Reinforcement Learning Using Spike-Timing-Dependent Plasticity for Image Recognition

Wei Xie *

Posted Date: 18 September 2025

doi: 10.20944/preprints202509.1555.v1

Keywords: Spiking Neural Networks (SNN); Supervised Learning; Hyperparameter Optimization; Spike-Timing-Dependent Plasticity (STDP); Backpropagation; MNIST

Article

Reinforcement Learning using Spike-Timing-Dependent Plasticity for Image Recognition

Wei Xie

Purdue University, West Lafayette 1; wxie@purdue.edu

Abstract

This study introduces a novel supervised learning approach for spiking neural networks that does not rely on traditional backpropagation. Instead, it employs spike-timing-dependent plasticity (STDP) within a supervised framework for image recognition tasks. The effectiveness of this method is demonstrated using the MNIST dataset. The model achieves approximately 40% learning accuracy with just 10 training stimuli, where each category is exposed to the model only once during training (one-shot learning). With larger training samples, the accuracy increases up to 87%, maintaining negligible ambiguity. Notably, with only 10 hidden neurons, the model reaches 89% accuracy with around 10% ambiguity. This proposed method offers a robust and efficient alternative to traditional backpropagation-based supervised learning techniques.

Keywords: spiking neural networks (SNN); supervised learning; hyperparameter optimization; spike-timing-dependent plasticity (STDP); backpropagation; MNIST

1. Introduction

Image recognition has become ubiquitous, powering applications such as self-driving cars, medical image analysis, and facial recognition systems. Deep learning approaches [1] have achieved remarkable success, but their reliance on extensive computations requires powerful hardware, leading to high energy consumption and hindering deployment in devices with limited resources. Furthermore, deep learning models often lack biological plausibility, limiting their potential to understand neural computation and develop neuromorphic systems that mimic brain-like processing. Spiking neural networks (SNNs) [2] offer a bioinspired alternative to deep learning for image recognition. SNN neurons communicate through discrete spikes, mimicking the information processing of biological neurons. This spiking behavior promises lower power consumption and potentially better reflects the brain's processing capabilities. Spike Timing-Dependent Plasticity (STDP) [3] is a learning rule that allows SNNs to learn by dynamically adjusting synaptic strengths based on the timing of pre- and postsynaptic neuron spiking. This learning mechanism allows SNNs to extract features from the data, which makes them potentially well suited for image recognition tasks.

Various STDP learning rules and SNN architectures have been explored [4]. Unsupervised learning, leveraging STDP, allows SNNs to autonomously acquire selectivity to recurring input patterns without external guidance. Supervised learning in SNNs using backpropagation is hindered by the nondifferentiable nature of spiking neurons and the "weight transport" problem [5]. While significant progress has been made in addressing these challenges, opportunities remain to develop more bioplausible solutions that better capture the complexities of biological systems. Beyond backpropagation, other supervised learning approaches include optimizing the probability of desired output spikes or implementing competitive dynamics among output neurons that represent distinct data classes. In terms of architectural design, researchers have proposed various SNN models, including deep-fully-connected SNNs, spiking convolutional neural networks, and spiking deep belief networks. These

 doi:10.20944/preprints202509.1555.v1

models have demonstrated performance comparable to traditional deep neural networks on a range of tasks.

This paper presents a supervised learning approach for SNNs that integrates gradient-free optimization with STDP for image classification on the MNIST dataset [6]. To enhance class selectivity, each hidden-layer neuron is assigned to a unique group that corresponds to a specific digit class. Synaptic weights are updated only when the target class aligns with the neuron's group, promoting specialization. A unique feature of the proposed approach is the absence of inhibitory synaptic connections between neurons during the training stage. This allows neurons to fully explore the input space, fostering the development of class preferences. Lateral inhibition, introduced only during validation and testing, enables competitive dynamics among neurons, leading to a clean classification based on firing rates. The network architecture is scalable, supporting the construction of deep SNN models for complex tasks. The experimental results on MNIST demonstrate the efficacy of the proposed method in achieving good classification accuracy. The SNN simulations are performed using the Brian2 simulator v2.7.1 [7] on HPC resources provided by the Purdue Rosen Center for Advanced Computing [8]. The software package and optimized hyperparameters for this simulation are provided on GitHub[1].

The remaining sections of this paper are organized as follows. Section 2 presents the neuron and synapse models, as well as the network architecture. Section 3 details the implementation of supervision in the STDP learning rule. Section 4 describes the hyperparameter tuning process. Section 5 evaluates the performance of the network on the MNIST dataset. Section 6 concludes the paper.

