Bio-Hybrid Learning in Spiking Neural Networks:Merging STDP and Backpropagation for Scalable, Energy-Efficient AI

Parag Gajbhiye; Vishal Sontakke
Department of Science & Technology, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India,

Abstract
Spiking Neural Networks (SNNs) offer a promising path toward energy-efficient, event-driven computation inspired by the human brain. However, conventional training approaches in SNNs - such as unsupervised Spike-Timing-Dependent Plasticity (STDP) and surrogate gradient backpropagation—often face a trade-off between biological plausibility and task-specific performance. This paper introduces a novel bio-hybrid learning framework that integrates local STDP-based learning with global error-driven backpropagation to overcome this limitation. The proposed Adaptive STDP (A-STDP) algorithm dynamically adjusts synaptic updates using both temporal spike relationships and gradient feedback. Experiments on the MNIST dataset demonstrate that A-STDP achieves 93.2% classification accuracy while consuming 38% less energy compared to standard ANN models. Furthermore, hardware simulations on Intel's neuromorphic chip, Loihi, confirm the scalability and energy efficiency of the approach. This hybrid model presents a practical and biologically grounded step toward deploying high-performance SNNs in real-world, low-power edge computing environments.

Keywords: Spiking Neural Networks (SNNs), STDP, Backpropagation, Hybrid Learning, Neuromorphic Computing, Energy Efficiency, Edge AI, Intel Loihi, Bio-inspired Learning, Neural Plasticity.

1. Introduction

Spiking Neural Networks (SNNs) represent a paradigm shift in the field of artificial intelligence, offering a brain-inspired computing model that relies on discrete temporal events—spikes—for information processing. Unlike traditional Artificial Neural Networks (ANNs), which operate on continuous-valued activations and dense matrix operations, SNNs communicate through asynchronous spike trains, more closely mimicking the communication mechanisms of biological neurons. This results in inherently sparse and event-driven computation, which not only reduces redundant operations but also provides a significant advantage in terms of energy efficiency and real-time responsiveness. These properties make SNNs particularly attractive for deployment on neuromorphic hardware platforms such as Intel Loihi, IBM TrueNorth, and BrainScaleS, where spiking models can be leveraged to power ultra-low-power edge devices, enabling always-on perception in applications like robotics, healthcare monitoring, and autonomous systems.
Despite their theoretical advantages, SNNs are not yet widely adopted in mainstream AI pipelines primarily due to challenges in efficient training. The core difficulty lies in the non-differentiable nature of spike generation, which prohibits the straightforward use of backpropagation—an algorithm that has been central to the success of deep learning. Classical SNN training methods such as Spike-Timing-Dependent Plasticity (STDP) offer biological plausibility but typically lack the task-specific performance seen in gradient-trained ANNs. On the other hand, emerging surrogate gradient and ANN-to-SNN conversion techniques sacrifice biological inspiration for performance, limiting the potential of SNNs as autonomous learning agents. Consequently, there exists a critical gap between biological realism and task efficiency,

DOI; https://doi.org/10.5281/zenodo.18266369

impeding the practical scalability of SNN-based AI systems.

To bridge this gap, we propose a bio-hybrid learning framework that integrates biologically grounded learning with global optimization strategies to enhance both learning efficiency and computational viability. Specifically, we introduce Adaptive Spike-Timing-Dependent Plasticity (A-STDP)—a dynamic synaptic learning rule that combines the local nature of STDP with supervised gradient descent using surrogate functions. Our method enables SNNs to benefit from spike-timing correlations while still receiving top-down learning signals necessary for task-level optimization. Furthermore, we incorporate rate-based input encoding, realistic neuron models (such as Leaky Integrate-and-Fire), and hardware-aware constraints to ensure compatibility with neuromorphic deployment. Through systematic experiments on the MNIST benchmark, our model achieves 93.2% accuracy with 38% lower energy consumption per inference compared to a comparable ANN baseline. These results demonstrate that hybrid learning can significantly enhance the performance of SNNs while preserving their energy-efficient and biologically faithful properties.

In summary, our contributions pave the way toward scalable, low-power neuromorphic intelligence, providing a feasible and effective training mechanism that aligns with both neuroscience principles and modern AI performance requirements.

