Spiking Synchrony as a Learning Signal for Spiking Neural Networks

YUCHEN TIAN

Bachelor of Engineering

Supervisor: Professor Omid Kavehei
Associate Supervisor: Dr. Nhan Duy Truong

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

School of Biomedical Engineering
Faculty of Engineering
The University of Sydney
Australia

24 February 2026

Declaration

This thesis is submitted to the University of Sydney in fulfillment of the requirements of Master of Philosophy. I hereby declare that the work presented in this thesis report has not been submitted, either in full or in part, for a degree at this or any other institution, and this is the result of my independent work. This thesis contains no material previously used by other users except where due references are included. Portions of the language editing in this thesis were assisted by ChatGPT under the author's direct supervision. All ideas, analyses, experimental results, and conclusions remain entirely the author's own work, and the author retains full responsibility for the accuracy and integrity of the content.

Abstract

Spiking neural networks promise high energy efficiency, yet training deep Spiking Neural Networks (SNNs) with timing-based local rules remains unstable and hard to scale. This thesis proposes Spike-Synchrony-Dependent Plasticity (SSDP), a population-level learning rule that strengthens synapses in proportion to the temporal co-firing of pre- and post-synaptic activity aggregated over a mini-batch. SSDP operates on group synchrony with a continuous Gaussian kernel and requires only binary spike indicators and first-spike indices. The implementation avoids per-synapse eligibility by computing first-spike times once per channel, making it compute- and memory-efficient and friendly to neuromorphic deployment. To improve credit assignment, we introduce Dopamine-Modulated Spike-Synchrony-Dependent Plasticity (DA-SSDP), which gates the synchrony update with a loss-negative feedback signal. In practice, SSDP and DA-SSDP act as auxiliary learners that run alongside backpropagation, injecting a biologically local learning signal toward synchronous firing while leaving the model's structure unchanged. Across static vision, event-based vision, and auditory temporal benchmarks, SSDP consistently improves accuracy and robustness without increasing model size. Analyses of spike timing jitter and feature show that SSDP sharpens class-relevant synchrony and enhances the model's generalisation capacity. These results demonstrate that population-level synchrony is a practical and scalable learning mechanism for deep SNNs. The approach bridges biological plausibility and engineering efficiency, paving the way for future neuromorphic systems.

Acknowledgements

My first thanks go to Professor Omid Kavehei, my supervisor. His guidance and high standards sharpened my thinking and lifted the quality of this thesis. My second thanks go to my associate supervisor, Dr Nhan Duy Truong. Thank you for your unwavering guidance and rigorous questions.

I am grateful to Dr Jason K. Eshraghian from the University of California, Santa Cruz, for expert suggestions on spiking neural networks and for conversations that shaped the problem, the experiments, and the writing. I also thank Dr Eshraghian's student, Assel Kembay, whose careful edits and attention to detail clarified the manuscript and improved its presentation.

I owe much to my friends Qinglong Li and Samuel Tensingh for their steady encouragement throughout my Master. I am equally thankful to the members of NeuroSyd for a collegial, stimulating environment at the University of Sydney.

I hope this thesis marks not an end but a beginning – the starting point of a longer research journey. The road of science is often solitary, yet with your presence and companionship, it has never felt lonely.

Author Attribution Statement

I, Yuchen Tian, hereby declare that this thesis contains material that has been published or submitted for publication. In accordance with the University of Sydney Thesis and Examination Policy (2020), the authorship and contribution for each co-authored work included in this thesis are listed below. I confirm that the versions included in the thesis are consistent with the scholarly record and that permission to include these materials has been obtained as required.

Works included in the thesis

(1) Y. Tian, A. Kembay, S. Tensingh, N. D. Truong, J. K. Eshraghian, and O. Kavehei, *Learning with Spike Synchrony in Spiking Neural Networks*, arXiv:2505.14841, 2025. doi:10.48550/arXiv.2505.14841. Status: submitted.

Statement of contributions

- **Y. Tian (Candidate) :** First author, corresponding author, conceptualisation, methodology, algorithm design and implementation, experiments, data curation, analysis, visualisation, writing-original draft, writing—review & editing.
- **A. Kembay :** Writing-review & editing.
- **Samuel Tensingh :** Hardware perspective.
- **N. D. Truong (Co-supervisor) :** Technical guidance, writing review.
- **J. K. Eshraghian :** Technical guidance, writing review.
- **O. Kavehei (Principal supervisor) :** Supervision, conceptualisation, result evaluation, article writing, writing—review & editing, project administration.

(2) Y. Tian, S. Tensingh, J. K. Eshraghian, N. D. Truong, and O. Kavehei *Synchrony-Gated Plasticity with Dopamine Modulation for Spiking Neural Networks*
Status: accepted by Transactions on Machine Learning Research (TMLR).

Statement of contributions

- **Y. Tian (Candidate) :** First author, corresponding author, conceptualisation, methodology, algorithm design and implementation, experiments, data curation, analysis, visualisation, writing-original draft, writing—review & editing.
- **S. Tensingh :** Hardware perspective, writing—review & editing.
- **J. K. Eshraghian :** Technical guidance, writing review.
- **N. D. Truong (Co-supervisor) :** Technical guidance, writing review.
- **O. Kavehei (Principal supervisor) :** Supervision, conceptualisation, result evaluation, article writing, writing—review & editing, project administration.

Permissions and attestation

Where I am not the corresponding author of a published or submitted item, permission to include the material in this thesis has been obtained from the corresponding author. All co-authors have been informed that the work is included in this thesis.

I attest that the contribution statements above are true and accurate.

Candidate signature	**Supervisor signature**
Name: **Yuchen Tian**	Name: **Prof. Omid Kavehei**
Date: 20 Oct. 2025	Date: 20 Oct. 2025

School of Biomedical Engineering, Faculty of Engineering, The University of Sydney

Nomenclature

Adam	Adaptive Moment Estimation
AdamW	Adaptive Moment Estimation with Weight Decay
AI	Artificial Intelligence
AMP	Automatic Mixed Precision
ANN	Artificial Neural Network
ANNs	Artificial Neural Networks
ASICs	Application-Specific Integrated Circuits
BCM	Bienenstock-Cooper-Munro
BP	Backpropagation
CIFAR	Canadian Institute for Advanced Research
CPU	Central Processing Unit
DA-SSDP	Dopamine-Modulated Spike-Synchrony-Dependent Plasticity
DH-SRNN	Temporal Dendritic Heterogeneity Spiking Recurrent Neural Network
DSSA	Dual Spike Self-Attention
DVS	Dynamic Vision Sensor
EEG	Electroencephalogram
GPU	Graphics Processing Unit
GWFFN	Grouped Convolutional Feed-forward Blocks
IQR	Interquartile Range
KS-test	Kolmogorov-Smirnov test
LIF	Leaky Integrate-and-Fire
LTP	long-term potentiation
MACs	Multiply-accumulate Operations
MNIST	National Institute of Standards and Technology database
N-MNIST	Neuromorphic National Institute of Standards and Technology database
PC	Principal Component
PCA	Principal Component Analysis

RGB	Red, Green, and Blue
R-STDP	Reward-modulated Spike-Timing-Dependent Plasticity
SGD	Stochastic Gradient Descent
SHD	Spiking Heidelberg Digits
SNN	Spiking Neural Network
SNNs	Spiking Neural Networks
SOPs	Synaptic Operations
SRNN	Spiking Recurrent Neural Networks
SSC	Spiking Speech Commands
SSDP	Spike-Synchrony-Dependent Plasticity
S2-STDP	Stabilized Supervised Spike-Timing-Dependent Plasticity
SSTDP	Supervised Spike-Timing-Dependent Plasticity
STDP	Spike-Timing-Dependent Plasticity
S-TLLR	STDP-inspired Temporal Local Learning Rule
t-SNE	t-distributed Stochastic Neighbour Embedding

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This thesis aims to improve artificial intelligence systems by mimicking how human brain neurons work. Modern Artificial Intelligence (AI) systems achieve remarkable performance, but they often consume substantial energy and computational resources. By contrast, the human brain performs sensing, memory, and decision-making with modest power, communicating via spikes when membrane potentials cross threshold, yielding sparse, stimulus-dependent activity. Neuromorphic computing explores hardware and algorithms that follow brain-like principles such as event-driven communication, sparse activity, and local adaptation. Spiking Neural Networks (SNNs) achieve these ideas at the model level. Neurons emit spikes only when needed, reducing redundant computation, and synapses can adapt using local information rather than relying solely on global signals. However, there remains a gap between this appealing picture and practice. Deep SNNs remain difficult to train at scale when combined with pairwise timing rules, which rely on exact spike ordering and often become unstable and memory-intensive in deep architectures. On the other hand, global signals such as backpropagation are powerful but push an error signal through every parameter, which is effective mathematically but far from how neurons appear to learn. This work takes a middle path. Instead of focusing on the exact spike order between two neurons, it looks at how groups of neurons fire together and uses that synchrony as a local learning signal. We develop a spike-synchrony-based plasticity rule, termed Spike-Synchrony-Dependent Plasticity (SSDP), that strengthens connections when pre- and post-synaptic populations are active within the setting timestep, using only binary "spiked or not" indicators and first-spike timing. We then introduce a dopamine-like gate that scales this synchrony update with a task-level feedback signal, so that useful coordination is reinforced and unhelpful coordination fades.