2. Network Design

2.1. Modeling Neurons and Synapses

The leaky integrate-and-fire model is chosen as a computational framework to simulate neuronal dynamics. Neurons are simplified as electrical circuits, incorporating capacitance and resistance, while essential features are captured without undue complexity. The membrane potential (V) evolves over time according to the following differential equation:

$$\tau_m \frac{dV}{dt} = (E_{\text{rest}} - V) + g_{\text{e}}(E_{\text{exc}} - V) + g_{\text{i}}(E_{\text{inh}} - V) \tag{1}$$

where E_{rest} represents the resting membrane potential, E_{exc} and E_{inh} denote the equilibrium potentials of the excitatory and inhibitory synapses, respectively, g_{e} and g_{i} correspond to the conductances of the excitatory and inhibitory synapses, and τ_m signifies the time constant of neurons. When a neuron's membrane potential exceeds its threshold, it generates an action potential and subsequently resets its potential to V_{reset}. The threshold, initially set at V_{thres}, is dynamically adjusted based on the neuron's firing rate. During the subsequent refractory period, neurons can not spike again. Synaptic strengths are modulated by changes in conductance. When a presynaptic spike occurs at the synapse, synapses increase their conductance by synaptic weight w. However, when the neuron is not active, the conductance decays exponentially. This dynamic can be mathematically described as

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e, \quad \tau_{g_i} \frac{dg_i}{dt} = -g_i \tag{2}$$

where τ_{g_e} and τ_{g_i} represent the time constant of the excitatory and inhibitory conductance of a postsynaptic neuron, respectively.

Table 1 presents the fixed parameters of the neuron and synapse model. These parameters remain constant for neurons and synapses in different layers of the network. Other parameters, such as τ_m, τ_{g_e}, τ_{g_i} are determined through an optimization process as described in Sect. 4.

[1] "https://github.com/wxie2013/GDFree-supervised-SNN-MNIST"

 doi:10.20944/preprints202509.1555.v1

Table 1. Parameters of hidden-layer neuron and synapse models that remains constant across all layers of a network.

E_{exc}	E_{inh}	E_{rest}	V_{thres}	V_{rest}
0 (mV)	-100 (mV)	-65 (mV)	-52 (mV)	-65 (mV)

2.2. Adaptive Threshold

Balanced neuronal activation is essential for optimal network performance**.** To prevent individual neurons from unduly dominating network responses, an adaptive membrane threshold mechanism, inspired by the work in [9,10], is implemented. The adaptive threshold (V_t) is initialized at V_{thres} and increases by a small value upon each neuronal firing. The small incremental change is calculated as $\Delta V_t \Theta_{vt}$, where Θ_{vt} prevents the threshold of a neuron from reaching too high and stops firing.

$$\Theta_{vt} = \left(0.5 - 0.5 \tanh\left(\left(\frac{-2(V_t - V_{thres}(V_{tshift} - 0.5))}{V_{thres}} + 1\right) / V_{tscale}\right)\right) \tag{3}$$

Here ΔV_t is the maximum potential increase upon each neuronal firing, V_{tshift} represents the point at which Θ_{vt} start to decrease significantly from one, and V_{tscale} regulates the rate of decrease. This formulation prevents excessive threshold elevation, thus maintaining neuronal responsiveness. When a neuron is not active, V_t decreases exponentially as a function of time with a time constant τ_{adpt}, as described by:

$$\tau_{adpt} \frac{dV_t}{dt} = (V_{thres} - V_t) \tag{4}$$

2.3. Network Architecture

The network architecture is a hierarchical feedforward structure composed of an input layer and hidden layers, as illustrated in Figure 1. The input layer comprises 784 neurons, each representing a pixel of a MNIST image. These neurons encode image information through Poisson-distributed spike trains, with firing rates directly proportional to pixel intensities. Each hidden layer is divided into ten groups of neurons, with each group dedicated to recognizing a specific class of digit. A hidden layer unit, as depicted in Figure 1a, consists of ten neurons, one for each digit group. All neurons within a unit share identical spatial coordinates (X, Y) positioned uniformly above the input layer. For this study, a fully connected architecture is employed, where the first hidden layer is fully connected to the input layer, and subsequent hidden layers maintain full connectivity to both the input layer and all preceding layers. The spatial coordinates assigned to neurons offer potential for future enhancements, such as localized connectivity between layers. To enhance digit selectivity and inter-digit competition during testing and validation, inhibitory connections are established between neurons belonging to different digit groups. The firing of a neuron in one group suppresses the activity of neurons in other groups, promoting focused digit representation. To facilitate the development of digit preferences by all neurons during training, these inhibitory connections are absent in the training phase. The image classification is determined by the group exhibiting the highest firing rate in the last hidden layer.