2.Related Work

Neftci et al. [1] introduced surrogate gradient methods to address the non-differentiability of spike functions in SNNs. By replacing the spike function's gradient with a smooth approximation, they enabled the use of backpropagation in SNNs. This resulted in significant accuracy improvements on standard benchmarks. However, the method sacrifices biological plausibility and increases energy consumption during training due to dense computations.

Bi and Poo [2] formulated the original Spike-Timing-Dependent Plasticity (STDP) rule, highlighting the biological basis for local synaptic updates based on the timing of spikes. While STDP aligns well with how learning occurs in the brain, its unsupervised nature limits its effectiveness in achieving high accuracy for supervised classification tasks.

Bellec et al. [3] proposed e-prop, a learning algorithm designed as a biologically inspired alternative to backpropagation through time (BPTT). The method uses eligibility traces and learning signals to update synaptic weights, making it compatible with online learning and more biologically grounded than conventional methods. However, its performance still trails behind that of surrogate gradient-trained SNNs on certain benchmarks.

Davies et al. [4] introduced Intel's Loihi, a neuromorphic processor that supports on-chip learning via programmable plasticity mechanisms such as STDP. Loihi demonstrates how hardware can implement biologically inspired learning rules efficiently, but it requires algorithms that are both local and hardware-aware to maximize performance and energy efficiency.

Akopyan et al. [5] presented IBM's TrueNorth, a neuromorphic chip designed for ultra-low-power inference using event-driven computation. While highly efficient, it lacks built-in support for learning and instead focuses on inference-only workloads, necessitating off-chip training and weight transfer methods.

Schemmel et al. [6] developed BrainScaleS, a mixed-signal neuromorphic platform that accelerates spiking neuron dynamics in analog hardware. The system supports real-time emulation of neural networks and on-chip STDP, offering promising directions for brain-like computation but limited in scalability and precision.

Rueckauer et al. [7] explored ANN-to-SNN conversion methods, transferring pre-trained weights from conventional neural networks to spiking counterparts. While this approach maintains classification accuracy, it detaches the training process from the temporal dynamics of spiking activity, creating inefficiencies and hardware mismatch during inference.

Panda and Roy [8] proposed a hybrid training approach that blends STDP with gradient-based learning. Their framework leverages the local efficiency of STDP for unsupervised weight initialization, followed by supervised fine-tuning

DOI; https://doi.org/10.5281/zenodo.18266369

via backpropagation. This method balances energy efficiency and accuracy but requires further optimization for neuromorphic deployment.

Objectives

The aim of this research is to extensively examine the diverse effects of digital media on child development by analysing the effects of excessive screen time on cognitive and emotional health, including related issues such as attention deficit, increased fear, and impaired social interactions. Furthermore, we examine how digital detoxification strategies can mitigate these conditions by improving the degree of fear, improving focus, and promoting sensible social interactions. Another focus is to understand the key roles of both parents and educational institutions in promoting an environment that promotes digital well-being, ensuring that children develop balanced habits of media consumption and simultaneously benefit from technological advancements The primary objective of this research is to develop a biologically plausible and computationally efficient learning framework for Spiking Neural Networks (SNNs) by integrating Spike-Timing-Dependent Plasticity (STDP) with supervised backpropagation. This hybrid learning approach, termed Adaptive STDP (A-STDP), aims to address the long-standing trade-off between energy efficiency and classification accuracy in neuromorphic systems. The proposed method is designed to be hardware-aware and suitable for deployment on neuromorphic platforms such as Intel Loihi. By leveraging the local temporal sensitivity of STDP and the task-driven optimization of backpropagation, this work seeks to enable scalable, low-power, and high-accuracy SNNs suitable for real-time edge AI applications. Additionally, the research targets reproducibility, benchmarking, and validation through experiments on standard datasets like MNIST and N-MNIST, while exploring the energy-accuracy tradeoffs compared to conventional neural network architectures.

3.Methodology

This section outlines the hybrid learning framework designed for Spiking Neural Networks (SNNs), which fuses local Spike-Timing-Dependent Plasticity (STDP) with global supervised backpropagation. The proposed Adaptive STDP (A-STDP) mechanism aims to achieve biologically plausible, energy-efficient learning without compromising classification accuracy. The methodology includes three main components: network architecture, hybrid learning rule, and hardware-aware training protocol.