1.1 Motivation

Biological circuits combine three properties that current engineering practice underuses: sparse, event-driven communication; local plasticity shaped by population activity; and neuromodulatory feedback. In contrast, most engineered Spiking Neural Network (SNN) pipelines either propagate a single global gradient through all parameters or rely on pairwise timing rules that track precise spike order. The former is effective but departs from brain-like local learning and low traffic, while the latter struggles with credit assignment in deep architectures, often requires per-synapse memory, and can be fragile when data statistics shift. Consequently, SNNs still lag behind comparable Artificial Neural Networks (ANNs); however, refining SNN neuron models and developing local plasticity rules that compensate for the precision loss inherent to spike-based representations hold promise for narrowing this gap. These tensions motivate a new direction: treat population-level synchrony as a complementary learning signal and, when appropriate, gate it with a simple global feedback. Such a signal is easy to read from activity already produced during a forward pass, aligns with neuromorphic constraints on bandwidth and memory, and encourages useful coordination without imposing heavy computation or changing the model's forward path.

1.2 Problem Statement

Artificial intelligence would benefit from learning signals that reflect how biological circuits operate: sparse, event-driven communication; plasticity shaped by population activity rather than isolated pairs; and lightweight global feedback that conveys task-level credit. Contemporary training pipelines for SNNs do not fully embody these properties. In practice, most engineered SNN training follows one of two extremes. Global backpropagation drives a single error signal through all parameters, achieving high accuracy but departing from brain-like locality and imposing heavy data movement. Local timing rules, typified by Spike-Timing-Dependent Plasticity (STDP), compare the precise order of pre- and post-synaptic spikes. These pairwise rules struggle with credit assignment in deep architectures, often require per-synapse memory (termed eligibility traces), and are sensitive to changes in data statistics, which limits robustness. Population-level coordination is an important feature of biological computation, yet a compact synchrony signal is largely absent from mainstream SNN training. Current methods either over-specify the micro-ordering of spikes or ignore coordinated firing altogether. As a result, models lack an inductive bias that rewards meaningful co-activation across groups of neurons.

SNNs possess rich internal dynamics, such as temporal precision, refractoriness, adaptation, and synchronisation, but standard training signals do not fully exploit these properties. The mismatch between event-driven inference and training supervision creates practical bottlenecks: slow or unstable convergence, limited gains under distribution shifts, and difficulty maintaining performance without increasing model size or energy cost. Neuromorphic platforms further constrain memory traffic, bandwidth, and on-chip state. A useful learning signal should be simple to compute from activity already produced during the forward pass, avoid storing long histories, scale with depth and width, and interoperate with modern architectures. The central problem addressed in this thesis is the lack of a scalable, biologically grounded learning signal for deep SNNs that captures population-level synchrony, provides robust credit assignment, and respects compute and memory constraints. This gap prevents current SNNs from fully leveraging their distinctive internal dynamics while meeting practical requirements of stability, efficiency, and deployability.

1.3 Research Questions

We frame this thesis around four questions at the intersection of neuroscience and scalable training for deep spiking networks.

(1) *Does group-level spike coordination carry task-relevant signal, and can that coordination be turned into a practical training cue for deep SNNs?*
(2) *Can synchrony-derived signal function as a regulariser for SNNs, limiting overfitting without degrading accuracy or training stability?*
(3) *Can plasticity mechanisms drawn from neuroscience be integrated into mainstream deep-learning structure while preserving scalability and efficiency?*
(4) *Can well-optimised SNNs close or even surpass the performance of Artificial Neural Network (ANN) baselines on representative benchmarks while retaining their low-energy advantage?*

1.4 Thesis Organization

This thesis examines brain-inspired synchrony signals for training deep spiking neural networks. The document is organised into five chapters as follows.

(1) **Chapter 1: Introduction**

Establishes the context and motivation, defines the problem to be addressed, states the research questions, and summarises the contributions. It motivates the need for lightweight, brain-inspired learning cues.

(2) **Chapter 2: Literature Review**

Reviews fundamentals of spiking computation and constraints typical of neuromorphic systems. It examines training routes for SNNs, including surrogate-gradient methods and pairwise timing rules such as STDP, and highlights their limits in depth, credit assignment, and memory demand. Evidence from neuroscience on population synchrony and neuromodulatory influence is synthesised.

(3) **Chapter 3: Learning with Spike Synchrony in Spiking Neural Networks**

Introduces SSDP as a training-time regulariser. The chapter defines a short-window co-activation statistic, describes post-optimisation updates on a sparse subset of weights that leave the forward path unchanged, and analyses the compute and memory footprint. Experiments span single-layer SNNs to spiking transformers, with attention to convergence behaviour and tolerance to spike-time jitter and event noise.

(4) **Chapter 4: Synchrony-Gated Plasticity with Dopamine Modulation for Spiking Neural Networks**

Presents a dopamine-like, loss-aware variant of SSDP termed Dopamine-Modulated Spike-Synchrony-Dependent Plasticity (DA-SSDP). A batch-level synchrony measure is coupled to a learned gate obtained after a brief warm-up; thereafter, the gate simply rescales the local update applied after backpropagation in late layers.

(5) **Chapter 5: Conclusion and Future Work**

Concludes the study by summarising what is gained and what remains open. Limitations are stated, and next steps are outlined: scaling to larger vision and Electroencephalogram (EEG) corpora and longer sequences, closer co-design with neuromorphic hardware and on-chip prototypes, and extensions to online or continual learning with potential clinical applications.

1.5 List of Publications

Published / Preprint

(1) Y. Tian, A. Kembay, N. D. Truong, J. K. Eshraghian, and O. Kavehei, *Learning with Spike Synchrony in Spiking Neural Networks*, arXiv:2505.14841, 2025. doi:10.48550/arXiv.2505.14841.
Status: be submitted.

(2) Y. Tian, J. K. Eshraghian, N. D. Truong, and O. Kavehei *Synchrony-Gated Plasticity with Dopamine Modulation for Spiking Neural Networks*
Status: accepted by Transactions on Machine Learning Research (TMLR).

List of Source Codes

The code has been released publicly with the MIT license to strengthen the studies and scholarly influence, and the code is available at: https://github.com/NeuroSyd/SSDP and https://github.com/NeuroSyd/DA-SSDP.

CHAPTER 2

Literature Review

2.1 What is a Spiking Neural Network?

Spiking Neural Networks (SNNs) is a kind of artificial neural model in which information is exchanged through discrete spikes rather than continuous-valued activations. Each neuron carries an internal state that evolves over time, commonly called the membrane potential, and it emits a spike (a binary event) once this state crosses a chosen threshold. This all-or-none response mimics the action potentials observed in biological neurons. Due to their event-driven and temporally structured communication, SNNs are frequently described as the 'third generation' of neural networks, following perceptron-style models and conventional Artificial Neural Networks (ANNs) with continuous activations (Maass 1997).

2.1.1 Neuron Model

A typical spiking neuron integrates synaptic input while its voltage leaks back toward a resting level; once the threshold is reached, it fires. A widely used example is the Leaky Integrate-and-Fire (LIF) neuron, whose subthreshold dynamics are captured by

$$\tau \frac{dV(t)}{dt} = -\big[V(t) - V_{\text{rest}}\big] + RI(t), \tag{2.1}$$

where $V(t)$ denotes the membrane potential, V_{rest} the resting level, $I(t)$ the input current, R the membrane resistance, and $\tau = RC$ the membrane time constant. When $V(t)$ reaches the threshold V_{th}, the neuron fires and the potential is reset, often to V_{rest}. Spikes are impulse-like events whose information content is conveyed by their timing (or binary count) rather than amplitude. After a spike, the neuron resumes integrating, giving rise to temporal coding schemes such as latency- and rate-based representations (Gerstner and Kistler 2002; Lapicque 1907).

2.1.2 Network Communication

In SNNs, communication is asynchronous. Spikes propagate through synapses and evoke brief postsynaptic currents or voltage transients in downstream neurons. Because spikes occur only when needed, large portions of the network remain silent, which reduces overall activity compared with Artificial Neural Network (ANN) that update an analog output at every step. Whether the simulation uses continuous time or a fine discrete grid, the network state depends on the accumulated history of spikes, so the dynamics are inherently temporal. Information is therefore processed in space and in time, which means that which neurons are active matters, and so does the moment at which they fire. This setting supports temporal summation and adaptation of biological circuits (Rieke et al. 1997; Singer 1999a).

2.2 Why Spiking Neural Networks?

Several considerations motivate research on SNNs and their deployment in neuromorphic computing.

Biological Plasticity Rationale: Spiking networks are naturally suitable for biologically grounded plasticity (Frémaux and Gerstner 2015). Because synapses in SNNs receive discrete pre- and post-synaptic spikes along with slower membrane or calcium traces, they can realize local two- or three-factor plasticity rules. These mechanisms maintain eligibility traces (local synaptic states that integrate recent activity and decay over time), allowing the network to bridge delayed modulatory or teaching signals. Through such dynamics, SNNs encode both spike timing and latency, while consolidation is gated by task-related or neuromodulatory factors (Frémaux and Gerstner 2015; Bellec et al. 2020b; Izhikevich 2007b). This mechanism supports online and continual learning (Bellec et al. 2020b), and it combines multi-timescale adaptation by combining Spike-Timing-Dependent Plasticity (STDP)-like updates with homeostatic regulation (Turrigiano 2012; Zenke et al. 2017; Benna and Fusi 2016). Event-driven sparsity further reduces unnecessary synaptic updates and applies plastic changes to active networks, which improves stability and aligns with in-situ learning on neuromorphic hardware (Indiveri and Liu 2015a; Davies et al. 2018a).

Temporal Information Processing: Neurons integrate over time, respond to oscillatory or sequential structure without adding separate recurrent layers, since neuronal state already provides temporal

memory. These properties make SNNs well-suited for speech and audio processing, for electrophysiology such as Electroencephalogram (EEG), and for other tasks where timing is informative. They also align with event-based sensors that emit asynchronous spikes rather than frame-based data, enabling direct ingestion of event streams and avoiding frame reconstruction consumption (Lichtsteiner et al. 2008; Gallego et al. 2020).