 doi:10.20944/preprints202509.1555.v1

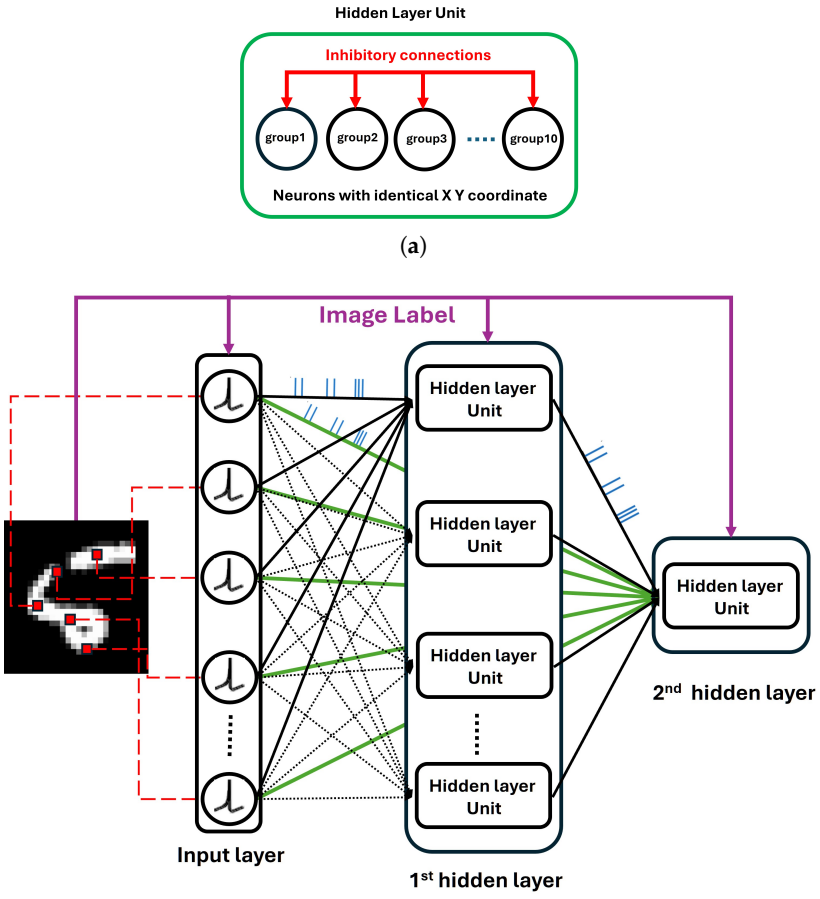

Figure 1. (a) A single hidden layer unit comprising ten neurons, each dedicated to a specific digit between 0 and 9. Neurons in a unit share identical spatial coordinates and are fully connected with inhibitory synapses during validation and testing. (b) An example of a network with two hidden layers. The input layer encodes image information through Poisson-distributed spike trains, with firing rates proportional to pixel intensities. Full connectivity exists between the input layer and the first hidden layer, as well as between the first and second hidden layers. During training, the image label is provided as supervisory information to each excitatory neuron.

3. STDP with Supervised Learning

Excitatory synaptic weights are adjusted using the triplet STDP learning rule [11], known for its superior performance as demonstrated in [12]. According to this learning rule, a presynaptic spike at time t^{pre} introduces a weight change:

$$w(t) \rightarrow w(t) - o_1(t)[A_2^- + A_3^- r_2(t-\varepsilon)] \quad \text{if} \quad t = t^{pre} \tag{5}$$

Conversely, a postsynaptic spike at time t^{post} results in a weight change:

$$w(t) \rightarrow w(t) + r_1(t)[A_2^+ + A_3^+ o_2(t-\varepsilon)] \quad \text{if} \quad t = t^{post} \tag{6}$$

Here, r_1 and r_2 are presynaptic event detectors, while o_1 and o_2 are postsynaptic detectors, each with biological counterparts. A_2^+ and A_2^- quantify the weight changes for pre-post pair or a post-pre pair, respectively. A_3^+ and A_3^- represent the amplitude of the triplet term for potentiation and depression. For simplicity, the minimal model ($A_2^+ = 0$, $A_3^- = 0$), which is intended to fit the visual cortex data, is used for this study.

The supervison is incorporated by modifying eq.(6) as follows:

$$w(t) \rightarrow w(t) + r_1(t) A_3^+ o_2(t-\varepsilon) \Theta_w (1-\beta) \tag{7}$$

 doi:10.20944/preprints202509.1555.v1

where $\beta = 0$ if the postsynaptic neuron group index matches the stimulus label, otherwise $\beta = 1$. The variable Θ_w prevents premature weight saturation, and is defined as:

$$\Theta_w = \left(0.5 - 0.5 \tanh\left(\left(\frac{2(w + w_{max}(w_{shift} - 0.5))}{w_{max}} - 1\right) / w_{scale}\right)\right) \tag{8}$$

with w_{shift} determining the oneset of a significant decrease in Θ_w start from one, and w_{scale} regulates the rate of decrease.