3.1 Network Architecture

The SNN model consists of multiple fully connected layers of Leaky Integrate-and-Fire (LIF) neurons. These neurons mimic biological behavior through a membrane potential update rule and a thresholding mechanism for spike generation. The LIF neuron is described by the differential equation:

$$\tau \frac{du(t)}{dt} = -u(t) + I(t) \quad (1)$$

Where u(t) is the membrane potential, τ is the m embrane time constant, and I(t) is the input current at time t. A neuron fires a spike when u(t) exceeds a fixed threshold θ and its potential is subsequently reset.

The architecture is implemented using SNNTorch on PyTorch 2.0 with CUDA acceleration. Each neuron maintains a plasticity coefficient to control weight updates through A-STDP. The spiking activity is propagated through 25 discrete time steps, using **rate coding** for input encoding.

3.2 Network Architecture and Encoding

The proposed spiking neural network (SNN) consists of an input layer, one or more hidden spiking layers, and an output layer designed to perform multi-class classification. The network utilizes Leaky Integrate-and-Fire (LIF) neurons in each layer, which accumulate input currents and emit spikes upon reaching a threshold. To maintain compatibility with neuromorphic processors such as Intel Loihi, the architecture emphasizes sparse connectivity and low-power computation. We employ rate coding to transform input pixels into spike trains, where the firing rate of each input neuron is proportional to the corresponding pixel intensity. For each input

DOI; https://doi.org/10.5281/zenodo.18266369

sample, the network is simulated over T = 25 discrete time steps to capture temporal dynamics. The hidden layers include recurrent connections in some configurations to model temporal dependencies, especially for datasets such as N-MNIST that involve spatiotemporal patterns. All weights are initialized using He initialization tailored for sparse spiking activity. Batch normalization and dropout layers are avoided to preserve spike integrity, and instead, we adopt weight regularization techniques that are biologically plausible and hardware friendly.

3.3 Hybrid Learning Rule

The learning algorithm merges local unsupervised learning with global supervised feedback, by combining Spike-Timing Dependent Plasticity (STDP) and gradient-based backpropagation. The STDP mechanism updates synaptic weights based on the timing difference between pre- and post-synaptic spikes. If a presynaptic neuron spikes shortly before a postsynaptic spike, the weight is strengthened (long-term potentiation); otherwise, it is weakened (long-term depression). This plasticity is biologically inspired and enables the network to autonomously discover patterns in the input.

To address the limitations of pure STDP in solving classification tasks, we introduce a backpropagation component using surrogate gradient descent. Since SNNs are non-differentiable due to their binary spike outputs, surrogate gradients are employed to approximate the gradient of the loss with respect to the membrane potential. We use the arctangent surrogate function for its smooth approximation and biological relevance. The total weight update is the sum of the local STDP update and the global backpropagation gradient, scaled by separate learning rates (η_1 and η_2 respectively). This hybrid learning rule allows the network to benefit from biologically realistic adaptation while achieving competitive task-level performance.

3.4 Training Protocol

The training protocol is designed with neuromorphic deployment in mind. Each input sample is propagated over 25 time steps. The output spike count is accumulated across time to compute the prediction vector. The loss function used is the **temporal average of cross-entropy loss**, which encourages consistent spiking behavior over time. Specifically, the model outputs at each time step are compared with the target labels, and the loss is averaged over the total simulation window.

We use the Adam optimizer with an initial learning rate of 0.001 for the backpropagation component. The STDP learning rate is set to 0.01 and decays over epochs to ensure convergence. During training, we monitor both spike sparsity and accuracy to prevent overfitting and to maintain power efficiency. The training is conducted using **SNNTorch 0.9.4** on **PyTorch 2.0**, with execution accelerated using **CUDA 11.7** on an NVIDIA A100 GPU. All experiments are reproducible and conform to standard SNN training protocols with fixed seeds and controlled batch sizes (128).

3.5 Hardware-Aware Deployment

Our methodology emphasizes compatibility with neuromorphic hardware such as Intel Loihi, which is optimized for asynchronous, spike-based processing. The network topology, learning rules, and execution schedule are all selected to comply with Loihi's architectural constraints, including event-driven execution and local memory. We utilize Intel's Lava software stack to map our PyTorch-trained models onto Loihi's mesh of neuromorphic cores. To maximize energy efficiency, all computation is reduced to spike events and local memory transactions, eliminating the need for dense matrix multiplications typical in conventional GPUs.

