



GLSNN: A Multi-Layer Spiking Neural Network Based on Global Feedback Alignment and Local STDP Plasticity

Dongcheng Zhao^{1,2†}, Yi Zeng^{1,2,3,4**†}, Tielin Zhang¹, Mengting Shi^{1,2} and Feifei Zhao¹

¹ Research Center for Brain-Inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing, China, ² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, ³ Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing, China, ⁴ National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Spiking Neural Networks (SNNs) are considered as the third generation of artificial neural networks, which are more closely with information processing in biological brains. However, it is still a challenge for how to train the non-differential SNN efficiently and robustly with the form of spikes. Here we give an alternative method to train SNNs by biologically-plausible structural and functional inspirations from the brain. Firstly, inspired by the significant top-down structural connections, a global random feedback alignment is designed to help the SNN propagate the error target from the output layer directly to the previous few layers. Then inspired by the local plasticity of the biological system in which the synapses are more tuned by the neighborhood neurons, a differential STDP is used to optimize local plasticity. Extensive experimental results on the benchmark MNIST (98.62%) and Fashion MNIST (89.05%) have shown that the proposed algorithm performs favorably against several state-of-the-art SNNs trained with backpropagation.

Keywords: SNN, plasticity, brain, local STDP, global feedback alignment

OPEN ACCESS

Edited by:

Ke Zhou,
Beijing Normal University, China

Reviewed by:

Jiang Wang,
Tianjin University, China
Radwa Khalil,
Jacobs University Bremen, Germany

*Correspondence:

Yi Zeng
yi.zeng@ia.ac.cn

[†] These authors have contributed
equally to this work

Received: 27 June 2020

Accepted: 12 October 2020

Published: 12 November 2020

Citation:

Zhao D, Zeng Y, Zhang T, Shi M and
Zhao F (2020) GLSNN: A Multi-Layer
Spiking Neural Network Based on
Global Feedback Alignment and Local
STDP Plasticity.
Front. Comput. Neurosci. 14:576841.
doi: 10.3389/fncom.2020.576841

1. INTRODUCTION

Deep neural networks (DNNs) have been advancing the state-of-the-art performance in many domain-specific tasks, such as image classification (He et al., 2016), visual object tracking (Danelljan et al., 2015), visual object segmentation (Chen et al., 2017), etc. However, they are still far from the performance of efficiency and accuracy of information processing in the biological system. The structural connections (e.g., long-term feedback loops in the cortex) and functional plasticity (e.g., neighborhood plasticity based on discrete spikes) are carefully designed by the million years of evolution in the biological brain. This phenomenon has led to the research of biologically plausible Spiking Neural Networks (SNNs). SNNs have received extensive research in recent years, and have a wide range of applications in various domains, such as brain function modeling (Durstewitz et al., 2000; Levina et al., 2007; Izhikevich and Edelman, 2008; Potjans and Diesmann, 2014; Zenke et al., 2015; Breakspear, 2017; Khalil et al., 2017a,b, 2018), image classification (Zhang et al., 2018a; Gu et al., 2019), decision making (Héricé et al., 2016; Zhao et al., 2018), object detection (Kim et al., 2019), and visual tracking (Luo et al., 2020). The discrete spike activation and high dimension information representation in SNNs make it more biologically plausible and energy-efficient. However, due to the non-differentiable characteristics, how to properly optimize the strength of synapses to improve the performance of the whole-brain network is still an open question.

Hebbian theory (Amit et al., 1994) could be considered as the first principle to demonstrate the relations between neurons, with the description of fire together, wire together. Later, Spiking Time Dependent Plasticity (STDP) (Bi and Poo, 1998) was proposed to model the synaptic plasticity. All the methods mentioned above are based on local adjustments without introducing global plasticity information.

Learning and inference in the brain are based on the interactions of feedforward connections and mutual feedback connections across the hierarchy of cortical areas, as shown in **Figure 1A**. Both anatomical and physiological evidences point to the feedback connections in the brain (Felleman and Van, 1991; Sporns and Zwi, 2004). A large number of feedback connections in the cortex connect the feedforward series in the reverse order, thereby bringing global information from the higher cortex to the early cortical areas during perceptual inference. Feedback connections from higher layers will make predictions represented by the lower layers, and the feedforward path will get the state of neurons in the entire hierarchy. Therefore, combining global long-term feedback connections with local plasticity rules to train the SNNs is an urgent problem to be explored.

In this paper, we proposed an SNN training method that combines global feedback connections and local differential STDP learning rule and performs favorably against several existing state-of-the-art methods. The contributions of this paper are summarized as follows:

- We introduce the feedback connections in SNNs, which will help to introduce global plasticity information. The feedback connections are random, and no additional calculations are introduced.
- The global feedback connections combined with the local STDP plasticity rule are combined to directly optimize the synaptic strengths of all layers, instead of transferring error layer by layer as Back-Propagation. Compared with other

methods, it provides an alternative method for training deeper SNNs.

- Extensive experimental results on different datasets indicated that the proposed algorithm could significantly improve the learning ability of SNNs.

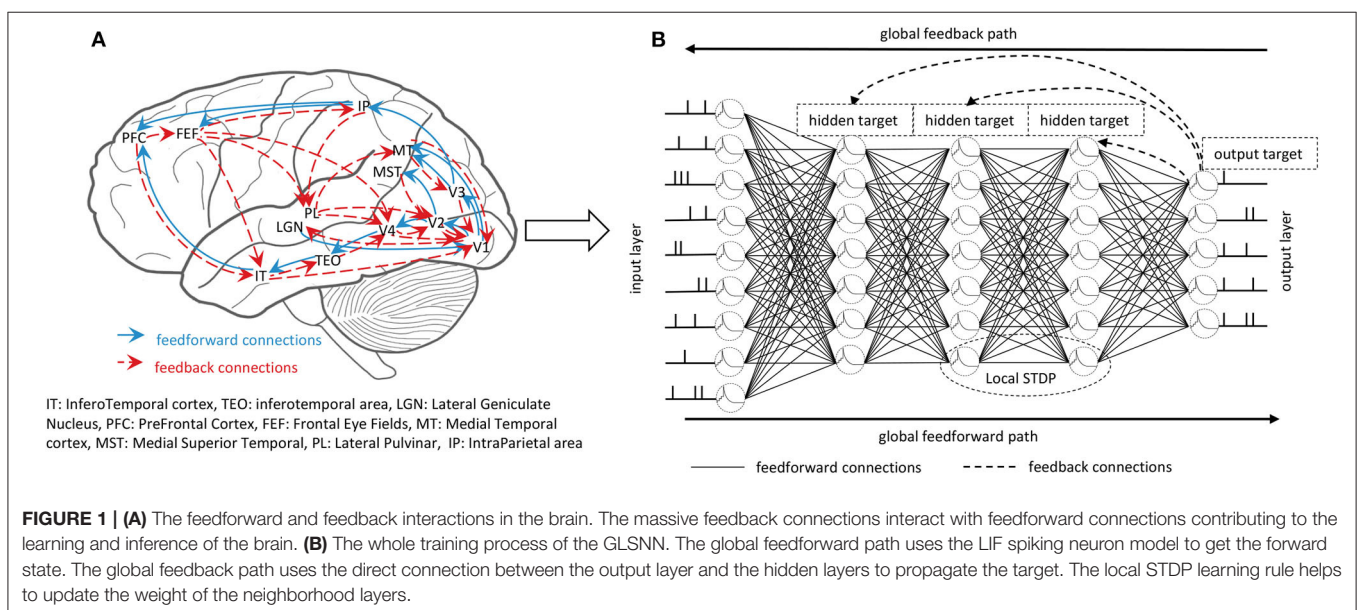
2. BACKGROUND

The success of DNNs is attributed mainly to the Back-Propagation algorithm (BP) (Rumelhart et al., 1986), which can take great advantage of the multilayer structure of neural networks to learn features related to a given task. However, firstly, the feedback path will have the symmetric weight of the forward path, which does not exist in biological systems, calling the weight transport problem (Lillicrap et al., 2016). Secondly, the precise derivatives of the operating point used in the corresponding feedforward path are needed. While for SNNs, information is transmitted in discrete spikes, and it is difficult to get the precise derivative of the operating point. Thirdly, the errors propagate layer by layer, which can easily lead to the problem of gradient vanishes or explosion. To tackle the problems mentioned above, many other learning rules are proposed to train the ANNs and further extended to train SNNs. In this section, we will review several of these approaches and several SNN frameworks in recent years.