To promote equitable participation of neurons in network dynamics [13], the average weights of a postsynaptic neuron are normalized to λw_{max} with w_{max} being the maximum possible weight. This normalization is achieved by scaling each synaptic weight according to the following formula:

$$w \to w \cdot \frac{\lambda w_{\max}}{w_{\text{tot}} / N_{\text{pre}}} \tag{9}$$

where w_{tot} is the sum of all synaptic weights of a postsynaptic neuron and N_{pre} is the number of presynaptic neurons connected to it. The hyperparameter λ ($0 \leq \lambda \leq 1$) controls the degree of normalization of the weight. This procedure allows synaptic weight to have ample space to grow with a large sample of stimuli and is performed after processing each stimulus.

4. Hyperparameter Optimization

Hyperparameter optimization was conducted using the Hyperopt algorithm [14] within the Ray framework [15]. Each stimulus was presented to the network for 0.5 seconds. The input layer converted images into Poisson spike trains, starting with a mean firing rate proportional to 25% of the original MNIST image intensity. To ensure sufficient spiking activity, the firing rate was adaptively increased by increments of 25% of the intensity until the spike count in each hidden layer reached a minimum of five, or the full intensity was attained. Preliminary experiments indicates that image recognition accuracy is relatively insensitive to this intensity modulation. Table 2 outlines the hyperparameter search ranges for different network configurations. A base network consists of a single hidden layer with one hidden layer unit, while a 2-hidden-layer network comprises two hidden layers.

Hyperparameter optimization was performed on a subset of 10,000 MNIST images for a single epoch and validated with a subset of 1000 MNIST images. For the optimization of a multilayer network, all hyperparameters except for $\lambda_{e\to e}$, $w_{e\to e}(max)$ and max_{delay}, were randomly selected from sets validated to have good performance during base network optimization. The parameters $\lambda_{e\to e}$ and $w_{e\to e}(max)$ can be influenced by the size of the preceding hidden layers, while the max_{delay} between hidden layers may depend on the max_{delay} between hidden layer and input layer. In this study, the sizes of all hidden layers in a multilayer network are fixed at 10 neurons and the optimized $\lambda_{e\to e}$, $w_{e\to e}(max)$ and max_{delay} are thus tailored for this structure. Multiple hyperparameter sets yielded comparable validation performance under these conditions and Table 3 presents one set of optimal hyperparameters identified in this study. It is acknowledged that hyperparameter with better performance may be attainable with access to greater computational resources.

 doi:10.20944/preprints202509.1555.v1

Table 2. Tuning ranges of hyperparameters with different network configurations. Base represent a network with one hidden layer which consists of one hidden layer unit. 2-hidden-layer represents a network with two hidden layers.

Par name	search range (base)	search range (2-hidden-layer)	Description
τ_{adpt}(ms)	$10.0 - 10^8$	N/A	Time constant of adaptive threshold eq.(4).
ΔV_t(mV)	$10^{-3} - 10^{-1}$	N/A	Maximum increment of neuron threshold in eq.(3).
τ_m(ms)	$10.0 - 200.0$	N/A	Time constant of neuron membrane potential.
τ_{g_e}(ms)	$1.0 - 10.0$	N/A	Time constant of the excitatory conductance.
τ_{g_i}(ms)	$1.0 - 10.0$	N/A	Time constant of the inhibitory conductance.
$w_{i\to e}(max)$	$0.1 - 100.0$	N/A	Maximum weight of a synapse between the input and a hidden layer.
$\lambda_{i\to e}$	$10^{-3} - 0.5$	N/A	Scaling factor in eq.(9) for a synaptic weight between the input and a hidden layer
$w_{e\to e}(max)$	N/A	$1.0 - 100.0$	Maximum weight of a synapse between two hidden layers.
$\lambda_{e\to e}$	N/A	$0.1 - 0.5$	Scaling factor in eq.(9) for a synaptic weight between two hidden layers
$\Delta w_{e\leftrightarrow e}$	$10^{-2} - 100.0$	N/A	Weight of the lateral inhibitory synapse among neurons in the same hidden layer.
max_{delay}(ms)	$0.0 - 200.0$	$0.0 - 200.0$	Maximum delay of excitatory synapses.
A_2^-	$10^{-5} - 10^{-2}$	N/A	A_2^- in STDP learning rule eq.(5)
V_{tscale}	$10^{-3} - 1.0$	N/A	V_{tscale} in eq.(3)
V_{tshift}	$0.0 - 1.0$	N/A	V_{tshift} in eq.(3)
w_{scale}	$10^{-3} - 1.0$	N/A	w_{scale} in eq.(8)
w_{shift}	$0.0 - 1.0$	N/A	w_{shift} in eq.(8)

Table 3. A set of hyperparameters demonstrating good validation efficiency, selected from multiple sets with comparable performance, for both base and a 2-hidden-layer networks.