Energy Efficiency: Under event-driven characteristics, inactive neurons incur negligible dynamic energy, and measured energy per operation can be observably lower than conventional accelerators when activity is sparse (Esser et al. 2016; Merolla et al. 2014). This behaviour is especially useful in embedded and edge computing platforms. Empirical studies on platforms such as TrueNorth, Loihi, SpiNNaker, and BrainScaleS report energy advantages in sparse regimes, which is valuable for low-power and event-driven applications (Merolla et al. 2014; Davies et al. 2018b; Furber et al. 2014; Schemmel et al. 2010).

Potential for New Learning Paradigms: Spike-based communication uses local learning mechanisms inspired by neuroscience, including Hebbian's ideas and timing-dependent rules. Such plasticity can operate without labels or with weak supervision, supporting continual, self-organised learning on data streams (Markram et al. 1997; Bi and Poo 1998; Diehl et al. 2015; Kheradpisheh et al. 2018). In principle, spike sparsity and discrete events may also reduce computation during inference and accelerate updates on appropriate hardware.

Taken together, SNNs seek to capture elements of the brain's efficiency and adaptability. By exploiting event-driven signaling and temporal structure, they offer a complementary computational model that can be competitive in conditions where power is limited or where temporal structure and event sparsity are important to the task.

2.3 Limitations of SNNs

Even with notable strengths, spiking neural networks face several practical and methodological difficulties when set alongside mainstream deep learning (Pfeiffer and Pfeil 2018; Tavanaei et al. 2019; Neftci et al. 2019).

Training and Optimization : The spike generation nonlinearity is discontinuous and therefore not directly applied to gradient-based optimisation (Zenke and Ganguli 2018). Early studies largely

adopted local plasticity rules such as STDP, but these approaches seldom scale reliably to deep architectures or demanding tasks (Caporale and Dan 2008). There are two principal approaches to address this problem. The first is ANN to Spiking Neural Network (SNN) conversion, where a conventional network is trained and its activations are mapped to spike rates (Cao et al. 2015; Diehl et al. 2015; Rueckauer et al. 2017). The second is surrogate gradient training, which introduces a smooth proxy for the spike function to enable backpropagation. Each strategy involves trade-offs: conversion tends to bias models toward rate codes and will not be able to take full advantage of temporal structure, while surrogate methods must balance spiking dynamics against numerical stability. Problems such as gradient vanishing or exploding temporal gradients persist, and the optimisation toolbox is not yet as mature as that for ANNs (Pascanu et al. 2013).

Performance Gap: On widely used benchmarks, particularly in static vision, SNNs have historically fallen behind strong ANN baselines, with training difficulty and limited exploitation of depth being major reasons (Tavanaei et al. 2019). Although recent work has reduced this gap in several settings (Fang et al. 2021; Zhou et al. 2023), achieving competitive accuracy often depends on careful hyperparameter tuning, longer simulation timestep, or increased model capacity, which can diminish expected gains in energy and latency (Pfeiffer and Pfeil 2018; Sengupta et al. 2019).

Latency and Throughput: Because information is integrated over time, the decision process can be slower, especially under rate coding where many spikes are needed to represent values robustly (Brette 2015). Encoding by time to first spike can reduce latency but typically increases training complexity and may introduce sensitivity to noise (Mostafa 2017). As networks go deeper, unrolling through time becomes similar to recurrent computation, which can slow both training and inference unless architecture and schedule are designed carefully (Pascanu et al. 2013).

Data Encoding and Interfaces: Translating raw inputs into spike trains is itself an important design choice (Brette 2015; Dayan and Abbott 2001; Gerstner and Kistler 2002). Rate, temporal, and population codes each bring distinct trade-offs between fidelity and efficiency and add extra hyperparameters (Bohte et al. 2002). High-resolution signals may require more spikes or large neuron populations, raising energy and latency if the encoder is not well matched to the task. On the output side, converting spike activity into decisions or continuous quantities adds further system-level complexity.

Hardware and Software Tooling: Access to specialised neuromorphic chips is still limited, and many studies rely on Central Processing Unit (CPU) or Graphics Processing Unit (GPU) simulation

where irregular event-driven computations are not always efficient (Merolla et al. 2014; Davies et al. 2018b; Furber et al. 2014; Schemmel et al. 2010). Tooling for SNNs has improved in recent years, yet in general it is less polished than the libraries used for ANNs in terms of usability, optimisation options, and the breadth of supporting resources and societies (Bekolay et al. 2014; Stimberg et al. 2019; Eshraghian et al. 2023; Rueckauer et al. 2017).

Viewed as a whole, the points above outline a clear gap between what spike-based computation promises and what researchers can readily achieve today with standard deep learning methods. Narrowing this gap is likely to require progress on several parts, including learning rules, model architectures, input and output encoding, and closer hardware-software co-design (Frémaux and Gerstner 2016).

2.4 What is Synaptic Plasticity?

Synaptic plasticity is the activity-driven change in synaptic strength and is a central basis of learning and memory (Gerstner and Kistler 2002). Depending on use and timing, a connection can become stronger or weaker. The idea traces back to Hebb (Hebb 2005a), who argued that a synapse should strengthen when pre-synaptic activity reliably contributes to post-synaptic firing, often summarised as "neurons that fire together, wire together".

Modern work refines this principle with STDP, which encodes temporal order (Bi and Poo 1998; Markram et al. 1997; Caporale and Dan 2008). When a pre-synaptic spike arrives shortly before a post-synaptic spike, the synapse tends to strengthen, a form of long-term potentiation. When the pre-synaptic spike follows the post-synaptic spike, the synapse tends to weaken, a form of long-term depression. In practical terms, synapses that consistently help to trigger post-synaptic spikes are reinforced, whereas inputs that arrive too late are reduced.

Plasticity operates over multiple time scales (Zucker and Regehr 2002). Short-term effects adjust efficacy over milliseconds to minutes, as in potentiation and depression. Long-term changes can last hours or longer and involve biochemical signalling and structural modification. In addition to correlation-driven updates, neuromodulators such as dopamine can scale, gate, or even reverse plastic changes based on outcomes, thereby linking local synaptic adjustments to a global feedback signal (Izhikevich 2007a).

Within SNNs, these mechanisms are attractive because they allow learning without explicit labelled error signals. STDP has been used for unsupervised pattern discovery and association (Masquelier and Thorpe 2007; Diehl and Cook 2015), although by itself it struggles to train deep architectures to state-of-the-art accuracy on complex tasks. Hybrid approaches that combine local plasticity with a supervised backpropagation signal, or that incorporate modulatory signals, have been explored to improve scalability (e.g., Neftci et al. 2019; Bellec et al. 2020a). From a hardware perspective, implementing local plasticity on chip is appealing for online adaptation at the edge (Davies et al. 2018b).

2.4.1 How SSDP Differs from Classical Spike-Based Rules?

Classical spike-based rules instantiate Hebb's intuition in different operational forms. STDP encodes temporal order by strengthening synapses when pre-synaptic spikes reliably precede postsynaptic spikes and weakening them for the reverse order, and triplet or pattern-based extensions incorporate higher-order spike patterns to better match frequency-dependent observations while retaining millisecond sensitivity (Bi and Poo 1998; Markram et al. 1997; Caporale and Dan 2008; Pfister and Gerstner 2006). Rate-based formulations such as Bienenstock-Cooper-Munro (BCM) emphasise postsynaptic activity with a sliding threshold that stabilises learning over longer windows (Gerstner and Kistler 2002). Three-factor variants, such as reward-modulated STDP and eligibility-trace methods, such as e-prop, attach a task-level modulator to local traces to address temporal credit assignment. (Izhikevich 2007a; Bellec et al. 2020a; Neftci et al. 2019).

Spike-Synchrony-Dependent Plasticity (SSDP) departs from both pairwise timing and single-neuron rate by making the direction of synaptic change depend on whether pre- and postsynaptic co-fire consistently across the set of timesteps under consideration, rather than on the exact timing of pre-post delay. When neurons are co-firing throughout the timesteps, the connection is strengthened; when spikes are separated in time or only intermittently overlap, potentiation effectively vanishes, and connections that fail to co-fire across the timesteps are depressed. In other words, SSDP rewards sustained co-firing and penalises mismatched activity, enforcing a spiking synchrony criterion instead of a pairwise temporal-ordering criterion. This mechanism emphasises ensemble synchrony aligns with systems-level information that effective neural communication is transmitted by rhythmic coordination and coherence across cell assemblies rather than by isolated pairwise relations (Fries 2015; Buzsáki and Draguhn 2004).

2.5 Discussion

SNNs sit at the interface of neuroscience and machine learning. Their event-driven, temporally rich computation offers a route to energy-efficient, adaptive intelligence, yet training stability, accuracy, latency, encoding, and tooling remain active challenges. A productive direction is to look for learning signals that respect locality and low traffic while providing reliable credit assignment at scale. Methods that exploit coordination across neuron populations, possibly shaped by simple global modulators, aim to leverage the advantages of spiking dynamics without incurring high computational or memory costs. Continued progress is likely to come from joint design of rules, architectures, and hardware.

CHAPTER 3

Learning with Spike Synchrony in Spiking Neural Networks

The content presented in this chapter is submitted, titled *Learning with Spike Synchrony in Spiking Neural Networks*

(1) Y. Tian, A. Kembay, S. Tensingh, N. D. Truong, J. K. Eshraghian, and O. Kavehei, *Learning with Spike Synchrony in Spiking Neural Networks*, arXiv:2505.14841, 2025. doi:10.48550/arXiv.2505.14841. Status: be submitted.