2.1. Biologically Plausible Methods in ANNs

Recently, non-BP methods used to train neural networks can be roughly divided into three categories.

One family of promising approaches is Contrastive Hebbian Learning (Movellan, 1991). Equilibrium Propagation approaches (Scellier and Bengio, 2017) can be seen as a particular case of Contrastive Hebbian Learning. These kinds of energy-based models consist of two phases, the free phase is used



to achieve the stationary distribution, and the clamp phase is used to update the network toward the target. Through the iteration of these two phases, the energy of the network can reach convergence gradually. However, due to the indirect feedforward process, the network state is obtained by minimizing the energy function. When the network becomes deeper, the entire algorithm will be unstable and therefore, difficult to train. We will give the experimental results below. Similarly, the free phase (feedforward propagation) and the clamp phase (feedback propagation) use the same weights, and the weight transpose problem still exists, as mentioned in backpropagation.

In order to solve the weight transport problem, the Random Feedback Alignment (RFA) algorithm (Lillicrap et al., 2016) uses a fixed random matrix B instead of the transposition of synaptic weights W , which can enable the network to converge to the optimal solution efficiently. Subsequent work DFA (Nøkland, 2016) propagates error signals through the direct connection matrix between the output layer and hidden layers. However, the error feedback does not influence the neural activity, which has not been confirmed by known biofeedback mechanisms based on neural communication.

In the Target Propagation (TP) family, for Difference Target Propagation (DTP) (Lee et al., 2015), targets for each hidden layer are passed through feedback connections, which avoids the weight transport problem, as the feedback connections are different from feedforward connections. The error-driven local representation alignment (LRA-E) (Ororbia and Mali, 2019), attempt to calculate the local target with the local error loss. Random feedback connections are utilized to transmit errors. However, the error is calculated and propagated layer by layer, and as the network deepens, performance will deteriorate.

2.2. Spiking Neural Networks

Much effort has been put into training SNNs, which can be roughly divided into three categories. First, directly convert the well-trained ANNs to SNNs. Second, SNNs are processed in some unique methods so that they can be trained with BP. Third, training SNNs with STDP and other biologically plausible methods.

For the conversion methods, SDBN (O'Connor et al., 2013) mapped an offline-trained deep belief network (DBN) onto an efficient event-driven SNN based on the Siegert approximation. The LIF response function is softened to lead to the bounded derivative value, which helps SDN (Hunsberger and Eliasmith, 2015) to convert the trained static network to a dynamic spiking network. WTSNN (Diehl et al., 2015) converted the DBNs into SNNs through weight and threshold balancing. Although these networks achieve good performance, the good results came from the well-trained ANNs, which does not reflect the characteristics of SNNs well.

For the BP training methods, DSN (O'Connor and Welling, 2016) proposed that SNN is equivalent to a deep network of ReLU units, and could be directly trained with BP. Event-SNN (Nefci et al., 2017) demonstrated an event-driven random BP rule for learning deep representations. SCSNN (Wu et al., 2019) used spike count as a surrogate for gradient backpropagation. BPSNN (Lee et al., 2016) treated the membrane potentials of

spiking neurons as differentiable signals, which enabled the backpropagation. HM2-BP (Jin et al., 2018) proposed a hybrid macro/micro level backpropagation algorithm for training multi-layer SNNs. Temporal SNN (Mostafa, 2017) trained the SNN with temporal coding. STBP (Wu et al., 2018) trained the SNNs with BP both in spatial and temporal domains. The excellent performance of these methods came from BP, which turns out to not existed in the brain.

For STDP and other biologically plausible methods, Unsupervised-SNN (Diehl and Cook, 2015) trained an SNN with STDP, lateral inhibition, and an adaptive spiking threshold with a poor little performance 95% on the MNIST dataset. LIF-BA (Samadi et al., 2017) approximated dynamic input-output relations with piecewise-smooth functions based on fixed feedback weights. STCA (Gu et al., 2019) trained SNNs with credit assignments both in spatial and temporal domains. Both of them update the weights layer by layer. VPSNN (Zhang et al., 2018a) and Balance-SNN (Zhang et al., 2018b) trained the SNNs with Equilibrium Propagation, Balance-SNN is an improved version of VPSNN, which introduced much more learning rules to get the training balance of SNNs. However, as they trained with Equilibrium Propagation, the problems in Equilibrium Propagation also exist in both of them.

To sum up, a model to propagate the global plasticity information with a random feedback connection directly to each layer combined with the local plasticity learning rule to train SNNs has so far been rarely studied.

3. METHODS

The pipeline of our model is shown in **Figure 1B**. First, we will introduce the spiking neuron model used in our framework. Second, the global and local plasticity learning process will be introduced. Third, the whole framework will be introduced to understand our model better.

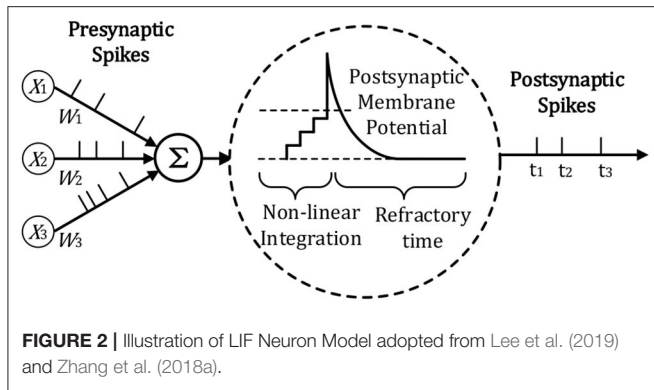
3.1. The Basic LIF Neuron Model

The spiking neuron model we use for temporal information processing is the Leaky integrate-and-fire (LIF) model, which is widely used in most SNN frameworks. As can be seen in **Figure 2**, for the LIF model, the neuron will accumulate the potential from the input, once its potential reaches the threshold, the neuron will be fired with a spike.