Par name	base	2-hidden-layer ($1^{st}/2^{nd}$)
τ_{adpt}(ms)	10^6	$10^6/10^6$
ΔV_t(mV)	4.4×10^{-3}	$4.0 \times 10^{-4}/3.0 \times 10^{-3}$
τ_m(ms)	200.0	170.0/190.0
τ_{g_e}(ms)	0.4	1.0/0.3
τ_{g_i}(ms)	4.0	3.0/3.0
$w_{i\to e}(max)$	29.0	58.0/72.0
$\lambda_{i\to e}$	0.28	0.34/0.24
$w_{e\to e}(max)$	N/A	100.0
$\lambda_{e\to e}$	N/A	0.15
$max_{delay}^{e\to e}$(ms)	N/A	50.0
$max_{delay}^{i\to e}$(ms)	0.0	10.0/0.0
$\Delta w_{e\leftrightarrow e}$	0.64	1.4/2.1
V_{tscale}	0.18	0.21/0.21
V_{tshift}	0.10	0.40/0.40
w_{scale}	0.23	0.14/0.14
w_{shift}	0.30	0.40/0.40

5. Results and Discussion

The performance of two different network training methods was evaluated using 10^4 testing samples. The direct approach involves training the network on a fixed number of training samples and then testing it as is. The disadvantage of this method is that the training time is proportional to the number of excitatory neurons in the hidden layers. As the number of neurons increases, the training process can become very slow. To overcome this limitation, we leverage the fact that the smallest network in this model, referred to as the base network, consists of only 10 neurons, each responsible

 doi:10.20944/preprints202509.1555.v1

for identifying a single MNIST label, with no lateral excitatory connections between them. During training, a larger network with, for example, one hidden layer containing 250 hidden neurons, can be decomposed into 25 base networks, each processing a different subset of training samples via a single computing process (worker). During testing, the 25 base networks are combined, applying inhibitory connections within each base network among neurons belonging to different groups. Each base network reads in its own respective training synaptic weights and neuron threshold. This approach significantly reduces training time, as each worker process only trains a smaller network. Furthermore, each small network uses a unique set of hyperparameters that have been validated to have good performance during the optimization stage, resulting in a combined network with a diverse set of hyperparameters. This approach is termed "parallel training with diversity".

Figure 2 presents testing accuracy results for networks trained using parallel training with diversity and direct training approaches. The data points of different colors represent the results with different network size and training approaches. The spike counts from all neurons within each neuron group in the last hidden layer are summed, and the group with the maximum spike counts is treated as the prediction of the network. For some stimuli, more than one neuron groups have the same maximum spike counts. In this case, if one of these groups matches the input label, it is counted as a correct identification, albeit with ambiguity. The left panel of the figure illustrates the dependence of the overall testing accuracy on size of the training sample per worker, including those identified with ambiguity. The right panel displays the fraction of correctly identified stimuli with ambiguity, i.e., test accuracy with ambiguity. The result labeled "250 (I)" comes from a network with one hidden layer containing 250 excitatory neurons, trained using parallel training with diversity. The result labeled "250 (II)" is from a network with two hidden layers, each containing 250 excitatory neurons, also trained using parallel training with diversity. In both cases, 25 workers are used during the training stage. The other results are from networks of various sizes trained using the direct training approach. Using the direct approach, the overall accuracy goes from $\sim$ 26% to $\sim$ 89% as the size of the training sample grows from 10 to 5×10^4. Increasing the number of excitatory neurons in the hidden layer from 10 to 250 does not improve the overall test accuracy but reduce ambiguity by more than 10 folds. For smaller training samples, parallel training with diversity significantly outperforms the direct approach. As the training set size increases, their accuracies converge. However, the test accuracy with ambiguity using parallel training with diversity is significantly lower with the same network configuration. Compared to the single hidden layer network (250(I)), adding a second hidden layer with the specified configuration (250(II)) does not enhance the performance. This suggests that the information processed from the first hidden layer is too weak to influence the decision-making of the second layer. Other configurations, such as increasing the number of neurons in the first hidden layer, introducing lateral connections, or localized feedforward synaptic connections, may improve the network performance, as demonstrated for artificial neuron networks.

