Lett.* **92**, 074104 (2004).
26. S. Ghosh-Dastidar and H. Adeli, "Improved spiking neural networks for EEG classification and epilepsy and seizure detection," *Integr. Comput.-Aided Eng.* **14**, 187–212 (2007).
27. P. Adibi, M. R. Meybodi, and R. Safabakhsh, "Unsupervised learning of synaptic delays based on learning automata in an RBF-like network of spiking neurons for data clustering," *Neurocomputing* **64**, 335–357 (2005).
28. S. Ghosh-Dastidar and H. Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," *Neural Netw.* **22**, 1419–1431 (2009).
29. A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "EDL: an extended delay learning based remote supervised method for spiking neurons," in *Neural Information Processing* (Springer, 2015), pp. 190–197.
30. A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "DL-ReSuMe: a delay learning-based remote supervised method for spiking neurons," *IEEE Trans. Neur. Netw. Learn. Syst.* **26**, 3137–3149 (2015).
31. A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks," *IEEE Trans. Neur. Netw. Learn. Syst.* **29**, 5394–5407 (2018).
32. M. L. Zhang, J. B. Wu, A. Belatreche, Z. H. Pan, X. R. Xie, and Y. S. Chua, "Supervised learning in spiking neural networks with synaptic delay-weight plasticity," *Neurocomputing* **409**, 103–118 (2020).
33. W. Xu, S. Y. Min, H. Hwang, and T. W. Lee, "Organic core-sheath nanowire artificial synapses with femtojoule energy consumption," *Sci. Adv.* **2**, e1501326 (2016).
34. I. Sourikopoulos, S. Hedayat, C. Loyez, F. Danneville, V. Hoel, E. Mercier, and A. Cappy, "A 4-fJ/spike artificial neuron in 65 nm CMOS technology," *Front. Neurosci.* **11**, 123 (2017).
35. J. D. Zhu, Y. C. Yang, R. D. Jia, Z. X. Liang, W. Zhu, Z. A. Rehman, L. Bao, X. X. Zhang, Y. M. Cai, L. Song, and R. Huang, "Ion gated synaptic transistors based on 2D van der Waals crystals with tunable diffusive dynamics," *Adv. Mater.* **30**, 1800195 (2018).
36. B. Irem, L. G. Manuel, S. R. Nandakumar, M. Timoleon, P. Thomas, R. Bipin, L. Yusuf, S. Abu, and E. Evangelos, "Neuromorphic computing with multi-memristive synapses," *Nat. Commun.* **9**, 2514 (2018).
37. Y. Zhou, N. Xu, B. Gao, Y. Y. Chen, B. Y. Dong, Y. Li, Y. H. He, and X. S. Miao, "Complementary graphene-ferroelectric transistors (C-GFTs) as synapses with modulatable plasticity for supervised learning," in *IEEE International Electron Devices Meeting (IEDM)* (2019), pp. 1–4.
38. S. G. Hu, G. C. Qiao, Y. A. Liu, L. M. Rong, Q. Yu, and Y. Liu, "An improved memristor model connecting plastic synapse synapses and nonlinear spiking neuron," *J. Phys. D* **52**, 275402 (2019).
39. J. Ohtsubo, R. Ozawa, and M. Nanbu, "Synchrony of small nonlinear networks in chaotic semiconductor lasers," *Jpn. J. Appl. Phys.* **54**, 072702 (2015).
40. A. Hurtado, I. D. Henning, and M. J. Adams, "Optical neuron using polarization switching in a 1550 nm-VCSEL," *Opt. Express* **18**, 25170–25176 (2010).
41. A. Hurtado, K. Schires, I. Henning, and M. Adams, "Investigation of vertical cavity surface emitting laser dynamics for neuromorphic photonic systems," *Appl. Phys. Lett.* **100**, 103703 (2012).
42. M. P. Fok, Y. Tian, D. Rosenbluth, and P. R. Prucnal, "Pulse lead/lag timing detection for adaptive feedback and control based on optical spike-timing-dependent plasticity," *Opt. Lett.* **38**, 419–421 (2013).
43. Q. Li, Z. Wang, Y. Le, C. Sun, X. Song, and C. Wu, "Optical implementation of neural learning algorithms based on cross-gain modulation in a semiconductor optical amplifier," *Proc. SPIE* **10019**, 100190E (2016).
44. S. Y. Xiang, Y. H. Zhang, X. X. Guo, J. F. Li, A. J. Wen, W. Pan, and Y. Hao, "Cascadable neuron-like spiking dynamics in coupled VCSELs subject to orthogonally polarized optical pulse injection," *IEEE J. Sel. Top. Quantum Electron.* **23**, 1700207 (2017).