Generally, the membrane potential V can be calculated with Equation (1)

$$I(t) - \frac{V(t)}{R_m} = C_m \frac{dV(t)}{dt} \quad (1)$$

R_m is the membrane resistance and C_m denotes the membrane capacitance. $I(t)$ denotes the total input current from pre-synaptic neurons. For simplicity, we denote $V(t)$ with V , $I(t)$ with I , g_L and V_L denote leaky conductance and leaky potential. In a network with a more realistic synapse model, the input current I is generated as a change in conductance, which is caused by spikes of presynaptic neurons. The excitatory conductance g_E will be non-linearly increased by the number of the input spikes δ_j (Gerstner et al., 2014). V_E is the reversal potential from neuron



i to neuron j . When the membrane reaches the threshold, the neuron will produce a spike, and the membrane will be reset to V_{reset} . $\tau_m = \frac{C_m}{g_L}$, τ_E is the conductance decay of excitatory neurons, $w_{j,i}$ is the synapse weight from neuron j to neuron i .

$$\begin{cases} \tau_m \frac{dV_i}{dt} = -(V_i - V_L) - \frac{g_E}{g_L}(V_i - V_E) \\ \tau_E \frac{dg_E}{dt} = -g_E + \sum_j^N w_{j,i} \delta_j \end{cases} \quad (2)$$

3.2. The Global Plasticity Learning Process of Our Model

The global plasticity learning process is applied to a multi-layer feedforward neural network to illustrate better our learning algorithm, in which neurons in the previous layer are fully connected to the subsequent layer. In the adjacent layers, information from pre-synaptic neurons will be transferred to the post-synaptic neurons. For a deep spiking neural network, if only the spike is used, it will take a long time for the information transfer to the subsequent deeper layers, which will make the network hard to converge. To solve the problems, Diehl and Cook (2015) has used the spike trace to adjust the network weights, Zhang et al. (2018a) and Lee et al. (2016)'s work use voltage-based weight adjustments. Inspired by the residual neural network (He et al., 2016), which transfers the information as $x + f(x)$, here we think that in addition to the spikes output by the LIF neuron can be used to regulate the weight, the input to the LIF neuron also contains a wealth of information. The final output of the neuron is denoted as $S_j(t + 1)$. To convert Equation (2) into discrete form, the whole process is shown in Equation (3):

$$\begin{cases} V_i(t + 1) = V_i(t) - \frac{dt}{\tau_m} [V_i(t) - V_L + \frac{g_E}{g_L}(V_i(t) - V_E)] \\ g_E(t + 1) = g_E(t) + \frac{dt}{\tau_E} (-g_E(t) + \sum_j^N w_{j,i} S_j(t + 1)) \\ \delta_i(t + 1) = 1 \quad V_i = V_{reset} \quad \text{if } V_i > V_{th} \\ S_i(t + 1) = \sum_j^N w_{j,i} S_j(t + 1) + \tau \delta_i(t + 1) \end{cases} \quad (3)$$

τ is the constant to control the magnitude of the output. To accelerate the calculation, we only calculate the loss at the end of the simulation to update the target and weight. We denote the target with S^T , S_{out} denotes the output of the last layer, M is the number of the samples. For the output layer, the loss function we

choose here is the L2 norm so that the prediction error can be written as Equation (4):

$$\begin{cases} loss = \sum_{i=1}^M \|S_{out} - S^T\|^2 \\ e = 2 * \sum_{i=1}^M |S_{out} - S^T| \end{cases} \quad (4)$$

Supposing a network with L layers. The output of the l_{th} layer is denoted with S_l . For supervised learning, the target of the penultimate layer \hat{S}_{L-1} can be directly calculated, as shown in Equation (5), W_l denotes the forward weight between the l_{th} layer and the $(l + 1)_{th}$. η_t represents the learning rate of the target.

$$\hat{S}_{L-1} = S_{L-1} - \eta_t \Delta S = S_{L-1} - \eta_t W_{L-1}^T e \quad (5)$$

For the target of the other hidden layers, the target can not be directly calculated as Equation (5). By introducing the feedback connections, the prediction error can be easily transmitted to the hidden layers, and we denote the feedback layer as G_l . Moreover, the target of the hidden layer can be written as Equation (6):

$$\begin{cases} \hat{S}_l = S_l - G_l(e) \\ G_l(e) = B_l * e + b_l \end{cases} \quad (6)$$

B_l denotes the random feedback weight of the l_{th} layer, and b_l represents the random feedback bias. With the operation of all layers, we can directly get the target of each layer.

3.3. The Local Learning Process of Our Model

STDP can be seen as the leading learning rule in the brain, and it can simulate the expected change of synaptic weights depending on states between pre-synaptic and post-synaptic (Bi and Poo, 1998), which can be regarded as a local learning rule. As introduced in (Xie and Seung, 2000; Hinton, 2007), STDP is associated with the change of postsynaptic activity. Here we use the difference between the feedforward state and feedback state to denote the change, as shown in Equation (7).

$$\Delta W \propto S_j S'_i = S_j(S_i - \hat{S}_i) \quad (7)$$

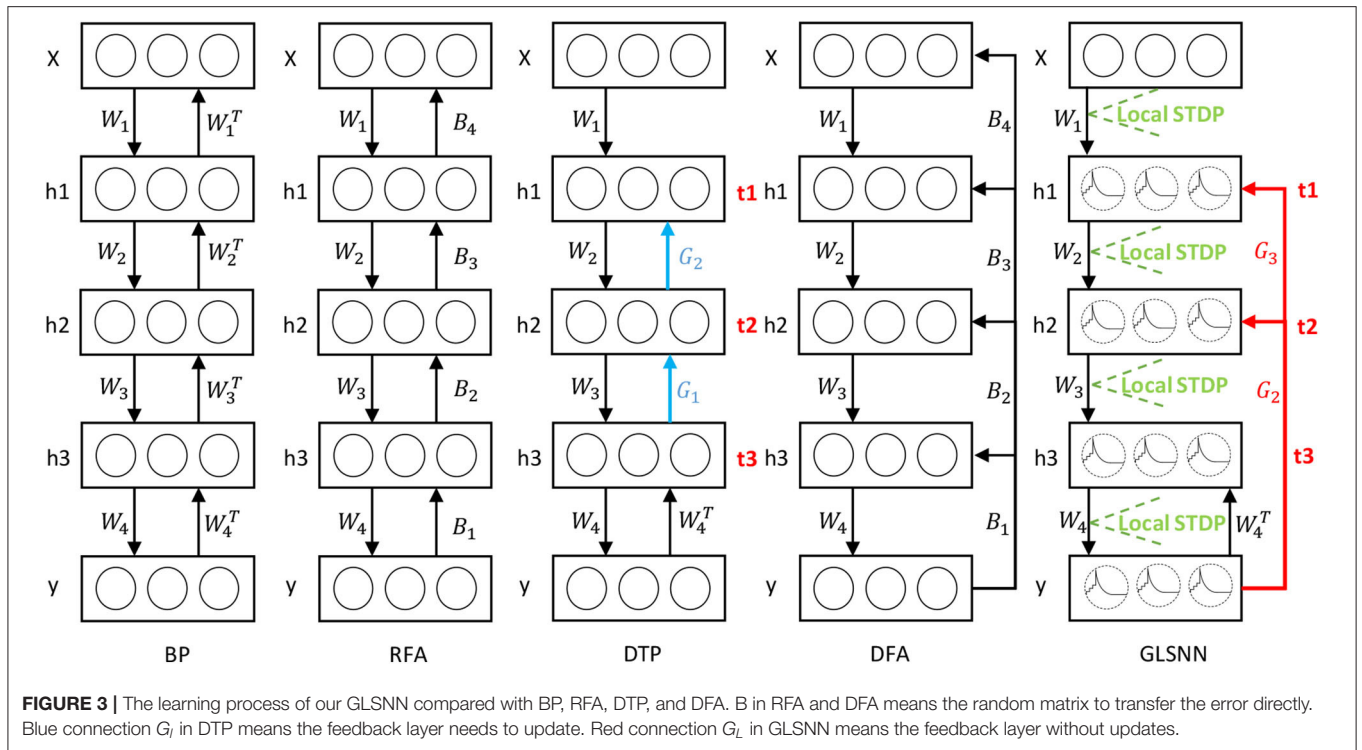
where S_j and S_i indicate the pre-synaptic and post-synaptic output in the forward learning process. \hat{S}_i denotes the target of the i_{th} layer calculated in Equation (6).

3.4. The Whole Learning Framework

For a multi-layer feedforward SNN, global plasticity information should be introduced so that STDP can train the whole network to obtain the desired result. Firstly, the feedforward process is used to obtain the feedforward state of the network, and then the feedback is used to obtain the targets of different hidden layers. Then, the change of weights in different neighborhood layers are calculated by local STDP plasticity rule in Equation (7). Finally, the weight of the forward propagation is updated with Equation (8):

$$W = W - \eta_w \Delta W \quad (8)$$

η_w denotes the learning rate of weight.