Figure 3 illustrates the spike counts from each neuron group in response to a specific label in the test sample, obtained through parallel training with diversity when the number of training samples per worker is 10. The network comprises of 250 neurons in the single hidden layer, with 25 workers (denoted as 250(I) in Figure 2). In this setup, each training stimulus is exposed to the network only once on average. Due to the limited number of training samples, the synaptic weights have not fully specialized, leading to similar responses to a given stimulus across all neuron groups. Notably, even with such small differences, the network achieves a $\sim$ 40% testing accuracy with low ambiguity. In contrast, Figure 4 displays the spike count distribution with a significantly larger training sample size per worker (3×10^4). With this extensive training set, the synaptic weights of different neuron groups have become more specialized. Consequently, the spike rate of the correct neuron group is substantially higher than that of other groups, indicating improved discrimination.

Figure 5 illustrates the testing accuracy as a function of input label for various numbers of excitatory neurons in a hidden layer. The left panel displays the overall accuracy, while the right panel shows the accuracy with ambiguity. The number of training samples per worker is 3×10^4. The

 doi:10.20944/preprints202509.1555.v1

color coding is consistent with Figure 2. Notably, the number of neurons in the hidden layer has a minimal impact on the overall test accuracy but significantly affects the test accuracy with ambiguity. For instance, the test accuracy with ambiguity for a network with 250 neurons trained directly is more than 10 times lower than that of a network with 10 neurons. With 250 neurons, the network trained through parallel training with diversity shows an accuracy with ambiguity approximately 2 times lower. The highest overall accuracy is achieved for input label 0, which also has the lowest ambiguity. In contrast, input label 5 yields the lowest overall testing accuracy. The ambiguity increases from label 0 to label 5, then decreases at label 6, and subsequently increases again until label 9.

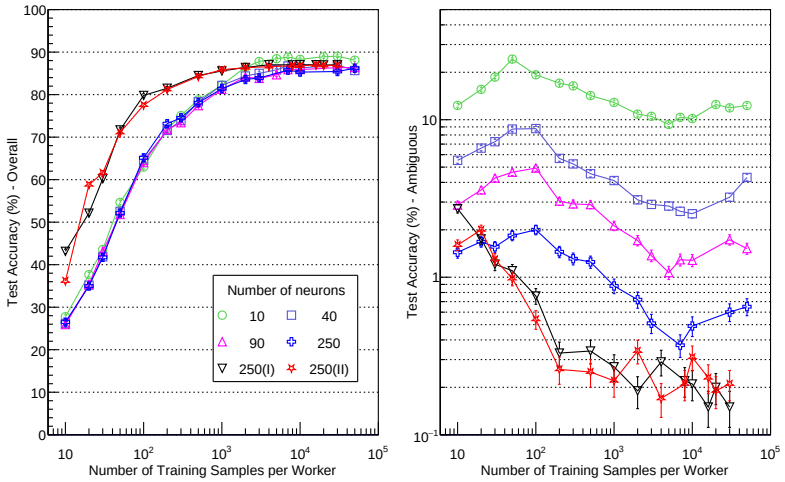

Figure 2. The influence of training sample size per worker on (left) overall test accuracy and (right) test accuracy with ambiguity for networks with different hidden layer sizes. Each color represents a specific number of excitatory neurons in the hidden layer. The size of test sample is 10^4 and the bars associated with data points represent statistical uncertainties.

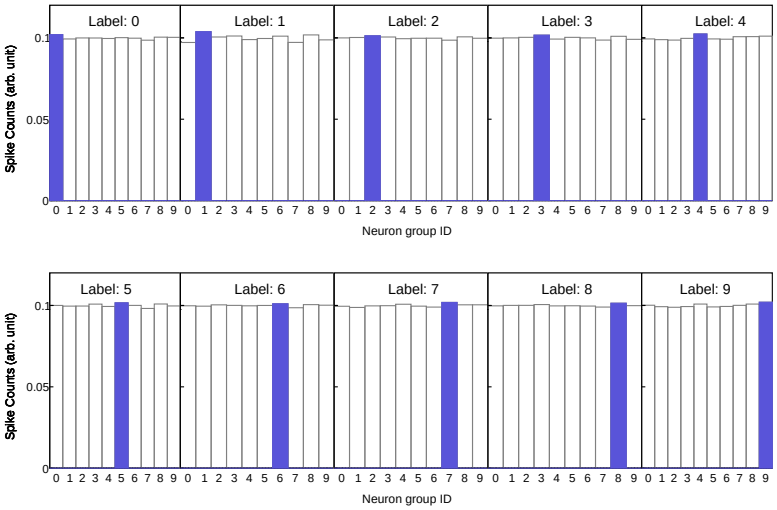

Figure 3. Spike count distribution of each neuron group for correctly identified test samples across different input labels (highlighted in blue). The network has a single hidden layer with 250 neurons (equivalent to 25 workers) employing parallel training with diversity. Each worker processed 10 training stimuli. The size of test sample is 10^4.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 September 2025 doi:10.20944/preprints202509.1555.v1