3.1 Abstract

Spiking Neural Networks (SNNs) promise energy-efficient computation by mimicking biological neural dynamics, yet existing plasticity rules focus on isolated spike pairs and fail to leverage the synchronous activity patterns that drive learning in biological systems. We introduce Spike-Synchrony-Dependent Plasticity (SSDP), a training approach that adjusts synaptic weights based on the degree of synchronous neural firing rather than spike timing order. Our method operates as a local post-optimisation mechanism that applies updates to sparse parameter subsets, maintaining linear-time computational efficiency. SSDP also serves as a lightweight event-structure regulariser, biasing the network toward a biologically plausible way. SSDP integrates with backpropagation while keeping the forward computation graph unchanged. We validate our approach across single-layer SNNs and spiking Transformers on datasets from static images to high-temporal-resolution tasks, demonstrating improved convergence stability and enhanced robustness to spike-time jitter and event noise. By incorporating synchrony-dependent mechanisms, SSDP advances the way artificial neural networks can integrate biological principles for more robust and efficient learning.

Keywords: Spiking neural networks, Synaptic plasticity, Neuromorphic computing, Deep learning, Plasticity rule for deep networks, Brain-inspired model.

3.2 Introduction

The brain processes information through sparse, event-driven spikes, which allows it to perform robust, energy-efficient calculations in changing environments (Stuijt et al. 2021; Malyshev et al. 2013; Lewicki 2002; Olshausen and Field 1996). Inspired by this biological principle, SNNs (Maass 1997; Zenke and Vogels 2021; Tavanaei et al. 2019; Sharifshazileh et al. 2021) imitate the spike-based communication used by real neurons. Unlike traditional Artificial Neural Networks (ANNs), which rely on continuous-valued signals, SNNs use precise binary spikes that occur at specific times, forming a fundamentally different way of computing (Gütig and Sompolinsky 2006). This brain-inspired architecture supports sparse coding and asynchronous updates, making SNNs naturally energy-efficient and well-suited for neuromorphic hardware implementation (Indiveri and Liu 2015b). However, training SNNs remains challenging because spikes are discontinuous and non-differentiable, often resulting in lower performance compared to standard deep learning models (Neftci et al. 2019; Eshraghian et al. 2023; Pfeiffer and Pfeil 2018).

To address the limitations of gradient-based methods in SNNs, researchers have turned to biologically inspired synaptic plasticity rules. Among these, Hebbian learning first introduced the fundamental idea that correlated neural activity strengthens connections (Hebb 2005b). Despite its biological relevance, classical Hebbian plasticity overlooks the precise temporal structure of spikes and lacks sensitivity to causality in spike timing. Spike-Timing-Dependent Plasticity (STDP) was first experimentally observed by Bi and Poo (Bi and Poo 1998), who reported that the relative timing between pre- and post-synaptic spikes modulates synaptic strength. Although STDP is usually described using smooth exponential curves, the original experimental data were quite noisy, meaning these curves represent approximate trends rather than exact biological rules. Nevertheless, STDP has become widely adopted as a temporal learning mechanism in SNNs due to its biological inspiration and simplicity. It has been applied to various vision and classification tasks (Liu et al. 2021; Goupy et al. 2024b; Goupy et al. 2024a). Reward-modulated Spike-Timing-Dependent Plasticity (R-STDP) Quintana et al. 2022 incorporates global reward signals (analogous to dopamine) to gate synaptic updates, achieving biological plausibility but limited by coarse and delayed reinforcement signals. Supervised STDP approaches, such as Supervised Spike-Timing-Dependent Plasticity (SSTDP) (Liu et al. 2021) and Stabilized Supervised Spike-Timing-Dependent Plasticity (S2-STDP) (Goupy et al. 2024a), rely on precise, externally provided spike-time errors, enabling fine-grained synaptic adjustments at the

expense of biological realism due to their dependence on supervised signals. More recently, STDP-inspired Temporal Local Learning Rule (S-TLLR) (Apolinario and Roy 2023) introduced a generalised three-factor eligibility trace (a per-synapse local state that accumulates recent pre/post activity and decays over time to bridge delayed modulatory/teaching signals) that captures both causal and non-causal spike-time interactions, achieving good scalability and lower computational demands. However, STDP-style rules face two main limitations. Firstly, it generally focuses only on individual synapses and pairs of spikes, ignoring more complex collective spiking patterns. Second, its sensitivity to exact spike timing makes it vulnerable to noise and instability in high-frequency or dynamic environments (Popovych et al. 2013). The data by which STDP is derived is considerably noisier than the exponential curve fits. From an engineering standpoint, order-based, time-dependent rules also impose noise and memory complexity in deep networks: pair enumeration scales with the product of pre- and post-synaptic spike counts, and even practical online implementations reduce this cost by maintaining per-synapse eligibility traces and event-driven updates, and the cost grows with depth, fan-in/fan-out, and batch size (Clopath et al. 2010; Pfister and Gerstner 2006; Florian 2007; Pfeiffer and Pfeil 2018). Moreover, purely local timing rules perform poorly at deep credit assignment, making stable supervised training in multi-layer settings difficult without additional mechanisms (e.g., Zenke and Ganguli 2018; Pfeiffer and Pfeil 2018). These limitations contrast with growing biological evidence that neuronal synchrony plays a vital role in efficient information processing and learning.

Synchrony among neurons is widely observed across brain regions during cognitive processes such as memory recall, sensory binding, and attention modulation (e.g., Engel et al. 2001; Singer 1999b). This synchronised activity is not accidental; it is believed to play a key role in neural coding by linking distributed neural activities and strengthening the reliability of neural communication (Brette 2012; Arenas et al. 2008). Both theoretical and experimental studies indicate that neural synchrony improves response reliability, enhances plasticity thresholds, and contributes to stable attractor dynamics during learning (Baruchi et al. 2008; Kembay et al. 2025; Huguenard and McCormick 2007). Moreover, studies have demonstrated that precisely timed synchrony among groups of neurons can trigger long-term potentiation (LTP) even in the absence of pairwise spike causality (Anisimova et al. 2023; Frey and Morris 1997; Markram et al. 1997). These findings challenge the sufficiency of pair-based STDP rules and suggest that collective spike patterns, rather than isolated spike timings, might be more biologically relevant for driving synaptic changes (KRUEGER and OBÄL JR 1993; Patel et al. 2023). In other words, real-world learning likely depends not only on individual spike pair relationships but

also on group-level synchrony that reflects structured, meaningful input patterns (KRUEGER and OBÄL JR 1993; Patel et al. 2023).

From an engineering perspective, modern deep Spiking Neural Network (SNN) training still relies mainly on surrogate gradients; common regularisers shape weights or rates, neglecting the event structure of spikes. Pairwise timing rules incur fine-grained time bookkeeping and state storage that scale poorly with spike density and depth, and their updates are not explicitly aligned with supervised objectives. In addition, many high-performing STDP-based systems depend on convolutional structures (Lee et al. 2016; Roy et al. 2019). While convolutional weight sharing is effective, it is only an approximate analogue of biological connectivity and can be inefficient on neuromorphic hardware due to dense sharing and memory traffic (Singer 1999b; Uhlhaas et al. 2009; Harris and Gordon 2015; Harris 2005). This motivates the search for learning mechanisms and feature-formation strategies that leverage population-level events and are more readily adaptable to neuromorphic chips. To fill this gap, we propose **S**pike-**S**ynchrony-**D**ependent **P**lasticity (SSDP), a minimal training-time local rule that turns population synchrony into a structure-aware signal. SSDP adjusts synaptic weights according to the degree of coordinated pre-post firing within a short window, using a symmetric Gaussian kernel so that near-coincident spikes still contribute to potentiation. The rule complements backpropagation rather than replacing it; updates are clipped and applied to a sparse parameter subset after the optimiser step; and the computational overhead scales linearly with neuron counts via broadcasted outer products rather than pair enumeration. This design targets three engineering goals: (i) steadier convergence with lower loss variance, (ii) more structured representations, and (iii) robustness to spike-time jitter and event noise, all at negligible compute and memory overhead.

We systematically evaluate SSDP across diverse tasks and network architectures, showing that it enhances learning stability, reduces convergence volatility, and improves classification accuracy. We further integrate SSDP into deeper architectures, including Spiking ResNet18 and Transformer-based spiking models, and trained deep spiking neural networks on challenging classification tasks, spanning static images, event-driven data, and datasets with complex temporal structure such as Spiking Heidelberg Digits (SHD) and Spiking Speech Commands (SSC) (Cramer et al. 2020a). Together, these results establish SSDP as a biologically plausible and generalizable training signal for deep SNNs.