Inspired by FAs (Lillicrap et al., 2016; Nøkland, 2016), random weights can be used to transmit the error in the network. In this paper, we use the random feedback layer to get the target of the hidden layers. As shown in **Figure 3**, in our model, feedback connections are directly connected from the output layer to the hidden layers, which means that the neural network can update the parameters of all hidden layers simultaneously, and the random feedback connections do not introduce extra computations. The details are shown in Algorithm 1.

4. EXPERIMENTS

In this section, we experimentally evaluate the performance of our model on two benchmark datasets, basic MNIST (LeCun, 1998) and Fashion MNIST (Xiao et al., 2017). The experiments are performed with PyTorch on TITAN RTX. To fully reflect the performance of our algorithm, the fully connected network is considered to carry out the experiment without batch normalization or weight regularization. The update method of the weight is the Stochastic Gradient Descent (SGD) method. In addition, we compare our GLSNN with other state-of-the-art biological plausible methods. The initiation method of the weight is the same as DTP (Lee et al., 2015). Also, the ablation studies are performed to study the effect of the feedback layers. For the parameters of the network, the learning rate for the target $\eta_t = 0.5$, the learning rate for the weight $\eta_w = 0.015$. The batchsize is 10. For the hyper-parameter of the LIF neuron as described in section 3, we set $V_E = 0.2$, $V_I = 0$, $V_L = 0$,

Algorithm 1 The whole learning process of our GLSNN.

```

Require: Initialize a multi-layer neural network with  $L$  layers
Feedforward process in Equation (3)  $F_i$ 
Feedback process in Equation (6)  $G_i$ 
 $t = 0$ , simulation interval  $dt$ , and simulation time  $T$ , max
iteration  $EPO$ 
1: for epoch = 1 to  $EPO$  do
2:   while  $t \leq T$  do
3:     for  $i = 1$  to  $L-1$  do
4:        $S_i = F_i(S_{i-1})$ .
5:        $t = t + dt$ 
6:     end for
7:   end while
8:   Get the prediction error  $e$  with Equation (4).
9:   Get the target of the penultimate layer  $\hat{S}_{L-1}$  with
Equation (5).
10:  for  $i = 1$  to  $L-2$  do
11:     $\hat{S}_i = S_i - G_i(e)$ 
12:  end for
13:  for  $i = 1$  to  $L-1$  do
14:    Update synapse weights Equations (7) and (8)
15:  end for
16: end for
    
```

$V_{th} = 0.0009$, $V_{reset} = 0$, $\tau_m = 0.5$, $\tau_E = 0.2$, $\tau = 0.01$, $g_{leak} = 20$, the simulated time interval $dt = 0.01$, and the total simulation time $T = 0.1$.

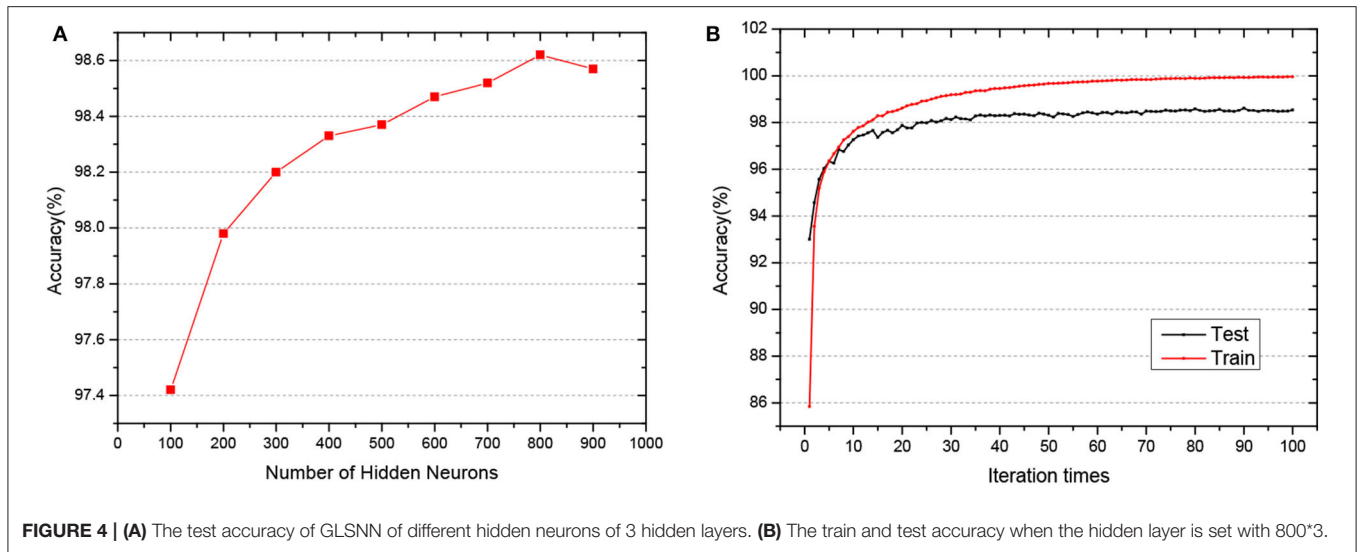


TABLE 1 | Comparison of classification accuracies of GLSNN with other SNN frameworks on the MNIST dataset.

Model	Structure	Neural coding	Learning rule	Acc
SDBN (O'Connor et al., 2013)	FC	Spike	ANN to SNN	94.09
Unsupervised-SNN (Diehl and Cook, 2015)	FC	Spike	STDP	95
LIF-BA (Samadi et al., 2017)	FC	Spike	Broadcast Alignment	97.05
SN (O'Connor and Welling, 2016)	FC	Rate	BP	97.93
Event-SNN (Nefci et al., 2017)	FC	Rate	BP	97.98
Temporal SNN (Mostafa, 2017)	FC	Spike	BP with Temporal Coding	98
SDN (Hunsberger and Eliasmith, 2015)	FC	Spike	ANN to SNN	98.37
VPSNN (Zhang et al., 2018a)	FC	Spike	Equi-prop + STDP	98.52
STCA (Gu et al., 2019)	FC	Spike	Spatial + Temporal Credit Assignment	98.6
GLSNN (This study)	FC	Spike	Global Feedback + STDP	98.62
Balance-SNN (Zhang et al., 2018b)	FC	Spike	Equi-Prop + Multiple Balance Rules	98.64
SCSNN (Wu et al., 2019)	FC	Rate	BP	98.66
BPSNN (Lee et al., 2016)	FC	Rate	BP	98.71
HM2-BP (Jin et al., 2018)	FC	Rate	Macro/Micro level BP	98.88
STBP (Wu et al., 2018)	FC	Rate	Spatial + Temporal BP	98.89

4.1. MNIST

MNIST is the most widely used dataset to measure the performance of the algorithm in machine learning. It consists of 60,000 training samples and 10,000 test samples, used to describe the hand-written digits from 0 to 9. The sample size is 28*28. The number of epochs is set with 100. We wonder how our model fares in this benchmark as the model goes deeper in that target is directed computed from the output layer. To that end, we have trained a network of 3 hidden layers of different hidden neurons to evaluate the performance of the network.

As shown in **Figure 4**, when the network structure is set with [784-800-800-800-10], the test accuracy is the highest at 98.62%. To demonstrate the superiority of our GLSNN, we compare our methods with several different SNN frameworks,

TABLE 2 | The average training time (seconds) per epoch.