45. T. Deng, J. Robertson, and A. Hurtado, "Controlled propagation of spiking dynamics in vertical-cavity surface-emitting lasers: towards neuromorphic photonic networks," *IEEE J. Sel. Top. Quantum Electron.* **23**, 1800408 (2017).
46. Z. G. Chen, R. Carlos, W. H. P. Pernice, C. D. Wright, and H. Bhaskara, "On-chip photonic synapse," *Sci. Adv.* **3**, e1700160 (2017).
47. S. Y. Xiang, J. K. Gong, Y. H. Zhang, X. X. Guo, A. Wen, and Y. Hao, "Numerical implementation of wavelength-dependent photonic spike timing dependent plasticity based on VCSOA," *IEEE J. Quantum Electron.* **54**, 8100107 (2018).
48. Y. H. Zhang, S. Y. Xiang, J. K. Gong, X. X. Guo, A. J. Wen, and Y. Hao, "Spike encoding and storage properties in mutually coupled vertical-cavity surface-emitting lasers subject to optical pulse injection," *Appl. Opt.* **57**, 1731–1737 (2018).
49. I. Chakraborty, G. Saha, A. Sengupta, and K. Roy, "Toward fast neural computing using all-photonic phase change spiking neurons," *Sci. Rep.* **8**, 12980 (2018).
50. I. Chakraborty, G. Saha, and K. Roy, "Photonic in-memory computing primitive for spiking neural networks using phase-change materials," *Phys. Rev. Appl.* **11**, 014063 (2019).
51. Y. H. Zhang, S. Y. Xiang, X. Guo, A. Wen, and Y. Hao, "All-optical inhibitory dynamics in photonic neuron based on polarization mode competition in a VCSEL with an embedded saturable absorber," *Opt. Lett.* **44**, 1548–1551 (2019).
52. J. Robertson, E. Wade, Y. Kopp, J. Bueno, and A. Hurtado, "Toward neuromorphic photonic networks of ultrafast spiking laser neurons," *IEEE J. Sel. Top. Quantum Electron.* **26**, 7700715 (2020).
53. B. W. Ma and W. W. Zou, "Demonstration of a distributed feedback laser diode working as a graded-potential-signaling photonic neuron and its application to neuromorphic information processing," *Sci. China Inf. Sci.* **63**, 160408 (2020).
54. S. Y. Xiang, Y. H. Zhang, J. K. Gong, X. X. Guo, L. Lin, and Y. Hao, "STDP-based unsupervised spike pattern learning in a photonic spiking neural network with VCSELs and VCSOAs," *IEEE J. Sel. Top. Quantum Electron.* **25**, 1700109 (2019).
55. J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. P. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature* **569**, 208–215 (2019).

56. C. Ríos, N. Youngblood, Z. Cheng, M. L. Gallo, W. H. P. Pernice, C. D. Wright, A. Sebastian, and H. Bhaskaran, "In-memory computing on a photonic platform," *Sci. Adv.* **5**, eaau5759 (2019).
57. Z. W. Song, S. Y. Xiang, Z. X. Ren, G. Q. Han, and Y. Hao, "Spike sequence learning in a photonic spiking neural network consisting of VCSELs-SA with supervised training," *IEEE J. Sel. Top. Quantum Electron.* **26**, 1700209 (2020).
58. S. Y. Xiang, Z. X. Ren, Y. H. Zhang, Z. W. Song, and Y. Hao, "All-optical neuromorphic XOR operation with inhibitory dynamics of a single photonic spiking neuron based on VCSEL-SA," *Opt. Lett.* **45**, 1104–1107 (2020).
59. S. Y. Xiang, Z. X. Ren, Z. W. Song, Y. H. Zhang, X. X. Guo, G. Q. Han, and Y. Hao, "Computing primitive of fully-VCSELs-based all-optical spiking neural network for supervised learning and pattern classification," *IEEE Trans. Neural Netw.* (2020).
60. S. Moallemi, R. Welker, and J. Kitchen, "Wide band programmable true time delay block for phased array antenna applications," in *IEEE Dallas Circuits and Systems Conference (DCAS)* (2016), pp. 1–4.
61. G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. B. Miller, and D. Psaltis, "Inference in artificial intelligence with deep optics and photonics," *Nature* **588**, 39–47 (2020).
62. A. Roy, S. Govil, and R. Miranda, "An algorithm to generate radial basis function (RBF)-like nets for classification problems," *Neural Netw.* **8**, 179–201 (1995).