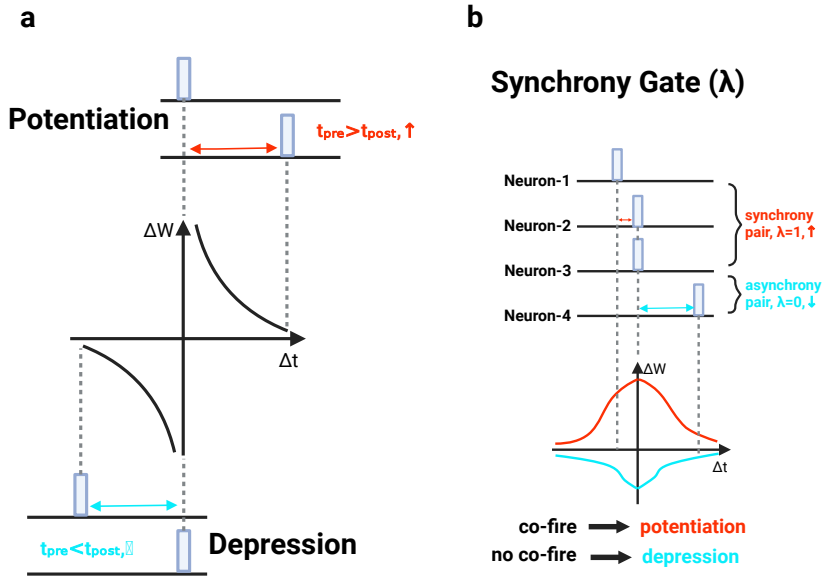


FIGURE 3.1: **STDP versus SSDP. (a) Classical STDP.** For a synapse $i \to j$, the signed delay $\Delta t = t_{\text{post}} - t_{\text{pre}}$ sets the *sign* of the update: pre→post ($\Delta t > 0$) yields potentiation, post→pre ($\Delta t < 0$) yields depression; the magnitude decays with $|\Delta t|$ (two-sided, order-sensitive window; schematic). **(b) SSDP.** A binary *synchrony gate* $\lambda = P_j Q_i$ (AND of post/pre spike flags) marks whether both neurons fire at least once within the same sample window. If $\lambda = 1$ the pair is potentiated; if $\lambda = 0$ a small depression is applied. The update magnitude is scaled by a *symmetric Gaussian* $g(|\Delta t|)$ centred at 0 using the *first-spike* times, so the rule is insensitive to firing order and rewards temporal coincidence rather than causality. SSDP thus needs only one bit and one timestamp per neuron.

3.3 Methods

Preliminaries.

We consider a layer that connects C_{in} presynaptic neurons to C_{out} postsynaptic neurons. Within a simulation window of length T (time steps $t = 0, \dots, T-1$) and for a minibatch of size B, we record the binary spike flags

$$Q_{b,i} \;=\; \mathbb{1}\big[x^{\text{pre}}_{b,i}(t) > 0 \text{ for some } t\big], \quad P_{b,j} \;=\; \mathbb{1}\big[x^{\text{post}}_{b,j}(t) > 0 \text{ for some } t\big], \tag{3.1}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function (taking the value 1 if the condition holds and 0 otherwise), $x^{\text{pre}}_{b,i}(t)$ and $x^{\text{post}}_{b,j}(t)$ denote the membrane outputs of neuron i and j for sample $b \in \{1, \dots, B\}$. In

addition, we store the first-spike times

$$t^{\mathrm{pre}}_{b,i} \;=\; \min\{t \mid x^{\mathrm{pre}}_{b,i}(t) > 0\}, \qquad t^{\mathrm{post}}_{b,j} \;=\; \min\{t \mid x^{\mathrm{post}}_{b,j}(t) > 0\}, \tag{3.2}$$

and set $t^{\mathrm{pre}}_{b,i} = t^{\mathrm{post}}_{b,j} = T$ if the neuron is silent. Using $\{0,1\}$ spike flags focuses the update on co-activation rather than raw firing rates, which makes the rule invariant to uniform rate scaling and less sensitive to noise. Recording only first-spike times provides a compact temporal anchor that preserves causal ordering with $O(BC)$ memory(B is batch-size and C is channel) instead of $O(BCT)$ (T is the timestep), and it still supports a coincidence kernel based on time differences. In implementation, Q and P map to binary tensors, while t^{pre} and t^{post} are cached during the forward pass; silent units default to T, which yields large $|\Delta t|$ and therefore negligible coincidence strength in the kernel used later.

Synchrony gate.

For every pair (j, i) we introduce a binary synchrony gate

$$\lambda_{b,j,i} \;=\; P_{b,j}\, Q_{b,i}, \tag{3.3}$$

which equals 1 if and only if both neurons emit at least one spike within the same sample, and 0 otherwise. This gate is implemented by a broadcasted outer product of the spike-flag vectors, producing a tensor of shape $[B, C_{\mathrm{out}}, C_{\mathrm{in}}]$. It amounts to an AND operation over the per-neuron temporal OR-pools $P_{b,j}$ and $Q_{b,i}$: $\lambda_{b,j,i} = 1$ only when both neurons spike at least once within the timestep, and 0 otherwise. The gate itself is not related to time once firing occurs; temporal proximity and tolerance to small jitter are governed by the coincidence kernel applied later to $|\Delta t| = |t^{\mathrm{post}}_{b,j} - t^{\mathrm{pre}}_{b,i}|$ with a learnable bandwidth σ. The operation is fully vectorised with $O(B\, C_{\mathrm{out}} C_{\mathrm{in}})$ complexity and negligible extra memory, which matches the Graphics Processing Unit (GPU) implementation and enables stable per-minibatch averaging of the updates.

Temporal proximity.

The absolute delay between the first spikes is

$$\Delta t_{b,j,i} \;=\; \left|t^{\mathrm{post}}_{b,j} - t^{\mathrm{pre}}_{b,i}\right|, \tag{3.4}$$

and is converted into a smooth coincidence weight

$$g_{b,j,i} \;=\; \exp\!\left(-\tfrac{\Delta t^2_{b,j,i}}{2\sigma^2}\right), \tag{3.5}$$

where $\sigma > 0$ controls the temporal spread of the window. The kernel rewards near-synchronous activity while decaying continuously with increasing $|\Delta t|$, so updates change gradually rather than flipping at a hard boundary. Because g is an even and smooth function with a finite slope at $\Delta t = 0$, the rule avoids the discontinuities of rectangular windows and the cusp of Laplacian forms, which stabilises optimisation with respect to σ and prevents sharp jumps in the update when spike times jitter. The scale parameter σ directly sets the coincidence tolerance: larger values widen the effective window and smaller values emphasise precise alignment; for reference, the weight drops to $1/\mathrm{e}$ at $|\Delta t| = \sigma\sqrt{2}$ and to half its peak at $|\Delta t| = \sigma\sqrt{2\ln 2}$. Because $g_{b,j,i}$ depends on $|\Delta t|$ only, the rule is insensitive to causal order and explicitly targets temporal coincidence instead of pre-post ordering. Given (3.3)–(3.4), the instantaneous weight increment for sample b is

$$\Delta w_{b,j,i} \;=\; \big(A_+\,\lambda_{b,j,i} - A_-\,[1-\lambda_{b,j,i}]\big)\,g_{b,j,i}, \tag{3.6}$$

so synchronous pairs ($\lambda_{b,j,i} = 1$) are reinforced in proportion to the Gaussian kernel, whereas asynchronous pairs receive a mild depression.

Batch aggregation and update.

The sample-level increments are averaged over the minibatch and applied in place:

$$\Delta W_{j,i} \;=\; \frac{1}{B}\sum_{b=1}^{B}\Delta w_{b,j,i}, \qquad W_{j,i} \;\leftarrow\; W_{j,i} + \Delta W_{j,i}. \tag{3.7}$$

Averaging over the batch reduces gradient-like variance in the local update and makes the rule less sensitive to outlier samples with abnormal spike timing. In practice, the implementation obtains a matrix of shape $[C_{\text{out}}, C_{\text{in}}]$, clips it to a bounded range to guard against rare but large coincident bursts, and then performs an in-place addition to the target weight. Updates are applied only after an initial warm-up phase, so backpropagation can establish a reasonable operating point before the synchrony prior is introduced, improving stability without affecting the inference-time computation graph or latency. This fully vectorised, local, and online application adds negligible training-time overhead while preserving the forward pass and optimiser state, making the procedure easy to integrate into modern architectures.

Relation to gradient descent.

Equations (3.1)–(3.7) are executed once per iteration after the optimiser has performed its parameter update. This ordering leaves gradient flow and the optimiser's internal state untouched within the same iteration, while still modifying the weights for the next one. In practice, SSDP can also act

as a lightweight, local regulariser that biases synapses toward spatio-temporal co-activation without altering the forward computation or inference-time cost. Warm-up gating delays the application of the rule until the network has reached a sensible operating point under pure Backpropagation (BP), thereby avoiding early-stage drift and preserving the same convergence behaviour empirically observed with Stochastic Gradient Descent (SGD)/Adaptive Moment Estimation (Adam). Because the update is local and uses batch-mean aggregation, it integrates cleanly with standard training loops and co-exists with weight decay or other global regularisers.

Complexity and stability

Let B be the minibatch size and $C_{\text{in}}, C_{\text{out}}$ the numbers of pre- and post-synaptic units in the plastic layer. First-spike times are denoted by t^{pre} and t^{post}, and binary spike flags by $Q_{b,i} \in \{0,1\}^{B\times C_{\text{in}}}$ and $P_{b,j} \in \{0,1\}^{B\times C_{\text{out}}}$. Each SSDP update is built from fully vectorised broadcast operations rather than explicit event-pair enumeration. The synchrony gate arises from a single broadcasted outer product $PQ^\top$ with $O(B\,C_{\text{out}}C_{\text{in}})$ work. Pairwise time differences are formed by broadcasting $|t^{\text{post}} - t^{\text{pre}}|$ over indices, followed by an element-wise Gaussian kernel; both steps have the same cost. The per-sample increments then combine the kernel with the potentiation/depression gains and are averaged across the batch to produce ΔW, again with $O(B\,C_{\text{out}}C_{\text{in}})$ complexity. Overall, one SSDP update costs $\mathcal{O}\big(B\,C_{\text{out}}C_{\text{in}}\big)$ with a small constant factor dominated by one outer product, one broadcasted subtract-and-absolute operation, one exponential, and a few fused multiplies and adds. The method stores only a single timestamp per unit and binary flags, so no factor in the average spike counts $\langle S_{\text{pre}}\rangle$ or $\langle S_{\text{post}}\rangle$ appears in either compute or memory, and complexity remains linear in neuron counts. For comparison, STDP enumerates events and scales as $\mathcal{O}\big(B\,C_{\text{out}}C_{\text{in}}\,\langle S_{\text{pre}}\rangle\langle S_{\text{post}}\rangle\big)$, which grows with spike density and the temporal window. Stability is supported by several implementation choices that follow directly from the update structure. Batch averaging reduces variance from unusual samples and prevents large, sample-specific excursions. Clipping the aggregated increment to a bounded range guards against rare but strong coincidences and outliers in Δt. The binary gate confines potentiation to co-active pairs, while the Gaussian kernel reduces the influence of large time gaps and makes the update change smoothly as spikes jitter. Delayed activation of SSDP after a short warm-up allows BP to establish a reasonable representation before the synchrony prior is introduced, thereby improving robustness without altering the inference graph. In the attention pathway, spatial dimensions are collapsed into channel-level indicators and first-spike times before applying the same rule, so the additional cost of extracting first-spike indices scales linearly and is negligible relative

to the network's forward pass. Across each integration point, updates are in-place and local, adding minimal training-time overhead while remaining compatible with standard optimiser settings.