500 * 1	500 * 2	500 * 3	500 * 4	500 * 5	500 * 6
80.86	124.99	133.33	178.25	200.08	256.98

as can be seen in **Table 1**, our GLSNN has surpassed all other SNN frameworks trained without BP, such as Unsupervised-SNN (Diehl and Cook, 2015), VPSNN (Zhang et al., 2018a), and so on. Moreover, for the BP trained SNNs, we have exceeded most of them. For the Balance-SNN (Zhang et al., 2018b), in addition to the STDP learning rule, several other rules were introduced, such as LTP, LTD, STF, STD, however only 0.2% accuracy improved compared to our GLSNN. For SCSNN (Wu et al., 2019), BPSNN (Lee et al., 2016), HM2-BP (Jin et al.,

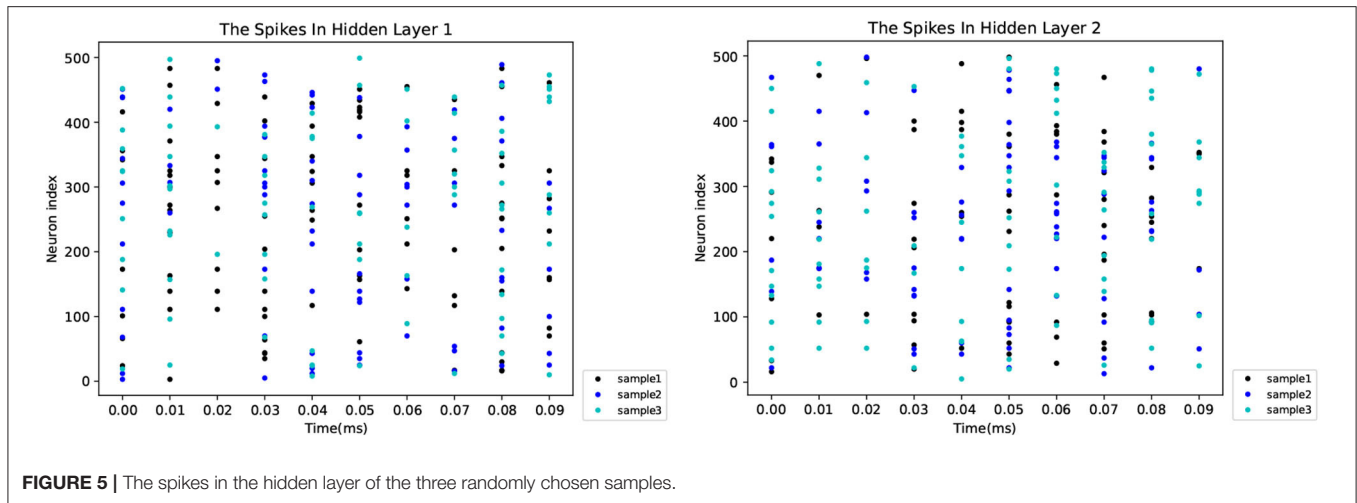


FIGURE 5 | The spikes in the hidden layer of the three randomly chosen samples.

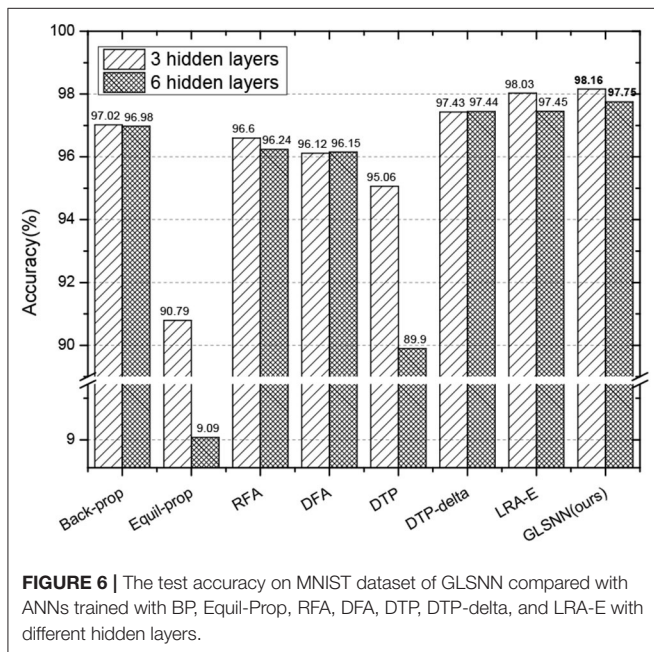


FIGURE 6 | The test accuracy on MNIST dataset of GLSNN compared with ANNs trained with BP, Equil-Prop, RFA, DFA, DTP, DTP-delta, and LRA-E with different hidden layers.

2018), and STBP (Wu et al., 2018), the different levels of backpropagation was connected to contribute to their superior performance, however, which is non-existent in the human brains. To the best of our knowledge, our result could be a new record for the SNNs trained with STDP. The spike transfer process is shown in Figure 5, as the network structure is set with [784-500-500-10].

Also, to prove that our algorithm still performs well when the network is going deeper, we test the results with different hidden layers, whose hidden neurons are set with 256 for consistency with the paper (Ororbia and Mali, 2019). As can be seen in Figure 6, for Equil-prop methods, the accuracy quickly drops down when the network is deeper. Also, the accuracy of the DTP method begins to struggle from 95.06 to 89.9%, which shows the instability of them. Compared with other stable methods, our GLSNN outperforms better

than them both for the five hidden layers and the eight hidden layers, which indicates the stability and superiority of our algorithm.

Also, to measure the computation speed of our model, we test the average runtime per epoch with different hidden layers as shown in Table 2.

