

**FAST AND LOW-PRECISION LEARNING IN GPU-ACCELERATED SPIKING
NEURAL NETWORK**

A Dissertation
Presented to
The Academic Faculty

By

Xueyuan She

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2020

Copyright © Xueyuan She 2020

**FAST AND LOW-PRECISION LEARNING IN GPU-ACCELERATED SPIKING
NEURAL NETWORK**

Approved by:

Dr. Saibal Mukhopadhyay, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
School of Computer Science
Georgia Institute of Technology

Dr. Tushar Krishna
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Date Approved: July 23, 2020

ACKNOWLEDGEMENTS

First, I would like to thank my parents. I would not be where I am without them. I also want to thank my advisor Dr. Saibal Mukhopadhyay for the support and mentorship. I would like to thank Dr. and Dr. for their help on this thesis. Last but not least, I would like to thank Yun Long and other members in my lab for helping me on work and life throughout the years.

TABLE OF CONTENTS

Acknowledgments	iii
List of Tables	vi
List of Figures	vii
Chapter 1: Introduction	1
Chapter 2: Background	3
2.1 Spiking Neural Network	3
2.2 SNN Simulators	4
Chapter 3: Technical Approach	5
3.1 Spiking Neuron Model	5
3.2 Synapse Model	5
3.3 Stochastic behavior of synapses	7
Chapter 4: Experimental Platform	9
4.1 Design of the Simulator	9
4.2 Network Architecture and Configuration	10
4.3 Low precision learning and rounding options	12
4.4 Parameters and other details	13

Chapter 5: Results	14
5.1 Accuracy Comparison with Existing Simulators	14
5.2 Improved Learning Accuracy with Stochastic STDP	15
5.3 Fast Learning with Higher Input Frequency	16
5.4 Low precision learning	20
5.5 Summary of Results	21
Chapter 6: Conclusion	22
References	24

LIST OF TABLES

4.1	Parameters for different learning options	12
5.1	Accuracy results (%) for rounding options	20

LIST OF FIGURES

3.1	Neuron models: (a) Spiking frequency vs. input current of LIF neurons, (b) spiking behavior (c) synaptic behavior under stochastic STDP, (d) Conversion of input image to spike trains;	6
3.2	(a) An pre-synaptic and post-synaptic neuron connected by a synapse; and spike timing of input spike and back-propagated spike from post-synaptic neuron. The specific timing of the back-propagated spike induce an LTP behavior. (b) Flow chart of STDP algorithm.	7
4.1	ParallelSpikeSim: a GPU accelerated SNN simulator with stochastic STDP, low precision learning and frequency control module	9
4.2	Network architecture of the SNN implemented in this work.	10
5.1	Simulation of spiking activity and performance	14
5.2	Conductance distribution of SNN with (i) stochastic STDP learning MNIST, (ii) deterministic STDP learning MNIST, (iii) stochastic STDP learning fashion MNIST and (iv) deterministic STDP learning fashion MNIST; x axis is conductance value and y axis is the count	15
5.3	Visualization of synapse conductance (a) Baseline and stochastic STDP for MNIST and Fashion MNIST, and (b) effect of input spike train frequency on stochastic STDP.	16
5.4	High-frequency and low-precision operations (a) Input spike trains at low (left) and high (right) frequencies (each dot represents one spike). (b) Conductance distribution of Q1.7 precision (MNIST) with stochastic (top) and deterministic STDP (bottom).	17
5.5	High-frequency learning (a) Accuracy loss vs. max input frequency, and (b) Accuracy vs. run-time.	18

5.6 Comparison of learning configurations: (a) conductance map, (b) Accuracy and run-time, and (c) accuracy loss vs. simulation time. 19

SUMMARY

Spiking neural network (SNN) uses biologically inspired neuron model coupled with Spike-timing-dependent-plasticity (STDP) to enable unsupervised continuous learning in artificial intelligence (AI) platform. However, current SNN algorithms shows low accuracy in complex problems and are hard to operate at reduced precision. This paper demonstrates a GPU-accelerated SNN architecture that uses stochasticity in the STDP coupled with higher frequency input spike trains. The simulation results demonstrate 2 to 3 times faster learning compared to deterministic SNN architectures while maintaining high accuracy for MNIST (simple) and fashion MNIST (complex) data sets. Further, we show stochastic STDP enables learning even with 2 bits of operation, while deterministic STDP fails.

CHAPTER 1

INTRODUCTION

Spiking neural network (SNN) are artificial neural networks (ANN) with biological plausible neuron and synapse models. This type of network has drawn significant attention in the field of artificial intelligence. In particular, SNN demonstrates the ability of unsupervised learning using Spike-Time-Dependent-Plasticity (STDP) as the synapse model. The STDP is a phenomenon observed in biology experiments[1, 2], which can be used as the algorithm to change synapse conductance inside SNN. With STDP, conductance is modulated based on the time difference between pre-synaptic and post-synaptic spikes, enabling the learning ability of spiking neurons.

Along with the development of learning algorithms, simulations of SNN has received significant attention in recent years to facilitate both understanding brain and developing AI algorithms. The existing SNN simulators such as NEST and CARLSim use deterministic STDP learning which suffers from several drawbacks, as shown in result chapter for details. First, while networks demonstrate good accuracy for simple tasks such as MNIST-based digit recognition [3], the learning accuracy for difficult tasks such as Fashion-MNIST[4] (images of apparel items that contains complex features) is much lower. Second, deterministic STDP provides limited opportunity for fast and low-precision simulation of unsupervised learning in SNN. For example, deterministic STDP for MNIST performed in 8-bit fixed-point (28%) shows significantly lower accuracy than floating-point (92%). This paper presents a GPU-accelerated SNN simulator, ParallelSpikeSim for high-accuracy, fast, and low-precision unsupervised learning. The key innovation of this paper is to demonstrate stochastic STDP for unsupervised learning in SNN, instead of well-explored deterministic STDP algorithms [5][6] used in prior simulators. Moreover, we provide controllability to precision (down to 2-bit) with different rounding options, and frequency of input spike

trains during unsupervised learning (and inference) in SNN to effectively exploit stochastic STDP for fast and low-precision learning. This paper makes following key contributions:

- We present the ParallelSpikeSim as a GPU-accelerated SNN simulator supporting unsupervised learning. The simulator is designed for parallel computing, programmed in C++ using CUDA libraries, and support different neuron/synaptic models.
- We demonstrate that the stochastic STDP allows good learning accuracy for both simple (MNIST[3], 96.1% accuracy) and complex (feature-rich) (Fashion-MNIST [4], 77.2% accuracy) data sets.
- We show that the stochastic STDP allows SNN to operate under input frequency ranges much higher than that of deterministic STDP design, and hence, enables up to 3x lower learning time with graceful quality degradation.
- We show that stochastic STDP enables robust learning from 32-bit floating-point (accuracy 96.1%) down to 2-bit fixed-point (accuracy 64.6%) learning, whereas deterministic STDP based SNN fails to provide meaningful results (accuracy drops from 92.2% to 9.6%).