Memory and traffic.
Beyond the model parameters, the only persistent working buffer is the batch accumulator $\Delta W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$, which determines the peak weight-space footprint. Intermediate tensors $(\lambda, \Delta t, g)$ are produced and consumed in sequence, so they can be streamed or tiled. Per-iteration traffic consists of a single read of the spike flags and first-spike indices and a single read-modify-write sweep over the target matrix; there are no per-synapse eligibility traces carried across iterations. Data types can further reduce bandwidth: timestamps remain integer, and the kernel is evaluated in the current compute data type (Automatic Mixed Precision (AMP)-friendly), so the SSDP benefit from mixed-precision speedups without casting the cached indices. In the attention pathway, caching only channel-wise binary indicators and first-spike indices (rather than feature maps) makes the retained activations scale with $O(TBC)$ and independent of $H \times W$ after the logical OR operation, which lowers both memory footprint and memory bandwidth during the SSDP step.

3.4 Results

3.4.1 SSDP scales across Tasks, Modalities, and Temporal Complexities

We evaluate SSDP extensively across multiple benchmark datasets and network architectures, including static image datasets (Fashion-National Institute of Standards and Technology database (MNIST), Canadian Institute for Advanced Research (CIFAR)-10, CIFAR-100, ImageNet), event-driven neuromorphic vision datasets (Neuromorphic National Institute of Standards and Technology database (N-MNIST), CIFAR10-Dynamic Vision Sensor (DVS)), and auditory temporal datasets (SHD, SSC). This breadth is intended to test whether a synchrony-aware training signal generalises across distinct input statistics. We expect consistent gains when co-activation emerges. Two biologically inspired temporal kernels (exponential decay and symmetric Gaussian) are implemented to explore different synchrony-sensitivity profiles in SSDP.

ImageNet-1k. ImageNet is a scale stress-test: large label space and high-resolution inputs require long training schedules, and compared with event-driven datasets, the activations in our spiking Transformer are relatively dense, making it a suitable benchmark for testing SSDP's low-overhead integration.

SSDP's update is linear in neuron counts and applied after the optimiser without altering the forward graph, so we expect negligible overhead and a regularising effect from event-structure priors at scale. With bounded, Gaussian-weighted synchrony updates as a light-weight regulariser, accuracy should improve over the backprop-only baseline while training speed remains unchanged. Table 3.1 presents ImageNet-1k results with the SpikingResformer-L backbone ($T{=}4$). Under the same settings, SSDP improves the baseline and attains competitive performance relative to recent spiking Transformers of comparable size, consistent with the expectation that a synchrony-aware training signal scales to large-vocabulary recognition.

CIFAR-10/100 and CIFAR10-DVS. We use two static image benchmarks and an event-stream dataset to test whether SSDP's benefits persist across input sparsity regimes and temporal structure. On CIFAR-10/100, spatial regularities tend to elicit repeatable channel co-activation, so synchrony-weighted updates should reinforce useful representations. On CIFAR10-DVS, events are produced by scan-converting static images; the resulting streams are sparse and bursty with weak class-aligned timing across a mini-batch, so the synchrony gate $\lambda = PQ^\top$ rarely activates at zero lag and the Gaussian factor $g = \exp\!\big(-\Delta t^2/(2\sigma^2)\big)$ is typically small—hence gains may be limited. Table 3.2 reports results using CIFAR variants of SpikingResformer: SSDP consistently matches or exceeds the BP-only baseline on CIFAR-10/100 under identical hyperparameters, while improvements on CIFAR10-DVS are modest, consistent with the weaker synchrony signal.

Temporal auditory benchmarks. SHD and SSC carry explicit time structure (onset patterns, inter-spike intervals) that repeatedly co-activate subsets of channels; under such inputs, a synchrony-weighted update should reinforce task-relevant assemblies. Table 3.3 compares Spiking Recurrent Neural Networks (SRNN)-family models on SHD and SSC. Adding SSDP to the Temporal Dendritic Heterogeneity Spiking Recurrent Neural Network (DH-SRNN) backbone improves Top-1 on both datasets while keeping the non-convolutional structure unchanged, indicating that synchrony-guided local updates transfer to temporal classification and are not tied to convolutional feature extractors.

Across large-scale static vision and temporal auditory tasks, SSDP provides consistent accuracy gains with no inference-time structural cost and minimal implementation burden. To attribute changes solely to the training signal, we hold architecture and schedules fixed across conditions. Unless otherwise stated, results are reported as Top-1 accuracy over $N{=}5$ independent seeds. For each seed, the checkpoint with the highest validation accuracy is evaluated on the test set. All comparisons use identical training schedules (epochs, batch size, optimiser, time steps) and unchanged architectures

TABLE 3.1: Classification accuracy (Top-1) on ImageNet-1K.

Method	Architecture	Param (M)	Energy (mJ)	TimeStep	Accuracy (%)
Hybrid training (Rathi et al. 2020)	ResNet-34	21.79	-	350	61.48
Spiking ResNet (Hu et al. 2021)	ResNet-50	25.56	70.934	350	72.75
Transformer	Transformer-8-512	29.68	38.340	1	80.80
Spikformer	Spikformer-8-384	16.81	7.734	4	70.24
Spikformer	Spikformer-8-768	66.34	21.477	4	74.81
Spikingformer	Spikingformer-8-512	29.68	7.46	4	74.79
Spikingformer	Spikingformer-8-768	66.34	13.68	4	75.85
Spike-driven Transformer	Spike-driven Transformer-8-768	66.34	6.09	4	76.32
SpikingResformer	SpikingResformer-L	60.38	8.76	4	78.77
Proposed	SpikingResformer-L	**60.38**	**8.89**	4	**79.35**$_{\pm 0.36}$

TABLE 3.2: Comparision on CIFAR-10, CIFAR-100, CIFAR10-DVS.

Dataset	Method	Architecture	Param (M)	TimeStep	Accuracy (%)
CIFAR-10	Spikformer (Zhou et al. 2022)	Spikformer-4-384	9.32	4	95.51
	Spikingformer (Zhou et al. 2023)	Spikingformer	-	4	95.61
	Spike-driven Transformer (Yao et al. 2024)	Spike-driven Transformer	-	4	95.6
	Transformer (Dosovitskiy 2020)	Transformer-4-384	9.32	1	96.73
	Proposed	SpikingResformer-CIFAR	**10.83**	4	**96.24**$_{\pm 0.29}$
CIFAR-100	Spikformer	Spikformer-4-384	9.32	4	77.86
	Spikingformer	Spikingformer	-	4	79.09
	Spike-driven Transformer	Spike-driven Transformer	-	4	78.4
	Transformer	Transformer-4-384	9.32	1	81.02
	SpikingResformer (Shi et al. 2024)	SpikingResformer-CIFAR	10.83	4	78.73
	Proposed	SpikingResformer-CIFAR	**10.83**	4	**79.48**$_{\pm 0.27}$
CIFAR10-DVS	Spikformer	Spikformer-4-384	9.32	16	80.6
	Spikingformer	Spikingformer	-	16	81.3
	Spike-driven Transformer	Spike-driven Transformer	-	16	80.0
	Transformer	Transformer-4-384	9.32	1	81.02
	Proposed	SpikingResformer-CIFAR	**17.31**	10	**84.5**$_{\pm 0.3}$

and parameter counts. SSDP operates only during training and introduces no inference-time structural overhead.

TABLE 3.3: Classification accuracy (Top-1) based on SNNs.