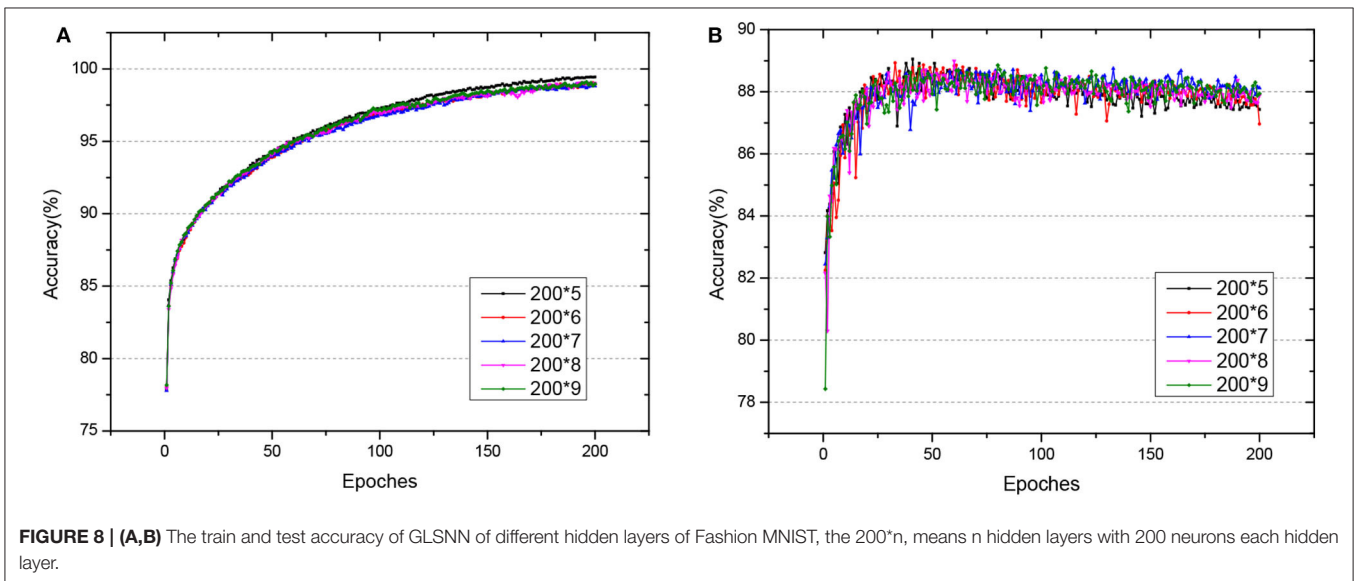
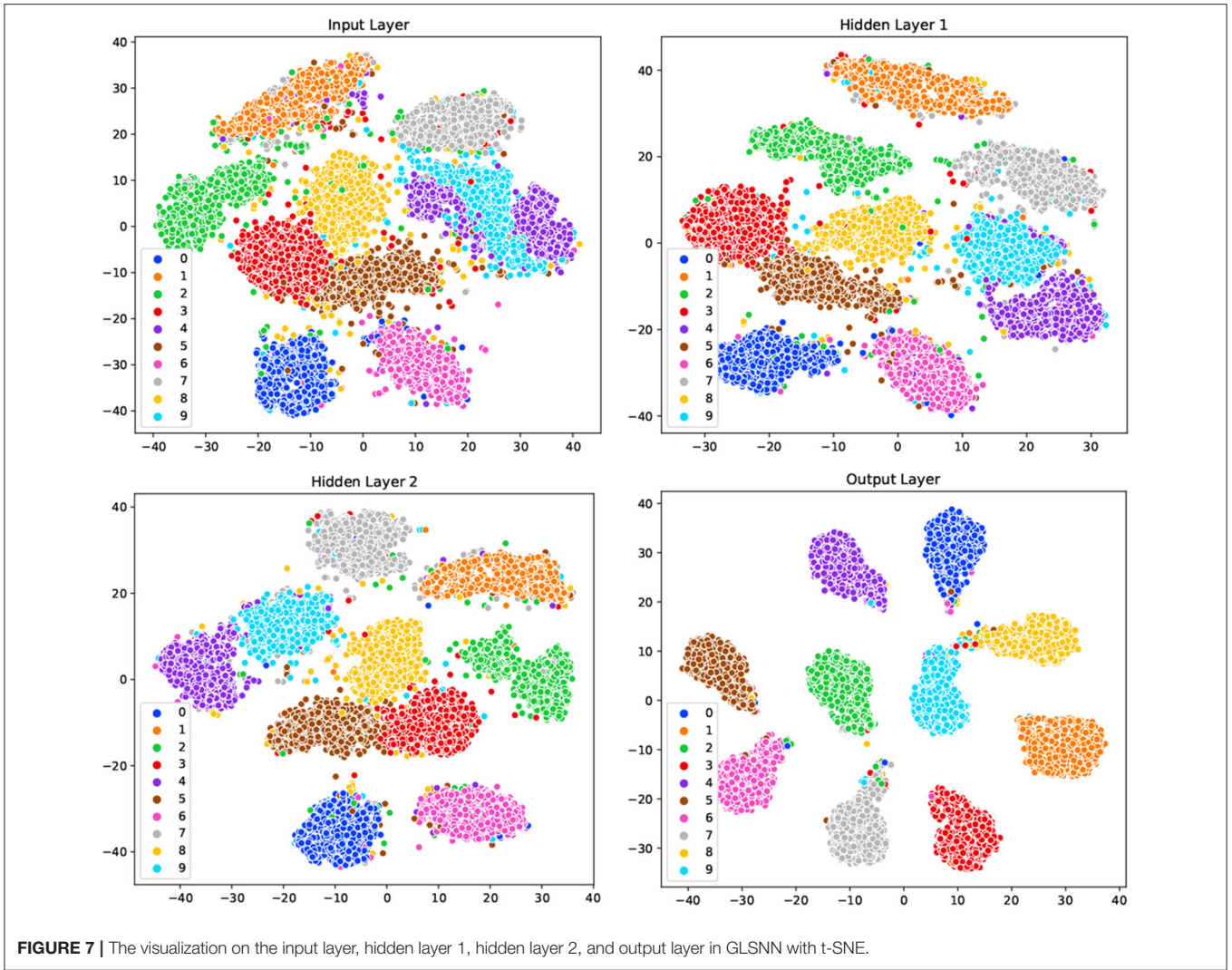
To demonstrate the underlying mechanism of our GLSNN model, the t-SNE method (Maaten and Hinton, 2008) was used to visualize the model’s clustering ability of different layers. The network structure is set with [784-500-500-10], as shown in Figure 7, for the original input, samples of different categories are very close to each other, and some clusters contain samples from other categories. After the training of SNN, the separability of the output information of the hidden layer shows more vital clustering ability than the input layer as the interval between the class clusters is coming larger. For the output layer, different categories are distinguished, which has shown that the learning process of our GLSNN has helped the network to perform better clustering and classification performances.

4.2. Fashion MNIST

Fashion-MNIST is a more complex version compared to MNIST, consisting of gray-scale images of clothing items. Since the dataset is more complicated compared with MNSIT, the training epoch is set with 200, and we tried networks of different hidden layers, as shown in Figure 8. When the network structure is set with five hidden layers of 200 hidden neurons each layer, the network achieves the best performance with 89.05% accuracy on the test dataset. Also, we compare our GLSNN with other biologically plausible methods shown in Table 3. We have chosen the best results of each method as recorded in (Ororbia and Mali, 2019). Our GLSNN exceeds all of them.

4.3. Ablation Studies

To study the effect of the feedback layers of the network, we create four networks with 7, 8, 10, and 12 layers separately. All of the hidden neurons are set with 200. First, we remove all the feedback connections of the network, which means only the weight of the last two-layers could be updated. Then we incrementally add



the feedback layers in the network to see the performance of the network.

As shown in **Figure 9**, with the increase of the number of feedback layers, the performance of the network gradually

TABLE 3 | The test accuracy on the Fashion MNSIT dataset of GLSNN compared with VPSNN and other ANNs trained with BackProp, Equi-Prop, RFA, DFA, DTP, DTP-delta, and LRA-E.

Model	Structure	Type	Performance
VPSNN (Zhang et al., 2018a)	FC	SNN	82.69
Equiprop (Scellier and Bengio, 2017)	FC	ANN	85.99
DTP (Lee et al., 2015)	FC	ANN	86.4
DTP_delta (Ororbia and Mali, 2019)	FC	ANN	87.01
LRA-E (Ororbia and Mali, 2019)	FC	ANN	87.69
RFA (Lillicrap et al., 2016)	FC	ANN	88.01
DFA (Nøkland, 2016)	FC	ANN	88.41
Backprop (Rumelhart et al., 1986)	FC	ANN	88.45
GLSNN (This study)	FC	SNN	89.05

improves. When all the feedback layers are added, the SNN reaches the highest accuracy. The performance of the network did not improve linearly with the increase of the feedback layers. The variation in accuracy can be roughly divided into three steps:

- In step 1, the linear increment of accuracy with weights tuning in only top layers.
- In step 2, the non-increment or stabilization of accuracy with weight tuning in both top and mid-layers.
- In step 3, the prominent increment toward the best accuracy with only adding into the weight tuning in the bottom layer.

The deeper layers play a role in decision-making, while the former layers play a role in feature extraction. That is to say, the feedback connections play a significant role in the perceptual inference, which is consistent with neurophysiology (Harris and Shepherd, 2015).