CHAPTER 2

BACKGROUND

2.1 Spiking Neural Network

SNN has been actively researched for the better understanding of biological neural systems, and for its potential to achieve learning ability parallel to human brain, which surpass the state-of-the-art CNNs in many performance criteria as well as power efficiency [7]. Neuron models used in SNN mimic the spiking behavior of biological neurons. When implementing spiking neural networks, there are different neuron models that achieve varying levels of biological significance (in terms of accuracy in tracing membrane potential value) based on how many differential equations are used. Long [8] shows in his work that it is possible to achieve similar spiking activities in different neuron models operating under a specific range of input current magnitude. This indicates that an event based network such as the one presented in this work, which uses spikes rather than membrane potential to encode information, can receive benefit in terms of simulation speed from using a mathematically less complex model.

STDP is an unsupervised learning algorithm that modulates neuron connection strength with the temporal information of spiking events. The learning rule originates from the temporal synaptic modification rule observed in hippocampus [9]. A more detailed theoretical work by Gerstner explains learning from the perspective of spatial-temporal patterns of spikes [10] and the algorithm is later named spike-timing-dependent-plasticity (STDP). Since then multiple experimental evidence of STDP has been observed [11][12][13][14], making STDP a biologically plausible algorithm that is suitable for the purpose of this work. Several research efforts have explored STDP algorithms as the main learning algorithm in SNN [5][6][15].

2.2 SNN Simulators

Due the parallel nature of neural activities in biological neural systems, an ideal design of SNN simulator requires a high level of parallelism. Early parallel SNN simulation tools like pGENESIS[16] required cluster computers to run. The recent developments focused on achieving SNN simulation with higher accuracy and better simulation speed[17]. In particular, advancements of Graphics Processing Units (GPUs) has led to feasibility of orders of magnitude improvement in computing speed of SNNs. Several SNN simulators with integrated STDP learning have been presented such as Brain[18], NEST[19] and CARLsim[20]. In a recent work, Long [8] has presented a region-of-interest (ROI) based approach that vary complexity of neuron models based on spiking activity in a region, but they did not discuss stochastic STDP-based learning using SNN.

CHAPTER 3

TECHNICAL APPROACH

3.1 Spiking Neuron Model

The spiking neuron model used in this work is leaky integrate-and-fire (LIF). For LIF model, membrane potential of a neuron is described by:

$$dv/dt = a + bv + cI \quad (3.1)$$

$$v = v_{reset}, \text{ if } v > v_{threshold} \quad (3.2)$$

The current I received by neuron a is described by:

$$I_a = \sum_{n=0}^N g_{n,a} v_{pre_n} \quad (3.3)$$

N is the total number of pre-neurons connected to neuron a . $g_{n,a}$ is the conductance of the synapse connecting neuron a and its pre-neuron n . v_{pre_n} is the voltage spike of pre-neuron n . Fig. 3.1 (a) shows change of spiking frequency of LIF model with parameters used in this work with respect to different input current.

3.2 Synapse Model

In SNN, a pair of two neurons (pre-synaptic neuron and post-synaptic neuron) can be connected by a synapse, as shown in Fig. 3.2(a). When the pre-synaptic neuron spikes, current signal is sent through the synapse to the post-synaptic neuron. Conductance of synapse determines how strongly two neurons are connected. In context of machine learning, synapse conductance can be thought of as the connection weight between neurons and learning is

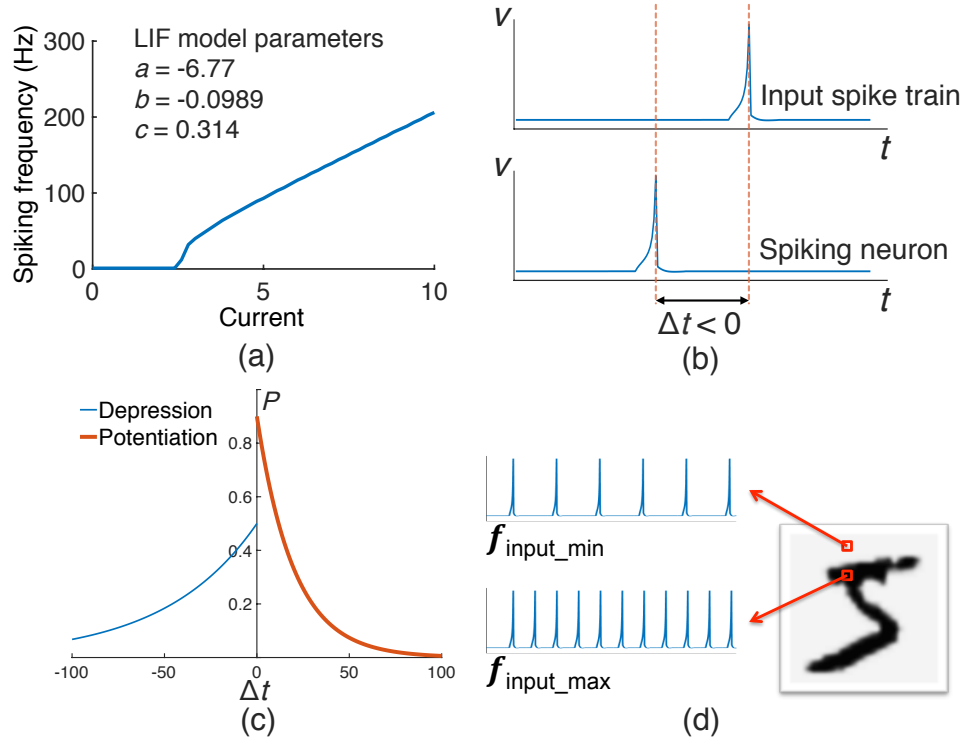


Figure 3.1: Neuron models: (a) Spiking frequency vs. input current of LIF neurons, (b) spiking behavior (c) synaptic behavior under stochastic STDP, (d) Conversion of input image to spike trains;

achieved through modulating the conductance.