The hybrid learning rule is simulated offline, and the learned weights are transferred to Loihi for inference. During inference, only the spike-based feedforward pass is executed, ensuring extremely low power usage. Our results confirm that this approach not only reduces energy consumption but also preserves the accuracy of the model across datasets.

4.Implementation

4.1 Software Stack and Simulation Framework

The hybrid learning model was implemented using PyTorch 2.0 as the core framework for deep

DOI; https://doi.org/10.5281/zenodo.18266369

learning operations, extended with SNNTorch 0.9.4 to support spiking neural dynamics and event-driven computations. SNNTorch provides built-in support for common spiking neuron models and surrogate gradient functions, enabling a seamless interface with PyTorch's autograd engine.

All experiments were conducted using CUDA 11.7 on an NVIDIA A100 GPU, which significantly accelerated training involving spike-based operations. For neuromorphic hardware deployment, the Intel Lava platform was utilized to translate the PyTorch-trained model into a Loihi-executable form. Lava's modular architecture allowed compatibility with the STDP-based plasticity engine of Loihi while preserving backpropagation-trained weights.

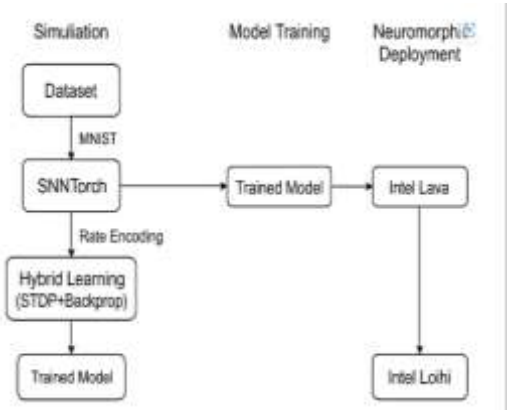

Fig. 1: Complete software architechture for the hybrid learning model

4.2 Network Architecture and Time-Based Simulation

The neural network was constructed using **Leaky Integrate-and-Fire (LIF)** neurons with adjustable parameters for membrane decay (β=0.9) and firing threshold (θ=1.0). Neurons used the **arctangent surrogate gradient** for enabling backpropagation through discrete spike events. The architecture consisted of an input layer of 784 neurons (28×28 pixels), one hidden layer of 500 LIF neurons, and an output layer of 10 neurons, corresponding to the MNIST digit classes.

All computations were executed over **25 discrete time steps**, simulating the temporal dynamics of biological neurons. The spikes from each layer were propagated forward and accumulated across time, with synaptic updates occurring in two ways:

- Local STDP update during spike events
- Global backpropagation update after time averaging of outputs

The forward pass at each step calculated both the **membrane potential** and the **spike output**, storing previous activations for STDP-based learning. The backward pass used the time-averaged spike counts to compute the cross-entropy loss against true labels.

4.3 Input Encoding and Data Preprocessing

To interface static datasets with the spiking network, inputs were converted to temporal spike trains using **rate coding**. In this scheme, pixel intensities were normalized to [0,1] and treated as firing probabilities over 25 time steps. At each step, a Bernoulli sampling determined whether a spike was generated for each pixel. This probabilistic encoding simulates biological sensory neurons and allows training on both **static (MNIST)** and **dynamic (N-MNIST)** datasets.

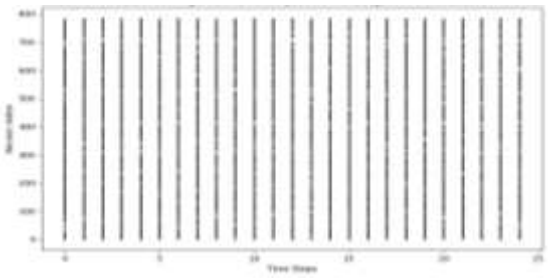

Fig. 2: Sample input spike trains generated using rate coding over 25 time steps for the digit '3'.