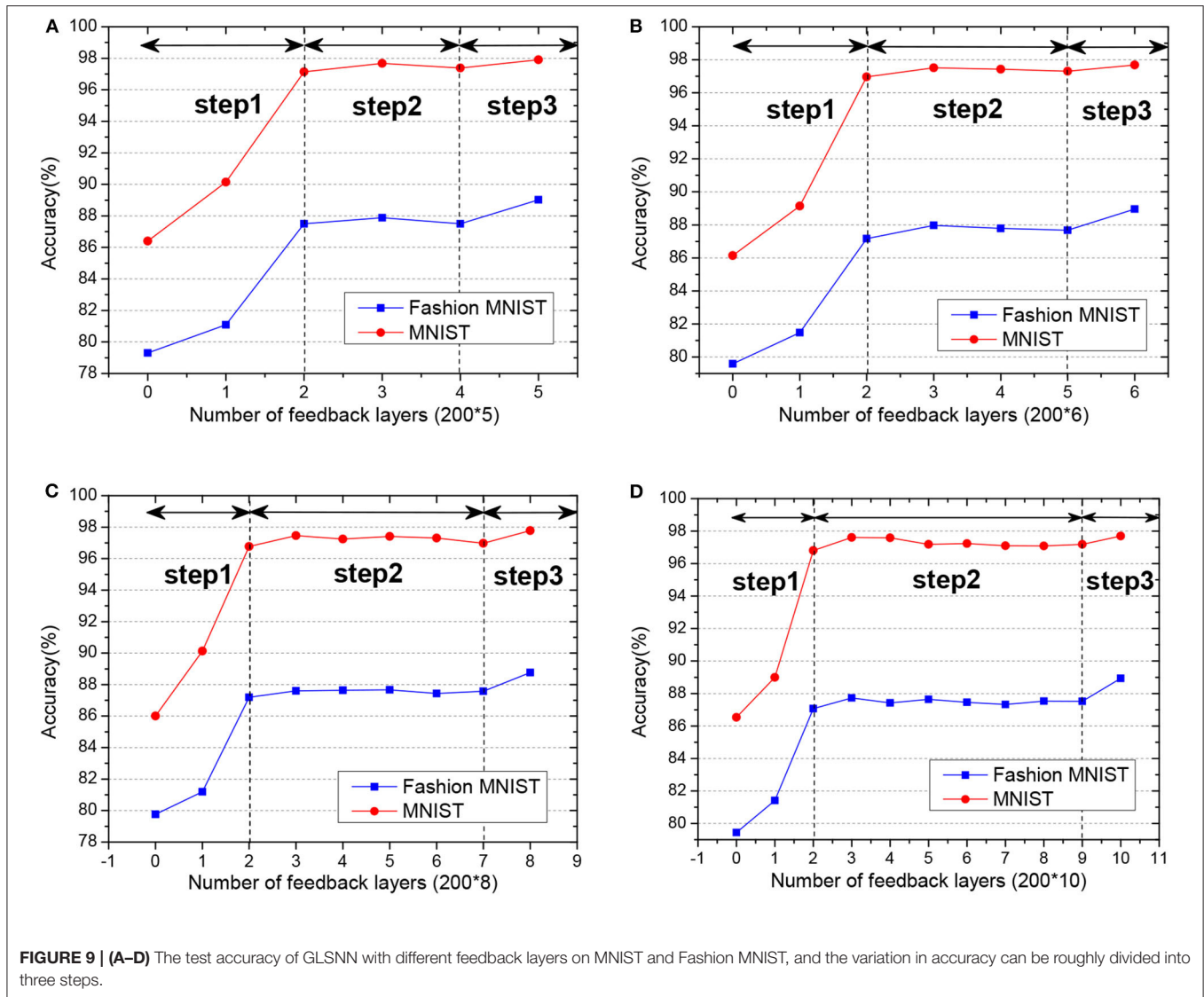


FIGURE 9 | (A–D) The test accuracy of GLSNN with different feedback layers on MNIST and Fashion MNIST, and the variation in accuracy can be roughly divided into three steps.

4.4. Comparison With Other Traditional SNNs Trained With STDP

For the SNNs trained with STDP, the problem is how to introduce global information. The success of the BP algorithm in deep neural networks training is mainly due to the chain rules, which introduce the global error. Traditional SNNs trained with STDP often sidestep this problem, that is they avoid multi-layer training. For Diehl's unsupervised SNN (Diehl and Cook, 2015), only the weight between the input and excitatory neurons is trained with STDP. The extension (Hao et al., 2020) modified the last clustering layer to a supervised classification layer. Masquelier (Masquelier and Thorpe, 2007) introduced a multi-layer SNN combined with convolutional/pooling layer, feature discovery layer and a classification layer. However, the first convolutional layer is set with the Gabor filters, and only the feature discovery layer is trained with STDP. To solve this, Tavanaei (Tavanaei and Maida, 2017) introduced a sparse coding model to replace the handcrafted features in Masquelier and Thorpe (2007). However, the training is layer-wise, the feature discovery layer can only be trained after the first convolutional layer is completed training. Recently, Zhang's work (Zhang et al., 2018a) introduced the equilibrium propagation, the forward and feedback process in SNNs are implicitly defined in the negative and positive phase in equilibrium propagation, which solved the multi-layer training in SNNs to a certain extent. However, due to the implicit definition, when the network went deeper, it becomes hard to converge to a stable situation. Our GLSNN explicitly introduced the global feedback connections, which provides a feasible solution to the training of the multi-layer SNN.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose an SNN training method, which takes full advantage of the global and local plasticity information. We mimic the global feedback connections and the local STDP learning rules in the brain, providing a powerful way to train a multi-layer SNN. The global random feedback connections help to propagate the target from the output layer to the hidden layers. The local STDP learning rule is utilized to optimize the local synaptic strength of the network with the obtained target. Our GLSNN offers an alternative way to solve the weight transpose problem in BP, as well as the feedback layers are directly connected to the hidden layers, leading the weight of

each layer can be directly updated without the error transmitted layer by layer. Experiments indicate that our GLSNN model has performed favorably against several state-of-the-art SNNs on the standard benchmark MNIST and Fashion MNIST dataset.

In terms of future work, the authors intend to study more biologically inspired learning rules in this work, as we only use the STDP local learning rule. The dynamic combination of different learning rules and different types of spiking neurons may further enhance the learning performance of the network. Also, we only verify the performance on the fully connected network structures, in the following work, we would consider more complex network structures such as convolutional neural network and recurrent neural network to accommodate more complex visual perception tasks, such as video object detection and visual tracking.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: <http://yann.lecun.com/exdb/mnist/>; <https://github.com/zalandoresearch/fashion-mnist>.

AUTHOR CONTRIBUTIONS

DZ and YZ designed the study, performed the experiments and the analyses. MS and FZ participated in the biological background discussion and refined the paper. DZ, YZ, and TZ were involved in algorithm discussion, result analysis, and wrote the paper. All authors contributed to the article and approved the submitted version.