With STDP learning rule integrated, the network is able to extract the causality between spikes of two connected neurons from their temporal relationship. As a result, the SNN can perform conductance update without using label of the input data. More specifically, there are two operations of STDP: long-term potentiation (LTP) and long-term depression (LTD). LTP is triggered when post-synaptic neuron spikes closely after a pre-synaptic neuron spike, indicating a causal relationship between the two events, and the conductance of the synapse is increased. On the other hand, when a post-synaptic neuron spikes before pre-synaptic spike arrives or without receiving a pre-synaptic spike at all, the synapse goes through LTD which decreases its conductance. An example of LTP is shown in Fig. 3.2(a).

We choose to use STDP model presented by Querlioz [6], which has been tested in vision based SNN applications. The model is described by the following equations:

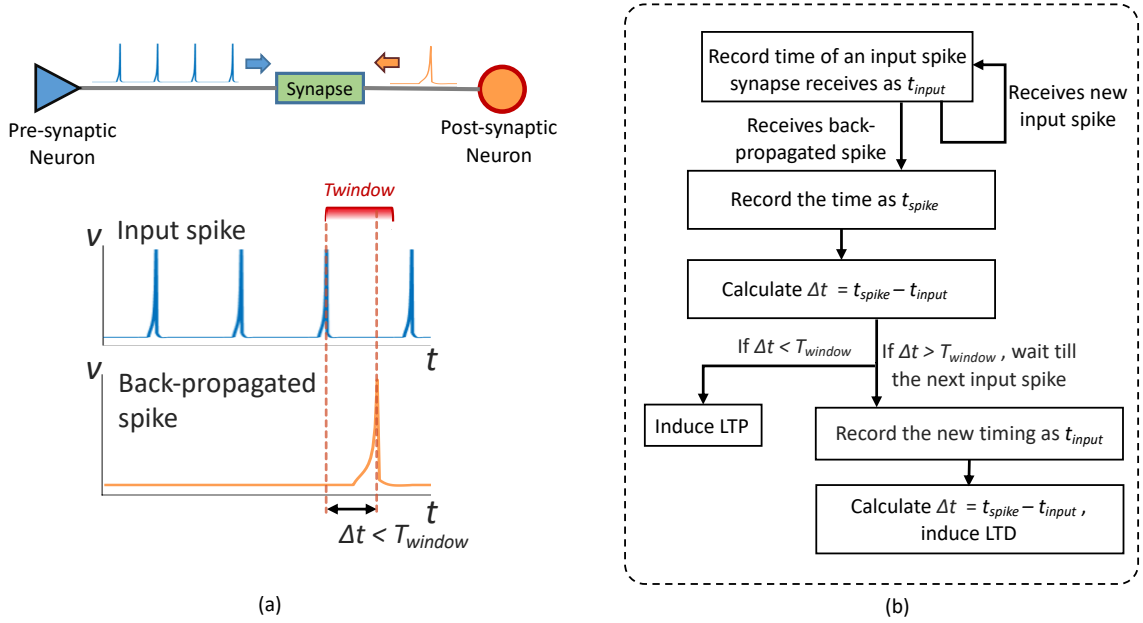


Figure 3.2: (a) An pre-synaptic and post-synaptic neuron connected by a synapse; and spike timing of input spike and back-propagated spike from post-synaptic neuron. The specific timing of the back-propagated spike induce an LTP behavior. (b) Flow chart of STDP algorithm.

$$\Delta G_p = \alpha_p e^{-\beta_p(G-G_{min})/(G_{max}-G_{min})} \quad (3.4)$$

$$\Delta G_d = \alpha_d e^{-\beta_d(G_{max}-G)/(G_{max}-G_{min})} \quad (3.5)$$

ΔG_p is the magnitude of LTP actions, and ΔG_d is the magnitude of LTD actions. α_p , α_d , β_p , β_d , G_{max} and G_{min} are parameters that are tuned based on other network configurations such as input matrix size, input spiking frequency and voltage. A illustration of the STDP process is shown in Fig. 3.2(b).

3.3 Stochastic behavior of synapses

For synapses with stochastic STDP behavior, potentiation or depression of synapses is not deterministic, but has a probability that depends on the time difference of the two spike events that initiates the modulation of conductance. For instance, as shown in Fig. 3.1 (b),

Δt is below zero when spiking neuron spikes before a spike from input train arrives at the synapse. Stochastic STDP is achieved with an algorithm inspired by the work of Srinivasan [21]. The probabilities for potentiation and depression are defined by:

$$P_{pot} = \gamma_{pot} e^{(-\Delta t / (\tau_{pot}))} \quad (3.6)$$

$$P_{dep} = \gamma_{dep} e^{(\Delta t / (\tau_{dep}))} \quad (3.7)$$

The probabilities are exponentially related to time difference, with maximum value controlled by γ_{pot} and γ_{dep} , as shown in Fig. 3.1 (c). In the event of potentiation, the probability is higher when Δt is smaller, indicating a stronger causal relationship. As for depression, the probability is higher when Δt is larger. This stochastic STDP algorithm captures the exponential dependence on timing of synaptic behavior as observed in biological experiments [22].

CHAPTER 4

EXPERIMENTAL PLATFORM

4.1 Design of the Simulator

Fig. 4.1 shows the flowchart of the unsupervised learning architecture with SNN achieved with ParallelSpikeSim. The SNN simulator has two major components. First, a spiking neuron simulator to simulate the differential equations governing the neuron dynamics (i.e. equations 3.1,3.2,3.3) for a given synapse conductance. The second component is the learning module that implement the synaptic models and allow synapse conductance to be updated based on the spiking activity using STDP rules (i.e. equations 3.4,3.5). Many past SNN simulators (e.g. CARLSim, BRAIN, NEST, etc.) includes the STDP based learning modules. The key innovation in ParallelSpikeSim is to augment the learning modules to include stochastic STDP (i.e. equations 3.6,3.7) and various precision control and rounding options (see Section 4.3). Moreover, we introduce an additional module between input images and spiking neuron simulator that allows controlling the frequency of the input spike train as shown in Fig. 4.1.