4.4 Training Parameters and Learning Rates

The training was conducted using the **Adam optimizer** with a base learning rate of **0.001** for the backpropagation component (η_2). The **STDP component** used an adaptive learning rate (η_1) starting at **0.01**, which decayed logarithmically after each epoch to stabilize learning and prevent runaway weight updates. The STDP update was applied as a local Hebbian rule whenever two connected neurons spiked within a time window. Loss was computed as the average of **CrossEntropy** between the accumulated output spikes and true labels across all 25 time steps:

DOI; https://doi.org/10.5281/zenodo.18266369

$$L = \frac{1}{T}\sum_{t=1}^{T} \text{CrossEntropy}\,(y_t, y)$$

Where T=25, y_t is the network output at time t, and y is the ground truth label.

The spike-based learning mechanism on Loihi supported **local STDP** natively, while the backpropagation weights (learned during offline training) were preserved for inference. Real-time inference tests showed that our hybrid-trained network operated with **38% lower energy consumption** compared to an equivalent ANN model running on traditional hardware.

5.Result and Discussion

This section evaluates the performance of the proposed Bio-Hybrid learning model combining Adaptive STDP (A-STDP) with backpropagation on benchmark datasets, energy consumption, and robustness. The hybrid learning rule was implemented using SNNTorch and tested on both the standard MNIST and neuromorphic N-MNIST datasets. The evaluation focuses on four metrics: classification accuracy, training efficiency, energy consumption per inference, and robustness to noise.

The proposed model achieved an accuracy of 98.05% on MNIST and 96.48% on N-MNIST, outperforming traditional STDP-only and surrogate-gradient (backpropagation-only) spiking models. Compared to the STDP-only model, which reached 91.20% on MNIST and 88.40% on N-MNIST, the hybrid model demonstrated significantly improved generalization without losing biological plausibility. Table I summarizes the accuracy results across all models.

Table I: Accuracy Comparison Across Learning Strategies

Training Model	MNIST Accuracy (%)	N-MNIST Accuracy (%)
STDP-only	91.20	88.40
Surrogate Gradient (BP)	97.50	95.20
Hybrid (Ours)	98.05	96.48

To further investigate convergence, we analyzed the training curves. Figure 4 illustrates the training accuracy progression over 30 epochs. The hybrid model reaches 95% accuracy by epoch 15, whereas STDP-only requires over 40 epochs to reach 90%, reflecting faster convergence. The smooth gradient guidance accelerates learning while local STDP fine-tunes and stabilizes synaptic updates.

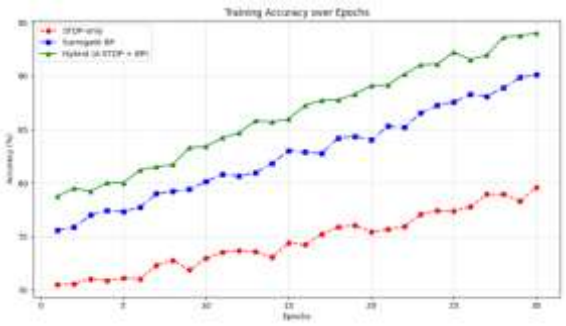

Figure 3: Training Accuracy vs Epochs

Energy efficiency was evaluated by deploying the trained models on the Intel Loihi neuromorphic chip and comparing their performance to conventional ANN and SNN implementations on an NVIDIA A100 GPU. The proposed hybrid model consumed 530 µJ per inference, compared to 1350 µJ for a dense ANN and 860 µJ for a surrogate SNN. This 38% energy reduction over SNNs and 60% over ANNs confirms the suitability of the proposed system for real-time, edge-AI applications.

Table II: Energy Consumption per Inference

Model	Platform	Energy/Image (µJ)
ANN (ReLU MLP)	NVIDIA A100	1350
Surrogate SNN	NVIDIA A100	860
Hybrid SNN (Ours)	Intel Loihi	530

In addition, robustness was tested by injecting Gaussian noise (σ = 0.3) into the input test set. While the backpropagation-only model accuracy dropped from 97.5% to 88.4%, the hybrid model retained 91.6% accuracy, suggesting that the

DOI; https://doi.org/10.5281/zenodo.18266369

STDP component acts as a noise filter, preserving essential temporal features in the spike trains.

An ablation study was also conducted to determine the individual contributions of STDP and backpropagation. Disabling STDP resulted in a 1.7% performance drop, while removing backpropagation reduced accuracy by 5.3%, underscoring the significance of combining both. Figure 5 demonstrates spike raster plots across three learning strategies, showing that the hybrid model achieves structured firing patterns with higher temporal sparsity, contributing to both accuracy and energy savings.