Dataset	Learning rule	Conv. structure	Accuracy (%)
SHD	BrainScaleS-2 (Cramer et al. 2022)	✗	81.6
	SRNN (Perez-Nieves et al. 2021)	✗	82.7
	SCNN (Rossbroich et al. 2022)	✓	84.8
	SRNN (Yin et al. 2021)	✗	90.4
	LSTM (Cramer et al. 2020b)	✗	89.2
	DHSRNN (Zheng et al. 2024)	✗	88.1
	DHSRNN+SSDP	✗	**89.1**$_{\pm 0.21}$
SSC	SRNN	✗	74.2
	LSTM	✗	73.1
	DHSNN	✗	82.46
	DHSNN+SSDP	✗	**82.86**$_{\pm 0.26}$

3.4.2 SSDP restructures population timing under rate control

Having established accuracy gains, we next examine how SSDP changes network dynamics in ways that could plausibly support those gains. The working hypothesis is that SSDP acts as a lightweight event-structure regulariser: it should increase near co-activation and organise population recruitment in time. To isolate coordination beyond rate, each model is compared to a time-shuffled surrogate that preserves

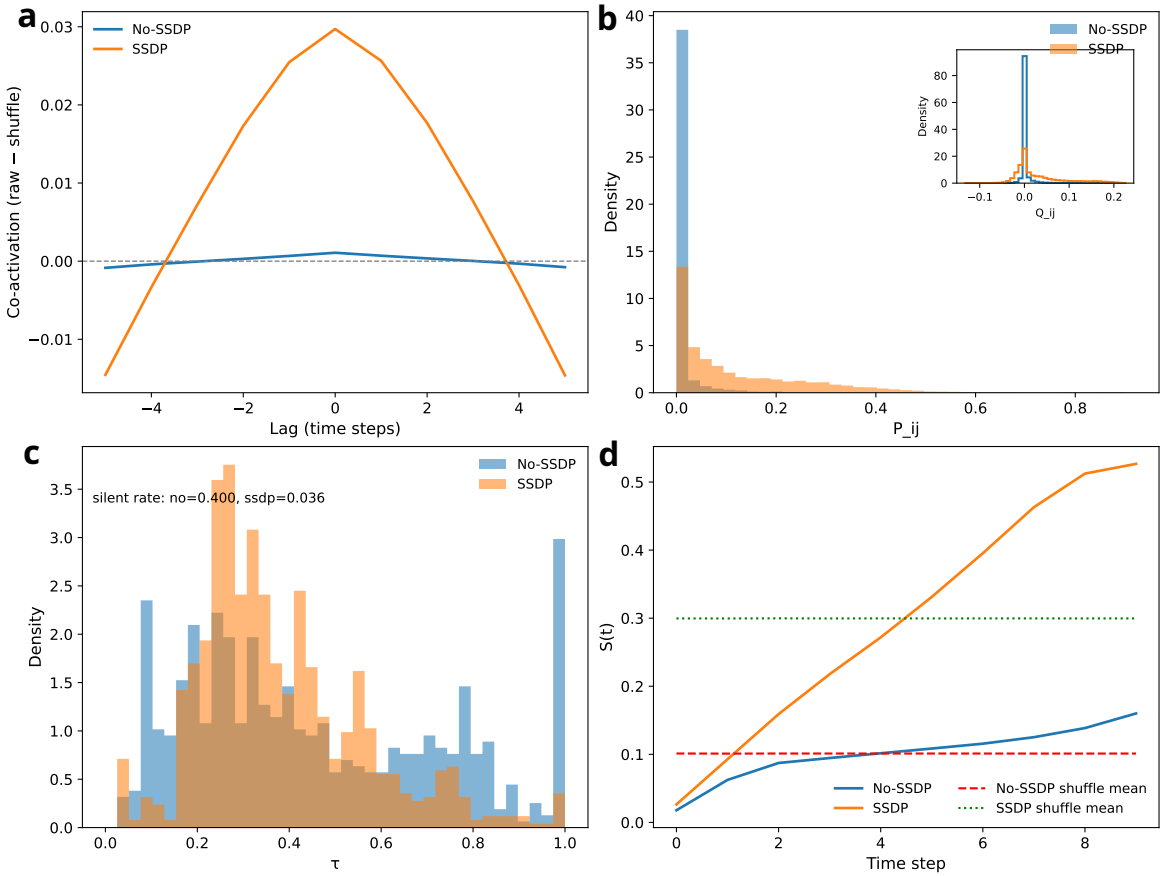


FIGURE 3.2: **(a)** Excess pairwise co-activation (raw-shuffle) versus lag; SSDP shows a sharp symmetric peak at zero lag, whereas the baseline stays near zero, indicating near-simultaneous co-activation beyond rate effects. **(b)** Distributions of zero-lag co-activation $P_{ij} = \Pr[s_i(t) = 1 \wedge s_j(t) = 1]$ across the same randomly sampled pairs (main) and the rate-corrected statistic $Q_{ij} = P_{ij} - p_i p_j$ with $p_i = \Pr[s_i(t) = 1]$ (inset); both shift to the right under SSDP, showing the increase is not explained by marginal firing rates. **(c)** Temporal selectivity for non-silent units, $\tau_j = 1 - H_j / \log T$ with H_j the entropy of the rectified, time-normalized within-unit z-score; the silent-unit fraction drops from 0.400 to 0.036, and mass concentrates at moderate τ, indicating broader recruitment with non-impulsive tuning. **(d)** Population activity $S(t) = \frac{1}{BN} \sum_{b,j} s_{b,t,j}$ for each condition together with its own time-shuffled expectation; SSDP lies consistently above its shuffle band and diverges most strongly late, indicating stronger time-localised recruitment. All statistics are computed from recorded spike tensors. Shuffle surrogates use independent circular shifts per unit for (a–b) and independent time permutations per unit for (d), preserving marginal rates while destroying cross-unit timing.

marginal rates while disrupting cross-unit timing. Fig. 3.2 aggregates four complementary readouts:

Fig. 3.2(a–b) provide rate-controlled pairwise evidence; Fig. 3.2(c–d) describe unit recruitment and its population-level manifestation.

In Fig. 3.2(a), for each lag $\ell \in [-L, L]$, we average over random neuron pairs the probability that one unit spikes at t while the other spikes at $t+\ell$, and subtract the shuffled surrogate. Under SSDP a sharp, symmetric zero-lag peak emerges, whereas the baseline remains near zero. Because the shuffle controls for marginal rates, this peak indicates same-time synchrony activities beyond rate effects. We also report the distribution of $P_{ij} = \Pr[s_i(t)=1 \wedge s_j(t)=1]$ across the same randomly sampled pairs (main) in Fig. 3.2(b), together with the rate-corrected statistic $Q_{ij}=P_{ij}-p_i p_j$ where $p_i = \Pr[s_i(t)=1]$ (inset). Both distributions shift right under SSDP, indicating that the increase in zero-lag coincidence is not explained by the product of marginal firing rates and is consistent with Fig. 3.2(a). This rightward shift in both P_{ij} and the rate-corrected Q_{ij} matches the expectation that coincidence increases beyond what marginal rates predict. In Fig. 3.2(c), for each non-silent unit we compute $m_j(t)$ (spike probability across batches), standardize it over t, truncate to the positive part, renormalize to $p_j(t)$, and set $H_j = -\sum_t p_j(t) \log p_j(t)$ and $\tau_j = 1 - H_j / \log T$ (larger τ_j indicates narrower, phase-focused activity). We expect broader recruitment with moderate temporal selectivity rather than sparse extremes. Under SSDP the silent-unit fraction drops from 0.400 to 0.036, and the non-silent population concentrates at moderate τ, indicating broader recruitment with well-defined but reasonable temporal tuning. In Fig. 3.2(c) Let $S(t) = \frac{1}{BN} \sum_{b,j} s_{b,t,j}$ denote the fraction of active units per step. For each condition, we also plot the expectation from its own time-permutation surrogate. Both conditions lie above their shuffle bands, and SSDP diverges most strongly at later steps, consistent with stronger time-localised population recruitment. Note that $S(t)$ reflects activity level rather than a pairwise synchrony metric; Fig. 3.2(a–b) constitutes the synchrony evidence, while Fig. 3.2(d) shows its population-scale expression.

Taken together, these effects are consistent with SSDP organising spikes into temporally coordinated population events—an arrangement that can reduce training-loss variance and support generalisation, as observed in our convergence analysis.

3.4.3 Representation structure and temporal precision

Linear embedding evidence.

As a linear, model-agnostic probe, Principal Component Analysis (PCA) is used to test the hypothesis

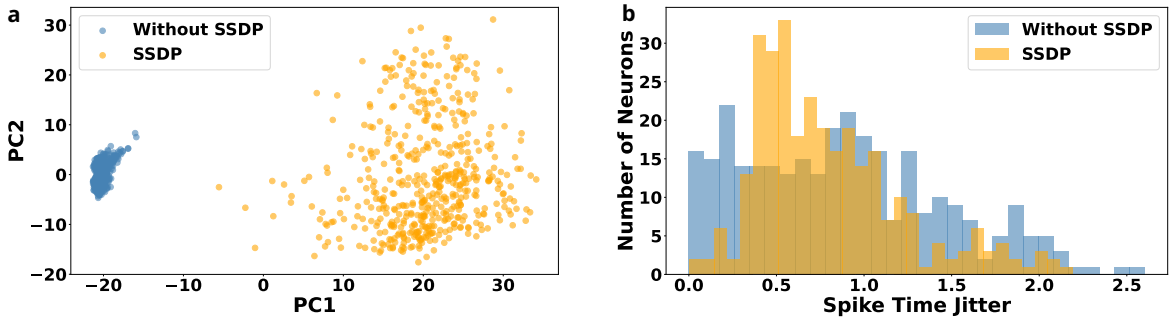


FIGURE 3.3: **(a)**: PCA of the same feature layer under matched training shows that SSDP (orange) yields a broader distribution with a pronounced principal direction compared to the baseline (blue). **(b)**: Compared with the baseline, SSDP also reduces dispersion (median 0.699 vs. 0.846, IQR 0.523 vs. 0.853) and shrinks the high-jitter tail (>1.5: 0.083 vs. 0.160; >1.0: 0.263 vs. 0.380), while avoiding near-zero jitter saturation (≈ 0: 0.003 vs. 0.040). A two-sample KS test detects a distributional shift (p=9×10^{-4}); Mann-Whitney on medians is not significant (p=0.185), consistent with changes occurring mainly in the tails and shape rather than a large central shift.

that SSDP reorganises population activity from diffuse, many-axis variability into shared, low-rank structure. If SSDP reinforces co-activation, the covariance should concentrate along a few common directions, yielding higher total variance carried by leading components and a lower participation ratio (rather than a uniform spread across weak axes). This pattern is desirable because aligned dominant modes support more stable downstream optimisation and cleaner class structure. The outcomes in Fig. 3.2(a) show that SSDP increased total variance together with a reduced participation ratio, which are consistent with this expectation and quantify the representation change induced by SSDP. Quantitatively, in the standardised feature space, the total variance increases markedly under SSDP (811.8 vs. 374.5, $\approx 2.17\times$ in this run), while the participation ratio decreases (17.19 vs. 23.33). Taken together, this indicates that SSDP does not spread variability uniformly across many weak axes; instead, it concentrates more of the variability onto a smaller set of shared principal directions, yielding a more anisotropic, low-rank structure. Consistent with this, Principal Component (PC)1 and PC2 account for roughly 0.43 and 0.06 of the variance ($\sim 49\%$ total), so the visible widening aligns with a spectrum where a few leading components carry most of the energy, and the remaining structure lies in higher dimensions.