CPU serves as data I/O and controls data flow of GPU. It constructs the simulation environment with configuration and input data file, allocate memory and transfer data in unified data structures to GPU memory when simulation starts. The unified data structures of ParallelSpikeSim encapsulate all network information into the network object and all

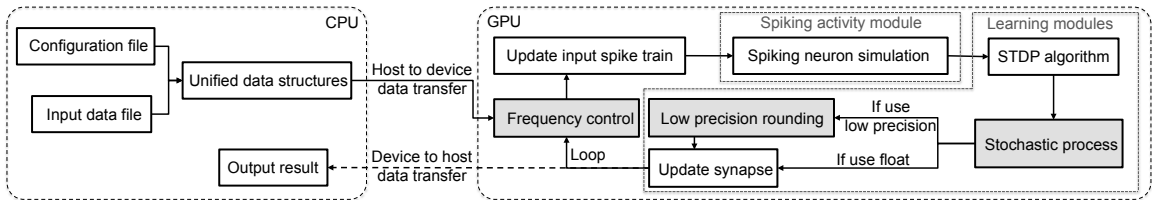


Figure 4.1: ParallelSpikeSim: a GPU accelerated SNN simulator with stochastic STDP, low precision learning and frequency control module

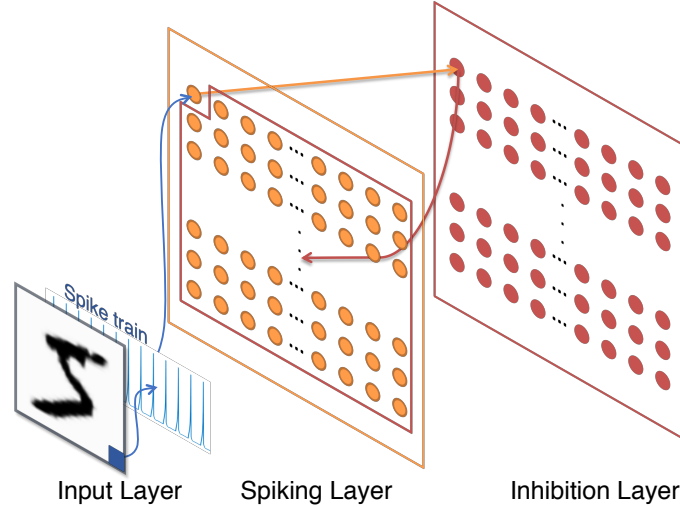


Figure 4.2: Network architecture of the SNN implemented in this work.

input into the data object, to facilitate swift addition of functionality and customization of network hierarchy, layer connectivity and behavior of each synapse and neuron. After initialization, simulation of spiking neurons runs in parallel on GPU threads. Stochastic STDP module uses spike timers to track the temporal relationship between pre-synaptic and post-synaptic spikes, and performs stochastic process on-board the GPU to leverage the fast CUDA random number generator. Low precision learning module operate with reduced bit-width down to 2 bits, and has three available rounding options: bit truncation, rounding to nearest and stochastic rounding. Frequency control module works in two phases: frequency boost and learning time reduction. More details about the impact of the three modules are discussed in the following sections.

4.2 Network Architecture and Configuration

We implement the SNN architecture shown in Fig. 4.2 to demonstrate the proposed simulator. It consists of three layers: input layer, spiking neuron layer and lateral inhibition layer.

First layer is the input layer. The 28x28 input units in the first layer, each corresponds to one pixel of the image from MNIST dataset, function as spike train generators. During

learning process, the 8-bit pixel intensity from the input data is converted into spiking frequency over a range from f_{min} and f_{max} . The relationship is direct proportional, i.e. darker pixels have higher spiking frequency. Each input unit keeps track of the corresponding spiking frequency and during learning period t_{learn} of one image, constantly sends excitatory spike signals to the next layer at such frequency. With a specific setting of f_{min} and f_{max} , t_{learn} needs to be at a proper value for the network generate enough spikes that distinguish the pattern of each input image.

The input layer connects to the spiking neuron layer (second layer) in an all-to-all fashion (fully connected). Conductance of each synapse connecting input to first layer neurons collectively forms a conductance array that learns to recognize a specific pattern. The inhibition layer (third layer) has the same dimension as the second layer. For a neuron in the third layer at location $\langle i, j \rangle$, there is only one input connection to it which is from the neuron at the same location, i.e. $\langle i, j \rangle$, in the second layer, and its output connects to all second layer neurons except for $\langle i, j \rangle$.

As a result, when one neuron in the second layer spikes, it sends excitatory signal to the one corresponding neuron in the third layer. The inhibitory neuron has low threshold that it activates immediately after receiving one excitatory signal, and then sends inhibitory signal to all other neurons in the second layer for a period of time t_{inh} . Membrane potential of neurons that receive the inhibitory signal is decreased by a value of v_{inh} , and can not spike during t_{inh} . It is worth noting that this membrane potential is not set to v_{reset} for inhibition as adopted in some other SNN designs. The reason is that subtraction can better maintain information about the original membrane potential level before the inhibition, and leads to better learning performance. With the inhibition layer implemented, the network achieves a winner-take-all principle throughout the spiking neuron layer. This prevents multiple neurons from learning the same pattern.

The causal relationship between pre-synapse and post-synapse neurons explored by STDP algorithm makes it possible for the network to achieve unsupervised learning. The

Table 4.1: Parameters for different learning options

	2 bit	4 bit	8 bit	16 bit	high frequency
α_P	-	-	-	0.01	0.01
β_P	-	-	-	3	3
α_D	-	-	-	0.005	0.005
β_D	-	-	-	3	3
G_{max}	-	-	-	1.0	1.0
G_{min}	-	-	-	0	0
γ_{pot}	0.2	0.3	0.5	0.9	0.3
τ_{pot}	20	30	30	30	80
γ_{dep}	0.2	0.3	0.5	0.9	0.2
τ_{dep}	10	10	10	10	5
f_{input_max}	22	22	22	22	78
f_{input_min}	1	1	1	1	5

MNIST and Fashion-MNIST data sets contain 60,000 training images and 10,000 testing images. In this work, the SNN learns the full set of training images. Pixel intensity of input images, which is an 8-bit value, is encoded into specific spiking frequency of one spike train. For darker pixels, the spiking frequency is higher, as shown in Fig. 3.1 (d). Frequency is in a range between f_{input_max} and f_{input_min} , and proportional to the pixel intensity. Each image is presented to the network for t_{learn} ms. After learning is complete, the first 1000 images in the test set are used to label all the neurons in the first layer. The rest of the test set, which contains 9000 images, are used for inference.