FUNDING

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100), the new generation of artificial intelligence major project of the Ministry of Science and Technology of the People's Republic of China (Grant No. 2020AAA0104305), the Beijing Municipal Commission of Science and Technology (Grant No. Z181100001518006), the CETC Joint Fund (Grant No. 6141B08010103), and the Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

- Amit, D. J., Brunel, N., and Tsodyks, M. (1994). Correlations of cortical Hebbian reverberations: theory versus experiment. *J. Neurosci.* 14, 6435–6445. doi: 10.1523/JNEUROSCI.14-11-06435.1994
- Bi, G.-Q., and Poo, M.-M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472. doi: 10.1523/JNEUROSCI.18-24-10464.1998
- Breakspear, M. (2017). Dynamic models of large-scale brain activity. *Nat. Neurosci.* 20, 340–352. doi: 10.1038/nn.4497
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184
- Danelljan, M., Hager, G., Shahbaz Khan, F., and Felsberg, M. (2015). "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (Boston)*, 58–66. doi: 10.1109/ICCVW.2015.84
- Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

- Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney: IEEE), 1–8. doi: 10.1109/IJCNN.2015.7280696
- Durstewitz, D., Seamans, J. K., and Sejnowski, T. J. (2000). Neurocomputational models of working memory. *Nat. Neurosci.* 3, 1184–1191. doi: 10.1038/81460
- Felleman, D. J., and Van, D. E. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1
- Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge, MA: Cambridge University Press. doi: 10.1017/CBO9781107447615
- Gu, P., Xiao, R., Pan, G., and Tang, H. (2019). "STCA: spatio-temporal credit assignment with delayed feedback in deep spiking neural networks," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao: AAAI Press), 1366–1372. doi: 10.24963/ijcai.2019/189
- Hao, Y., Huang, X., Dong, M., and Xu, B. (2020). A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule. *Neural Netw.* 121, 387–395. doi: 10.1016/j.neunet.2019.09.007
- Harris, K. D., and Shepherd, G. M. (2015). The neocortical circuit: themes and variations. *Nat. Neurosci.* 18:170. doi: 10.1038/nn.3917
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas), 770–778. doi: 10.1109/CVPR.2016.90
- Héricé, C., Khalil, R., Mofteh, M., Boraud, T., Guthrie, M., and Garenne, A. (2016). Decision making under uncertainty in a spiking neural network model of the basal ganglia. *J. Integr. Neurosci.* 15, 515–538. doi: 10.1142/S021963521650028X
- Hinton, G. (2007). "How to do backpropagation in a brain," in *Invited Talk at the NIPS'2007 Deep Learning Workshop* (Vancouver).
- Hunsberger, E., and Eliasmith, C. (2015). Spiking deep networks with lif neurons. *arXiv preprint arXiv:1510.08829*.
- Izhikevich, E. M., and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U.S.A.* 105, 3593–3598. doi: 10.1073/pnas.0712231105
- Jin, Y., Zhang, W., and Li, P. (2018). "Hybrid macro/micro level backpropagation for training deep spiking neural networks," in *Advances in Neural Information Processing Systems*, eds S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, and N. Cesa-Bianchi (Montreal: Curran Associates Inc.), 7005–7015.
- Khalil, R., Karim, A. A., Khedr, E., Mofteh, M., and Moustafa, A. A. (2018). Dynamic communications between GABAA switch, local connectivity, and synapses during cortical development: a computational study. *Front. Cell. Neurosci.* 12:468. doi: 10.3389/fncel.2018.00468
- Khalil, R., Mofteh, M. Z., Landry, M., and Moustafa, A. A. (2017a). Models of dynamical synapses and cortical development. *Comput. Models Brain Behav.* 321. doi: 10.1002/9781119159193.ch23
- Khalil, R., Mofteh, M. Z., and Moustafa, A. A. (2017b). The effects of dynamical synapses on firing rate activity: a spiking neural network model. *Eur. J. Neurosci.* 46, 2445–2470. doi: 10.1111/ejn.13712
- Kim, S., Park, S., Na, B., and Yoon, S. (2019). Spiking-yolo: Spiking neural network for real-time object detection. *arXiv preprint arXiv:1903.06530*.
- LeCun, Y. (1998). *The MNIST Database of Handwritten Digits*. Available online at: <http://yann.lecun.com/exdb/mnist/>
- Lee, C., Sarwar, S. S., and Roy, K. (2019). Enabling spike-based backpropagation in state-of-the-art deep neural network architectures. *arXiv preprint arXiv:1903.06379*. doi: 10.3389/fnins.2020.00119
- Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. (2015). "Difference target propagation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Coimbra: Springer), 498–515. doi: 10.1007/978-3-319-23528-8_31
- Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508
- Levina, A., Herrmann, J. M., and Geisel, T. (2007). Dynamical synapses causing self-organized criticality in neural networks. *Nat. Phys.* 3, 857–860. doi: 10.1038/nphys758
- Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7:13276. doi: 10.1038/ncomms13276
- Luo, Y., Xu, M., Yuan, C., Cao, X., Xu, Y., Wang, T., et al. (2020). SiamSNN: spike-based siamese network for energy-efficient and real-time object tracking. *arXiv preprint arXiv:2003.07584*.
- Maaten, L. V. d., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.
- Masquelier, T., and Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3:e31. doi: 10.1371/journal.pcbi.0030031
- Mostafa, H. (2017). Supervised learning based on temporal coding in spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 3227–3235. doi: 10.1109/TNNLS.2017.2726060
- Movellán, J. R. (1991). "Contrastive Hebbian learning in the continuous hopfield model," in *Connectionist Models*, eds D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton (San Mateo, CA: Elsevier), 10–17. doi: 10.1016/B978-1-4832-1448-1.50007-X
- Neftci, E. O., Augustine, C., Paul, S., and Detorakis, G. (2017). Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* 11:324. doi: 10.3389/fnins.2017.00324
- Nøkland, A. (2016). "Direct feedback alignment provides learning in deep neural networks," in *Advances in Neural Information Processing Systems*, eds D. D. Lee, U. von Luxburg, R. Garnett, M. Sugiyama, and I. Guyon (Barcelona: Curran Associates Inc.), 1037–1045.
- O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178
- O'Connor, P., and Welling, M. (2016). Deep spiking networks. *arXiv preprint arXiv:1602.08323*.
- Ororbia, A. G., and Mali, A. (2019). "Biologically motivated algorithms for propagating local target representations," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Honolulu), 4651–4658. doi: 10.1609/aaai.v33i01.33014651
- Potjans, T. C., and Diesmann, M. (2014). The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cereb. Cortex* 24, 785–806. doi: 10.1093/cercor/bhs358
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0
- Samadi, A., Lillicrap, T. P., and Tweed, D. B. (2017). Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Comput.* 29, 578–602. doi: 10.1162/NECO_a_00929
- Scellier, B., and Bengio, Y. (2017). Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 11:24. doi: 10.3389/fncom.2017.00024
- Sporns, O., and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145–162. doi: 10.1385/NI:2:2:145
- Tavanaei, A., and Maida, A. S. (2017). "Multi-layer unsupervised learning in a spiking convolutional neural network," in *2017 International Joint Conference on Neural Networks (IJCNN)* (Anchorage: IEEE), 2023–2030. doi: 10.1109/IJCNN.2017.7966099
- Wu, J., Chua, Y., Zhang, M., Yang, Q., Li, G., and Li, H. (2019). Deep spiking neural network with spike count based learning rule. *arXiv preprint arXiv:1902.05705*. doi: 10.1109/IJCNN.2019.8852380
- Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12:331. doi: 10.3389/fnins.2018.00331
- Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.
- Xie, X., and Seung, H. S. (2000). "Spike-based learning rules and stabilization of persistent neural activity," in *Advances in Neural Information Processing Systems*, eds T. K. Leen, T. Dietterich, and V. Tresp (Denver: MIT Press), 199–208.
- Zenke, F., Agnes, E. J., and Gerstner, W. (2015). Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories

- in spiking neural networks. *Nat. Commun.* 6, 1–13. doi: 10.1038/ncomms7922
- Zhang, T., Zeng, Y., Zhao, D., and Shi, M. (2018a). “A plasticity-centric approach to train the non-differential spiking neural networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence* (New Orleans).
- Zhang, T., Zeng, Y., Zhao, D., and Xu, B. (2018b). “Brain-inspired balanced tuning for spiking neural networks,” in *IJCAI* (Stockholm), 1653–1659. doi: 10.24963/ijcai.2018/229
- Zhao, F., Zeng, Y., and Xu, B. (2018). A brain-inspired decision-making spiking neural network and its application in unmanned aerial vehicle. *Front. Neurobot.* 12:56. doi: 10.3389/fnbot.2018.00056

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

Copyright © 2020 Zhao, Zeng, Zhang, Shi and Zhao. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.