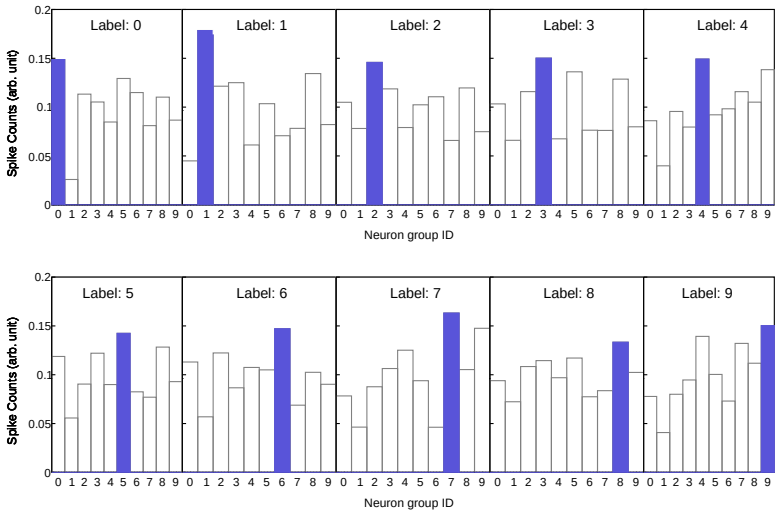

Figure 4. Spike count distribution of each neuron group for correctly identified test samples across different input labels (highlighted in blue). The network has a single hidden layer with 250 neurons (equivalent to 25 workers) employing parallel training with diversity. Each worker processed 10^4 training samples. The size of test sample is 10^4.

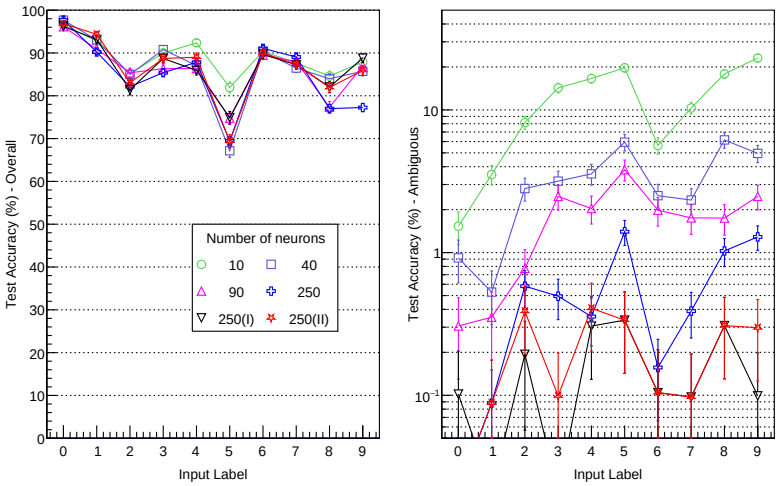

Figure 5. The influence of input labels on (left) overall test accuracy and (right) test accuracy with ambiguity for networks with different hidden layer sizes. Each color represents a specific number of excitatory neurons in the hidden layer. The size of test sample is 3×10^4 and the bars associated with data points represent statistical uncertainties.

Figure 6 displays the distribution of incorrect predictions for various input labels using a 1-hidden layer network with 250 neurons (10 neurons per worker) trained through parallel training with diversity. Each worker received 3×10^4 training stimuli, and the test set consisted of 10^4 samples. Several notable patterns emerge. For instance, the network most frequently confuses labels 4 and 9, as well as labels 3 and 5. Additionally, label 1 is often mispredicted as label 3 or 8, Label 6 is frequently misclassified as label 0 and 5, while label 7 is often mispredicted as label 9 and 2.