4.3 Low precision learning and rounding options

For low precision learning, conductance of synapses is represented in numbers with precision no greater than 32 bits. Quantization for low precision learning is performed before the LTP/LTD phase of synapse conductance. For 16-bit and 32-bit learning, after floating point calculation of change in conductance ΔG , the result is rounded to a value that can be represented in the current bit width. For 8-bit and lower precision learning, ΔG is set to $1/2^n$, with n being the bit width. Low precision learning in other neural networks such as recurrent neural network (RNN) is shown to have different performance with different

rounding options [23]. For low precision learning in SNN, we study the impact of rounding options to determine if there exists a similar influence. Three rounding options including rounding to nearest, bit truncation and stochastic rounding are tested in this work. For stochastic rounding, the probability of rounding up is related to the position between the two quantized value, and is defined as:

$$P_{round_up} = (\Delta G - \Delta G_{truncated}) \times 2^n \quad (4.1)$$

When a value does not round up in the stochastic operation, it rounds down automatically, i.e. probability of round down is $(1 - P_{round_up})$.

4.4 Parameters and other details

For the LIF model used in this work, V_{th} is -60.2, V_{reset} is -74.7, a is -6.77, b is -0.0989 and c is 0.314. Parameters for STDP algorithm and stochastic behavior of synapses in different precision learning are shown in TABLE 4.1. Initial membrane potential of all neurons on the first layer is -70.0 and conductance of each synapse is initialized with random number in the range of 0.2 to 0.7. Simulations are performed on a desktop machine with Intel Core i7-7700k and NVIDIA GTX 1080 Ti.

CHAPTER 5

RESULTS

5.1 Accuracy Comparison with Existing Simulators

We first evaluate the accuracy of the spiking neuron simulation (no learning) considering an SNN of 10^3 LIF neurons and 10^4 synapses. Fig. 5.1 shows that our platform is able to produce spiking activities similar to CARLsim [20]. However, we observe an increased simulation time in ParallelSpikeSim due to the use of more complex unified data structures. The impact of this increased spike simulation time is overshadowed by the higher learning rate achieved using stochastic STDP. We next verify that deterministic STDP (defined as baseline) in ParallelSpikeSim shows comparable accuracy with the state-of-the-art SNN design with deterministic STDP from Diehl [5]. In Diehl’s work, the network yields an accuracy of 91.9% for the MNIST data set while our baseline test achieves 92.2%.

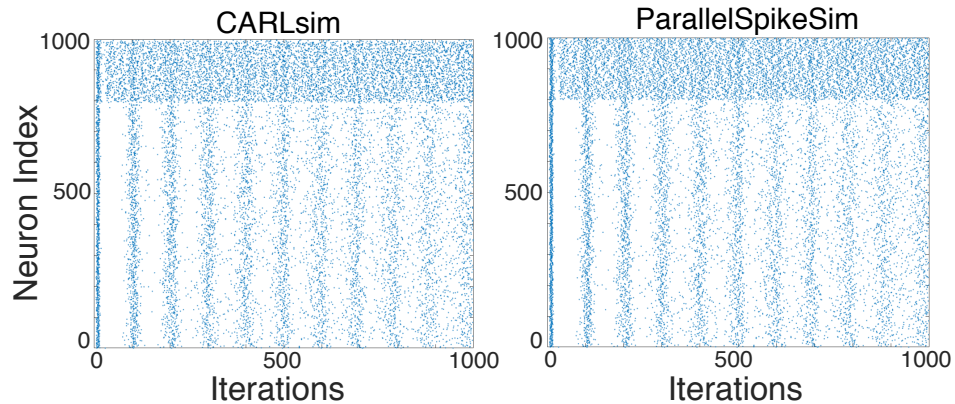


Figure 5.1: Simulation of spiking activity and performance

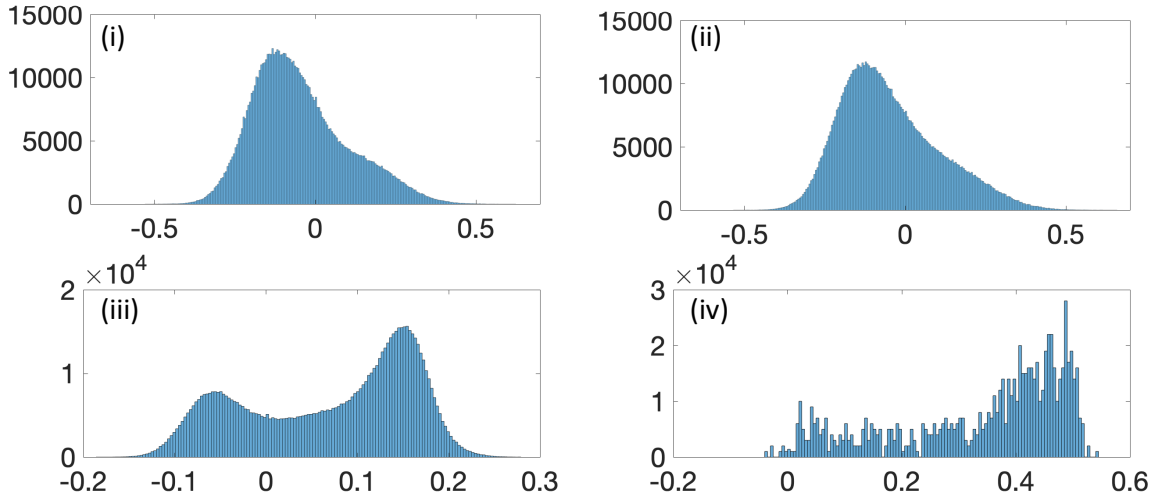


Figure 5.2: Conductance distribution of SNN with (i) stochastic STDP learning MNIST, (ii) deterministic STDP learning MNIST, (iii) stochastic STDP learning fashion MNIST and (iv) deterministic STDP learning fashion MNIST; x axis is conductance value and y axis is the count

5.2 Improved Learning Accuracy with Stochastic STDP

Distribution of all synapse conductance in the network is shown in the top two plots of Fig.5.2. For MNIST, the distribution of stochastic STDP (left) and deterministic STDP (right) both exhibit bell shaped profile. This difference in learning capability is reflected in the conductance distribution as shown in the plots for fashion MNIST in Fig.5.2. For deterministic STDP, conductance accumulates at multiple discrete points across the value range, showing a significant amount overlap of conductance states among all synapses. Meanwhile conductance values of SNN with stochastic STDP spread across continuous and distinct values, resulting in a smoother distribution profile than deterministic STDP.