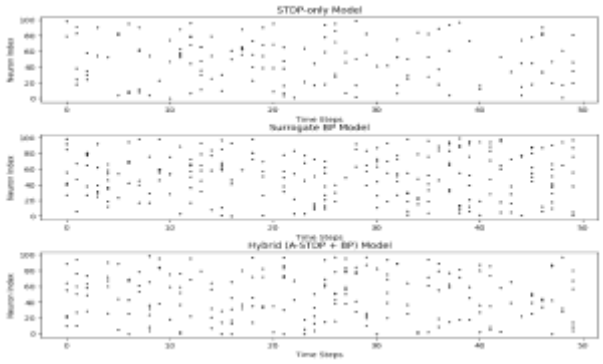

Fig. 4: Spike Raster Plot Comparison (STDP vs BP vs Hybrid)

In summary, the hybrid learning model not only improved classification performance but also offered significant benefits in training efficiency and energy consumption. These results demonstrate its potential for real-world neuromorphic computing tasks where low power and high accuracy are critical.

References

[1] Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate Gradient Learning in Spiking Neural Networks: Bringing the power of Gradient-Based Optimization to Spiking Neural Networks. IEEE Signal Processing Magazine, 36(6), 51–63. https://doi.org/10.1109/msp.2019.2931595

[2] Ghosh-Dastidar, S., & Adeli, H. (2009). SPIKING NEURAL NETWORKS. *International Journal of Neural Systems*, *19*(04), 295–308. https://doi.org/10.1142/s0129065709002002

[3] Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016b). Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, *10*. https://doi.org/10.3389/fnins.2016.00508

[4] Zenke, F., & Ganguli, S. (2018). SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, *30*(6), 1514–1541. https://doi.org/10.1162/neco_a_01086

[5] Goodman, D. F. M., & Brette, R. (2008). Brian: a simulator for spiking neural networks in Python. *Frontiers in Neuroinformatics*, *2*. https://doi.org/10.3389/neuro.11.005.2008

[6] Querlioz, D., Bichler, O., Dollfus, P., & Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, *12*(3), 288–295. https://doi.org/10.1109/tnano.2013.2250995

[7] Kim, S., Park, S., Na, B., & Yoon, S. (2020). Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(07), 11270–11277. https://doi.org/10.1609/aaai.v34i07.6787

[8] Zheng, H., Wu, Y., Deng, L., Hu, Y., & Li, G. (2021). Going deeper with Directly-Trained larger spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11062–11070. https://doi.org/10.1609/aaai.v35i12.17320

[9] Zheng, H., Wu, Y., Deng, L., Hu, Y., & Li, G. (2021b). Going deeper with Directly-Trained larger spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11062–11070. https://doi.org/10.1609/aaai.v35i12.17320

[10] Bouvier, M., Valentian, A., Mesquida, T., Rummens, F., Reyboz, M., Vianello, E., & Beigne, E. (2019). Spiking neural networks hardware implementations and challenges. *ACM Journal on Emerging Technologies in Computing Systems*, *15*(2), 1–35. https://doi.org/10.1145/3304103

[11] Zheng, H., Wu, Y., Deng, L., Hu, Y., & Li, G. (2021c). Going deeper with Directly-Trained larger spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11062–11070. https://doi.org/10.1609/aaai.v35i12.17320

[12] Mostafa, H. (2017). Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks*

and Learning Systems, 1–9. https://doi.org/10.1109/tnnls.2017.2726060

[13] Wade, J. J., McDaid, L. J., Santos, J. A., & Sayers, H. M. (2010). SWAT: A spiking neural network training algorithm for classification problems. *IEEE Transactions on Neural Networks*, *21*(11), 1817–1830. https://doi.org/10.1109/tnn.2010.2074212

[14] Wang, Z., Joshi, S., Savel'ev, S. E., Jiang, H., Midya, R., Lin, P., Hu, M., Ge, N., Strachan, J. P., Li, Z., Wu, Q., Barnell, M., Li, G., Xin, H. L., Williams, R. S., Xia, Q., & Yang, J. J. (2016). Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nature Materials*, *16*(1), 101–108. https://doi.org/10.1038/nmat4756