 doi:10.20944/preprints202509.1555.v1

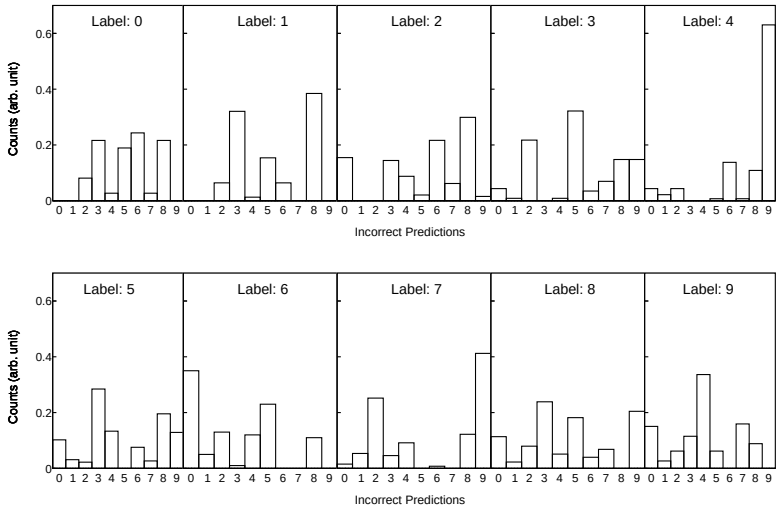

Figure 6. Distribution of incorrect predictions across different input labels. The network has a single hidden layer with 250 neurons (equivalent to 25 workers) employing parallel training with diversity. Each worker processed 3×10^4 training samples. The size of test sample is 10^4.

6. Conclusions

This study introduces a supervised learning approach for spiking neural networks that leverages spike-timing-dependent plasticity within a supervised framework for image recognition tasks, thereby eliminating the reliance on traditional backpropagation. The MNIST dataset effectively demonstrates the strengths of this approach. Using only 10 hidden neurons, the model obtains an 89% accuracy rate with approximately 10% ambiguity. When the model includes a larger number of hidden neurons and utilizes a diverse set of hyperparameters among them, it attains around 40% learning accuracy with just 10 training stimuli, where each category is presented only once during training (one-shot learning). As the training sample size increases, the accuracy improves to 87%, maintaining negligible ambiguity. Additionally, a parallel training and testing method is proposed, leveraging small networks per computing process, which significantly reduces the training time and increases the testing accuracy with small training samples. Future refinements, such as increasing the number of neurons in the first hidden layer, introducing excitatory lateral connections, or confining the region in feedforward synaptic connections, may further improve the network performance.

Funding: This research received no external funding.

Acknowledgments: The author would like to express sincere gratitude to Mark Linvill, Chris Orr, and the Department of Physics and Astronomy for facilitating access to HPC resources at the Purdue Rosen Center for Advanced Computing (RCAC). Thanks are also extended to the research service team at RCAC for their timely support. Moreover, the author would like to thank Marcel Stimberg and the Brian2 team for their invaluable discussions and assistance.

Data Availability Statement: The software package and optimized hyperparameters for this simulation are provided on GitHub at "https://github.com/wxie2013/GDFree-supervised-SNN-MNIST".

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SNN	Spiking neural networks
STDP	Spike Timing-Dependent Plasticity
MNIST	Database of Handwritten Digit Images for Machine Learning Research

 doi:10.20944/preprints202509.1555.v1

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444.
2. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks* **1997**, *10*, 1659–1671.
3. Abbott, L.F.; Nelson, S.B. Synaptic plasticity: taming the beast. *Nature Neuroscience* **2000**, *3*, 1178–1183.
4. Tavanaei, A.; Ghodrati, M.; Kheradpisheh, S.R.; Masquelier, T.; Maida, A. Deep learning in spiking neural networks. *Neural Networks* **2019**, *111*, 47–63.
5. Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science* **1987**, *11*, 23–63.
6. Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* **2012**, *29*, 141–142.
7. Stimberg, M.; Brette, R.; Goodman, D.F. Brian 2, an intuitive and efficient neural simulator. *eLife* **2019**, *8*, e47314. https://doi.org/10.7554/eLife.47314.
8. McCartney, G.; Hacker, T.; Yang, B. Empowering Faculty: A Campus Cyberinfrastructure Strategy for Research Communities. *Educause Review* **2014**, *1*.
9. Zhang, W.; Linden, D. The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience* **2003**, *4*, 885–900.
10. Querlioz, D.; Bichler, O. Dollfus, P.; Gamrat, C. Immunity to device variations in a spiking neural network with memristive nanodevices. *Nanotechnol. IEEE Trans.* **2013**, *12*, 288–295.
11. Pfister, J.P.; Gerstner, W. Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity. *The Journal of Neuroscience* **2006**, *26*, 9673–9682.
12. Diehl, P.U.; Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience.* **2015**, *9*, 1–9.
13. Goodhill, G.J.; Barrow, H.G. The role of weight normalization in competitive learning. *Neural Computation* **1994**, *6*, 255–269.
14. Bergstra, J. Komer, B.E.C.Y.D.; Cox, D.D. Hyperopt: a Python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **2015**, *8*, 014008.
15. Liaw.; Liang, R.; Eric.; Nishihara.; Robert.; Moritz, P.; Gonzalez.; E, J.; Stoica.; Ion. Tune: A Research Platform for Distributed Model Selection and Training. *arXiv:1807.05118 [cs.LG]* **2018**, *1*.