We observe that both baseline and stochastic STDP are able to produce good accuracy in learning the MNIST data set. Each is able to provide conductance array with good contrast for each class of image as can be seen in Fig. 5.3 (a). Inference result shows that Stochastic STDP is able to provide better result with around 4% higher accuracy. However, for the Fashion MNIST data set, baseline test fails to gain accuracy even after learning all 60,000 images. Visualization in Fig. 5.3(a) shows that all synapses learns the overlapping features

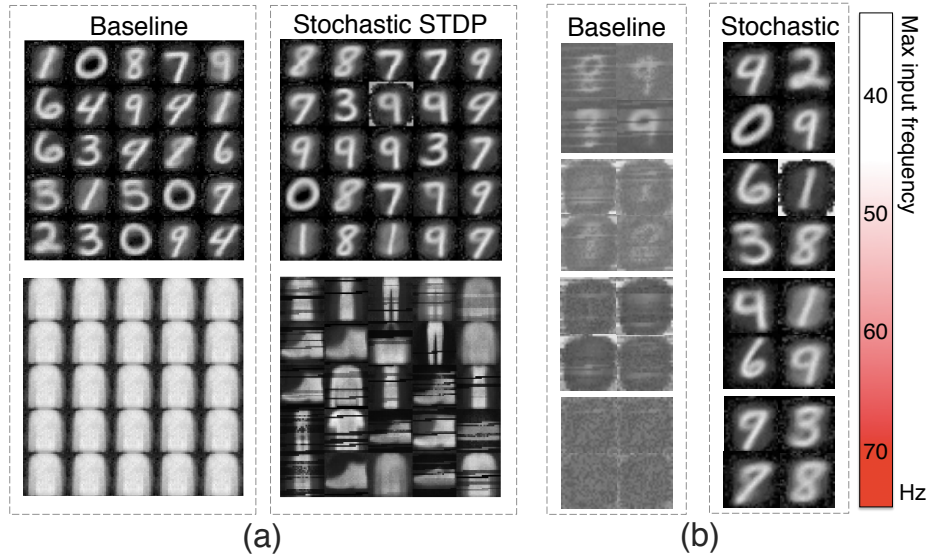


Figure 5.3: Visualization of synapse conductance (a) Baseline and stochastic STDP for MNIST and Fashion MNIST, and (b) effect of input spike train frequency on stochastic STDP.

of all classes. On the other hand, stochastic STDP is able to learn the more complex data set. Comparing visualization of synapse conductance from stochastic STDP learning on the right of Fig. 5.3 (a), baseline design struggles to learn any unique features from input images. This result shows that the level of causal relationship implied by stochastic STDP provides SNN with additional learning ability, and this effect is more prominent in more complex learning tasks.

5.3 Fast Learning with Higher Input Frequency

ParallelSpikeSim allows controlling the frequency of the input spike train to enable trade-off between learning rate and accuracy. One of the bottleneck of SNN learning rate is the time it takes to learn features in each individual image. Due to the influence of inhibition period of the WTA principle, and inherent nature of spiking neurons such as the reset of membrane potential after spikes, learning requires each image to be presented to the network for an extended period of time, so that a sufficient amount of spikes are generated. Since information in this spiking neural network architecture is transmitted in form

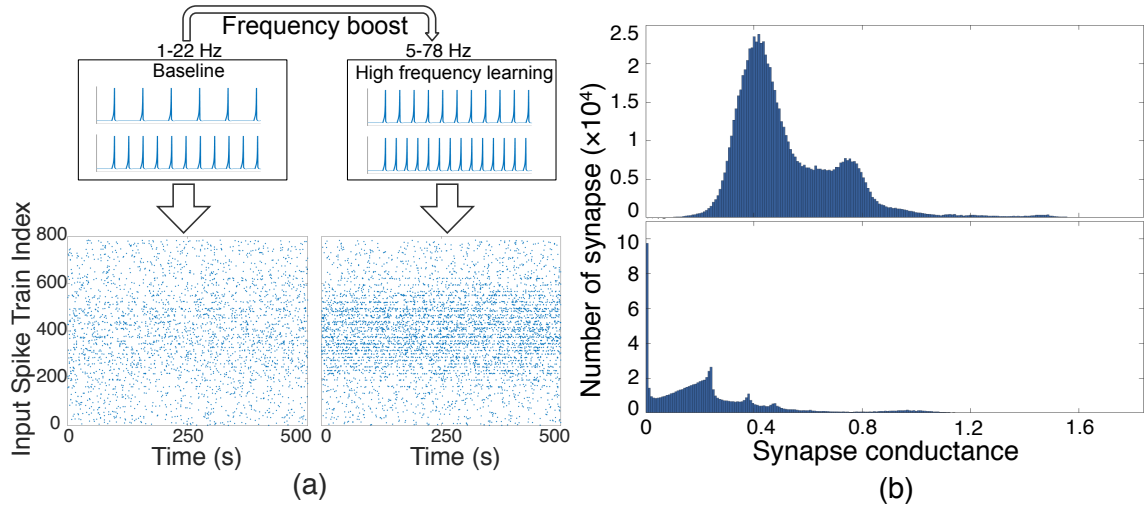


Figure 5.4: High-frequency and low-precision operations (a) Input spike trains at low (left) and high (right) frequencies (each dot represents one spike). (b) Conductance distribution of Q1.7 precision (MNIST) with stochastic (top) and deterministic STDP (bottom).

of spikes, the more frequently spikes can be sent, the faster information can be delivered. Therefore it is desirable to make the spike train operate with higher frequency.

On the other hand, a higher input spike frequency can degrade the learning accuracy. Fig. 5.5 (a) shows learning accuracy loss for different input spike train frequency. We observe that using a value of f_{input_max} above certain value will cause the network to drop sharply in accuracy. This is because at higher input frequencies, the rapid arriving current signal drives multiple spiking neurons to spiking state disregard of their previous learned features, making the inhibition layer less useful and the network gradually shifts to chaotic states. This effect can be observed in the conductance visualization of four frequency ranges shown in Fig. 5.3 (b). As a result, for SNN using deterministic STDP, the optimal f_{input_max} is limited to a relatively low value. In baseline test, the optimal spiking frequency range of neurons in the input layer is 1-22 Hz. At such frequency range, 500 ms learning time for each image is used in order to generate sufficient spikes. For a total simulation time of 542 minutes the baseline architecture is able to learn the 60,000 MNIST images.

In this work, we find that using stochastic STDP with short-term behavior, working frequency range of f_{input_max} can be expanded, as can be seen in Fig. 5.3 (b). More specif-

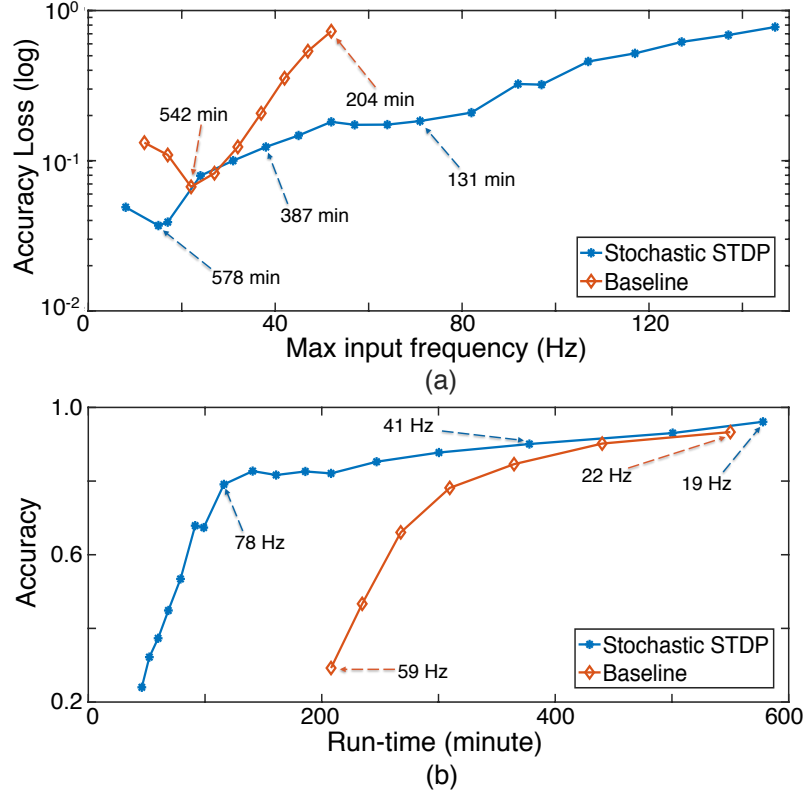


Figure 5.5: High-frequency learning (a) Accuracy loss vs. max input frequency, and (b) Accuracy vs. run-time.

ically, higher τ_{pot} and lower τ_{dep} values for 3.6 and 3.7 are used to create a short-term stochastic STDP behavior, which enhances its ability to adapt to the fast switching input feature. We find the frequency range with maximum error rate of 20%, and the result is $f_{input_min} - f_{input_max}$ at 5-78 Hz. Comparing Fig. 5.4 (a) left and Fig. 5.4 (a) right, which show spike train behavior of baseline and high frequency learning for the MNIST data set, it can be observed that the pattern of darker region, where the written digit is located, is more distinct in high frequency learning. Learning efficiency is therefore significantly increased as time for the network to learn features in each image is reduced. In this high frequency learning mode, frequency range of input spike train can be expanded up to 5-78 Hz before significant decrease of accuracy occurs. At this frequency range, learning time for each image is reduced to 100 ms, leading to a total simulation time of 131 minutes to learn the entire MNIST data set. In baseline test, as shown in Fig. 5.5 (b), achieving the

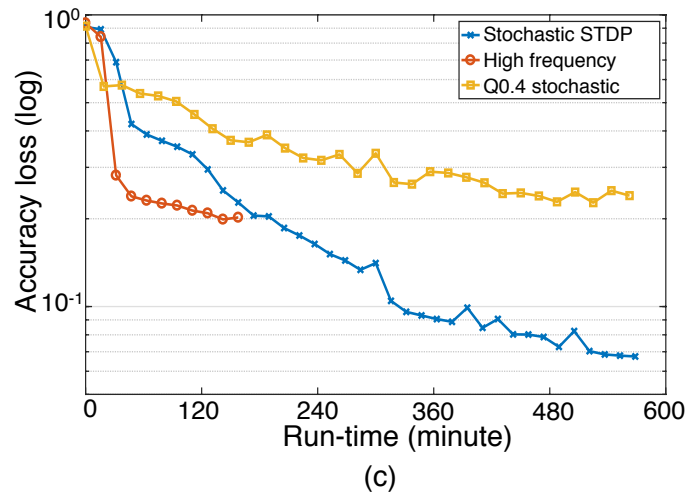
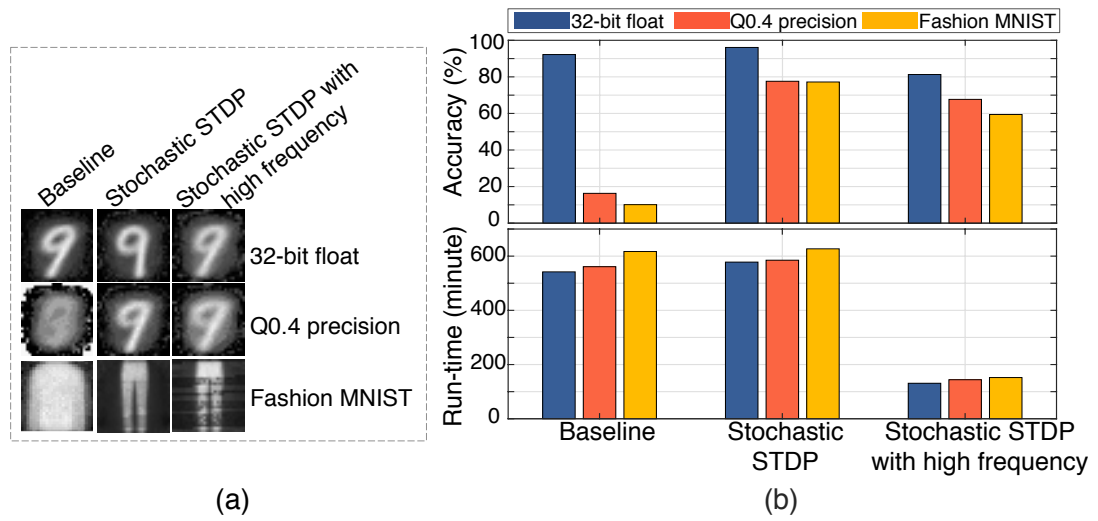


Figure 5.6: Comparison of learning configurations: (a) conductance map, (b) Accuracy and run-time, and (c) accuracy loss vs. simulation time.

same accuracy takes around 380 minutes, around 3 times longer than stochastic STDP.

5.4 Low precision learning

In this work, we performed learning in 2, 4, 8 and 16 fixed point numbers. Baseline test shows poor accuracy result for low precision learning as shown in Table 5.1. This is due to the fact that quantization of conductance in low precision learning increases gap between adjacent conductance values. This leads to rapid changes in conductance during LTP/LTD process and the network quickly lose memory of learned features. This effect is shown in Fig. 5.4 (b), which is the distribution of conductance of all 784,000 synapses connecting input and first layer, for Q1.7 precision learning of the MNIST data set. Distribution of stochastic STDP is on the top of Fig. 5.3 (b) and deterministic STDP on the bottom. Deterministic STDP results in less ideal distribution, in which a large portion of synapses drops to the minimal conductance value. Stochastic STDP in synapse model greatly improves accuracy in low precision learning as it prevents rapid changes from loosely correlated spiking events to help retain memory and at the same time guard the network from fast convergence. The improvement is present in all precision tested, as shown in Table 5.1. Low precision learning, especially the ones with Q1.7 and lower, exhibit larger gain in accuracy from the application of stochastic STDP. Such robustness observed in this SNN design is important as there are many well-known benefits for digital systems to operate in

Table 5.1: Accuracy results (%) for rounding options
Truncation Rounding to nearest Stochastic

	Truncation	Rounding to nearest	Stochastic
Baseline			
Q0.2	9.6	11.3	16.8
Q0.4	13.1	16.3	21.3
Q1.7	28.2	30.8	33.7
Q1.15	52.6	52.8	55.2
Stochastic			
Q0.2	62.3	66.7	64.6
Q0.4	72.4	77.6	79.0
Q1.7	88.5	91.1	90.1
Q1.15	93.2	94.2	94.7

lower bit-width, including less memory usage and less power consumption.

As shown in Table 5.1, for learning in different precision, accuracy drops significantly from Q1.15 to lower precision fixed point. Three rounding options for low precision learning tested exhibit different learning performance. Bit truncation shows the lowest accuracy in all precision tested, while stochastic rounding performs the best in most cases. This is because in low precision learning, stochastic rounding helps to maintain information about numeric position between two quantization points on a statistical point of view. It is also worth noting that stochastic rounding and round to nearest shows similar results for network using stochastic STDP, and the benefit of using stochastic rounding diminishes as bit width increases.

5.5 Summary of Results

Fig. 5.6 summarizes the comparison of comparison of baseline and stochastic STDP learning results. Stochastic STDP shows higher accuracy in challenging learning tasks and lower precision while using similar simulation time as baseline. The high frequency learning (with stochastic STDP) greatly reduces learning time (moving error rate reduces quickly as shown in Fig. 5.6 (c)) but with a graceful degradation of final accuracy.

CHAPTER 6

CONCLUSION

This thesis presents ParallelSpikeSim, a GPU accelerated SNN simulator with unsupervised learning capability using stochastic STDP. We show that ParallelSpikeSim enables accurate learning for complex tasks, facilitates fast learning with increased spike frequency, and allows low-precision operation of SNN even down to 2 bits. ParallelSpikeSim has been released as an open-source SNN simulation platform developed in C++ with CUDA library, to help researchers take advantage of stochastic STDP, and explore applications of SNN to various AI problems.

REFERENCES

- [1] W. B. Levy and O. Steward, “Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus,” *Neuroscience*, vol. 8, no. 4, pp. 791–797, 1983.
- [2] B Gustafsson, H Wigström, W. C. Abraham, and Y. Y. Huang, “Long-term potentiation in the hippocampus using depolarizing current pulses as the conditioning stimulus to single volley synaptic potentials.,” *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 7, no. 3, pp. 774–80, 1987.
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [4] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” *CoRR*, vol. abs/1708.0, 2017.
- [5] P. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in Computational Neuroscience*, vol. 9, no. August, p. 99, 2015.
- [6] D. Querlioz *et al.*, “Immunity to device variations in a spiking neural network with memristive nanodevices,” *IEEE Transactions on Nanotechnology*, vol. 12, no. 3, pp. 288–295, 2013.
- [7] F. Javed, Q. He, L. E. Davidson, J. C. Thornton, J. Albu, L. Boxt, N. Krasnow, M. Elia, P. Kang, S. Heshka, and D. Gallagher, “Brain and high metabolic rate organ mass: Contributions to resting energy expenditure beyond fat-free mass,” *American Journal of Clinical Nutrition*, 2010.
- [8] Y. Long, X. She, and S. Mukhopadhyay, “Accelerating biophysical neural network simulation with region of interest based approximation,” *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 159–164, 2018.
- [9] T. V. P. Bliss and A. R. Gardner-Medwin, “Long-lasting potentiation of synaptic transmission in the dentate area of the unanaesthetized rabbit following stimulation of the perforant path,” *The Journal of Physiology*, 1973.
- [10] W. Gerstner, R. Ritz, and J. L. van Hemmen, “Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns,” *Biological Cybernetics*, 1993.

- [11] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, “Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs,” *Science*, 1997.
- [12] D. Debanne, B. H. Gähwiler, and S. M. Thompson, “Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures,” *Journal of Physiology*, 1998.
- [13] C. C. Bell, V. Z. Han, Y. Sugawara, and K. Grant, “Synaptic plasticity in a cerebellum-like structure depends on temporal order,” *Nature*, 1997.
- [14] J. C. Magee and D. Johnston, “A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons,” *Science*, 1997.
- [15] M. Beyeler *et al.*, “Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule,” *Neural Networks*, vol. 48, pp. 109–124, 2013.
- [16] J. M. Bower, D. Beeman, and M. Hucka, “The GENESIS simulation system,” *The Handbook of Brain Theory and Neural Networks*, no. August 2000, pp. 475–478, 2003.
- [17] R. Brette, M. Rudolph, T. Carnevale, *et al.*, *Simulation of networks of spiking neurons: A review of tools and strategies*, 2007.
- [18] D. F. Goodman and R. Brette, *The brian simulator*, 2009.
- [19] S. Kunkel, S. Kunkel, A. Morrison, *et al.*, “Nest 2.12.0,” *doi.org*, pp. –, 2017.
- [20] T. S. Chou *et al.*, “CARLsim 4: An Open Source Library for Large Scale, Biologically Detailed Spiking Neural Network Simulation using Heterogeneous Clusters,” *IJCNN*, 2018.
- [21] G. Srinivasan, A. Sengupta, and K. Roy, “Magnetic Tunnel Junction Based Long-Term Short-Term Stochastic Synapse for a Spiking Neural Network with On-Chip STDP Learning,” *Scientific Reports*, vol. 6, 2016.
- [22] G.-q. Bi and M.-m. Poo, “Synaptic Modification by Correlated Activity: Hebb’s Postulate Revisited,” *Annual Review of Neuroscience*, 2001.
- [23] T. Na, J. H. Ko, J. Kung, and S. Mukhopadhyay, “On-chip training of recurrent neural networks with limited numerical precision,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, 2017, pp. 3716–3723, ISBN: 9781509061815.