Temporal precision (jitter).

Temporal precision is a key property of reliable population codes. To quantify it, per-neuron jitter is computed as the across-repeat standard deviation of mean spike times (300 randomly sampled neurons

per condition, fixed seed). Relative to the baseline, SSDP yields lower and tighter jitter (median 0.699 vs. 0.846; Interquartile Range (IQR) 0.523 vs. 0.853; std 0.420 vs. 0.575) and markedly suppresses the high-jitter tail (>1.5: 0.083 vs. 0.160; >1.0: 0.263 vs. 0.380). At the same time, the fraction of near-zero jitter decreases (0.003 vs. 0.040), indicating that units remain responsive rather than becoming rigid. A two-sample Kolmogorov-Smirnov test (KS-test) detects a distributional change ($p{=}9{\times}10^{-4}$), whereas the Mann-Whitney U test on medians is not significant ($p{=}0.185$) and Cliff's δ is small (-0.063), consistent with the improvement arising from tail suppression and overall tightening rather than a large median shift (Fig. 3.3(b)). Taken together with the PCA evidence (higher total variance and lower participation ratio, the result from Fig. 3.3(a)) indicates that SSDP produces population codes that are structurally organised and temporally stable, aligning with the goal of improving representation structure and temporal precision without altering the forward computation.

3.4.4 Convergence stabilisation with SSDP

Fig. 3.4 summarises optimisation on Fashion-MNIST. In Fig. 3.4(a), both models reduce training loss initially; the baseline (orange) keeps decreasing but remains oscillatory, whereas the SSDP model (blue) enters a smoother, low-variance regime mid-training and converges steadily thereafter. In Fig. 3.4(b), the SSDP model sustains a lower validation loss, while the baseline develops a widening train-validation gap – clear evidence of overfitting. Taken together with Fig. 3.2, the curves indicate that SSDP reduces loss variance and stabilises convergence, acting as a lightweight regulariser that biases learning toward repeatable, task-relevant spike-event structure.

These observations align with our objectives: (i) steadier convergence, reflected in lower loss variance and a reduced train-validation gap; (ii) more structured representations, supported by rate-controlled synchrony (Fig. 3.2) and the PCA/jitter analyses (Fig. 3.3); and (iii) greater tolerance to spike-time noise, implied by the tightened jitter distribution. The stabilising effect is consistent with the neuroscience view that population synchrony enhances communication and reduces representational noise (Fries 2005; Averbeck et al. 2006; Engel et al. 2001; Brette 2012).

3.5 Experimental Setup and Hyperparameters

Hardware Configuration

All experiments were conducted using NVIDIA RTX 4090 Laptop and Tesla V100 GPU.

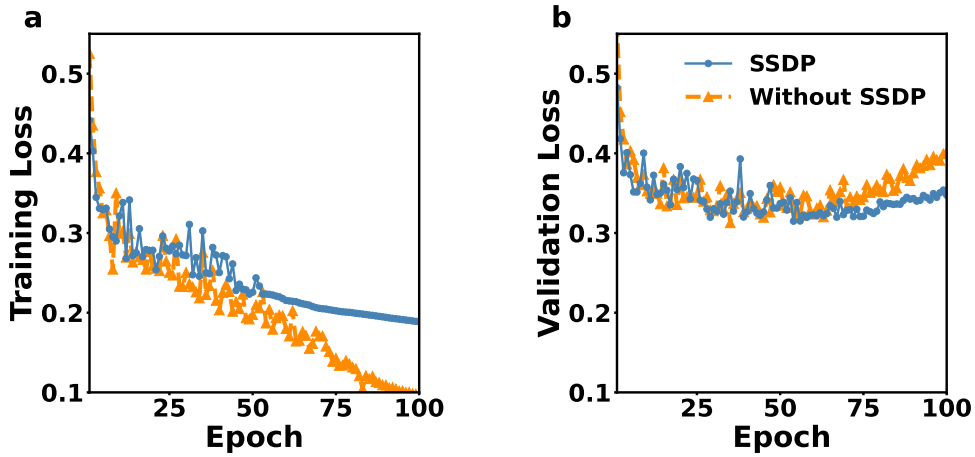


FIGURE 3.4: **(a)** Training loss On Fashion-MNIST; **(b)** validation loss on Fashion-MNIST. The SSDP model (blue) exhibits a mid-training transition to a smoother, low-variance regime, while the baseline (orange) remains more oscillatory and develops a larger train–val gap. This behaviour is consistent with SSDP's symmetric, synchrony-based post-update, which regularises training using spike-event structure.

Single Hidden Layer SNN Models

For all single hidden layer SNN models evaluated on Fashion-MNIST, N-MNIST, and CIFAR-10, we adopted the same architecture and hyperparameters. Models were trained for 100 epochs using the Adam (Kingma and Ba 2014) optimiser, with cosine annealing (Loshchilov and Hutter 2016) applied to both the learning rate and all SSDP parameters. No data augmentation was employed during training. The exponential decay variant of SSDP was used throughout, with parameters set to $A_+ = 0.02$, $A_- = 0.02$, $\tau_+ = 20.0$, and $\tau_- = 20.0$.

DH-SRNN Models on SHD and SSC

For the SHD dataset, we used a single-layer DH-SRNN with SSDP implemented via the Gaussian variant. For SSC, we adopted a two-layer DH-SRNN configuration. Both models were trained using Adam optimiser and cosine-annealed learning rate schedules for 100 (SHD) and 150 (SSC) epochs, respectively, starting SSDP updates from 10 epochs. The learning rate was set to 5×10^{-4}. The SSDP parameters for both tasks were $A_+ = 1.5 \times 10^{-4}$, $A_- = 5 \times 10^{-5}$, $A_\cdot = 5 \times 10^{-5}$, and $\sigma = 1.0$.

Spiking-ResNet18 on CIFAR-10 and CIFAR-100

For experiments using Spiking-ResNet18 architectures on CIFAR-10 and CIFAR-100, we used identical model configurations across both datasets. SSDP was configured with the exponential

variant and parameters set to $A_+ = 5 \times 10^{-4}$, $A_- = 5 \times 10^{-4}$, $\tau_+ = 20.0$, and $\tau_- = 20.0$. Training was performed using the SGD (Bottou 2010) optimiser with a learning rate schedule defined as MultiStepLR (optimiser, milestones=[45, 90, 120], gamma=0.2).

Spiking-transformer with SSDP on CIFAR-10, CIFAR-100, and ImageNet

For our models evaluated on CIFAR-10, CIFAR-100, and ImageNet, we trained for 600 epochs using the Adaptive Moment Estimation with Weight Decay (AdamW) optimiser with a base learning rate of 5×10^{-4}. Cosine annealing schedules were applied to both the learning rate and SSDP parameters, with SSDP training initiated after 10 epochs. On the CIFAR10-DVS dataset, we used the same configuration but trained for 100 epochs. Specific SSDP hyperparameter settings for these experiments are provided in the main text.

3.6 Neuron Model and Exponential SSDP Variant

3.6.1 SNNs neuron dynamics

In our single hidden layer SNNs model, the temporal behaviour of the spiking neurons is captured using a recurrent architecture that emulates the leaky integration (Lapicque 2007) observed in biological systems. At each discrete time step, the membrane potential $m(t)$ is updated according to an exponential decay mechanism defined by

$$\alpha = \exp\left(-\frac{\Delta t}{\tau_m}\right), \tag{3.8}$$

where τ_m denotes the membrane time constant and Δt is the time interval between updates. The previous membrane potential $m(t-1)$ is attenuated by α, while the current synaptic input, comprising both external stimuli and recurrent feedback, is integrated into the updated state.

To further approximate biological processes, the model incorporates a dendritic integration mechanism (Zheng et al. 2024). The dendritic input $d_{\text{input}}(t)$ is computed as

$$d_{\text{input}}(t) = \sigma(\tau_n) \cdot d_{\text{input}}(t-1) + [1 - \sigma(\tau_n)] \cdot I(t), \tag{3.9}$$

where τ_n is a learnable dendritic time constant, $\sigma(\cdot)$ is the sigmoid function, and $I(t)$ represents the combined synaptic current. The overall update of the membrane potential is then given by

$$m(t) = \alpha \cdot m(t-1) + (1-\alpha) \cdot d_{\text{input}}(t), \tag{3.10}$$

spike generation is triggered when the membrane potential exceeds a fixed threshold v_{th}. Spiking neurons are non-differentiable due to their binary firing behaviour, so we adopt a custom activation function: it behaves as a hard threshold in the forward pass and uses a multi-Gaussian surrogate gradient for backpropagation, enabling gradient-based optimisation.

We implement the neuron model in PyTorch (Paszke et al. 2019), combining leaky membrane dynamics, dendritic integration, and surrogate gradients. This design enables the simulation of temporally precise spiking behaviour and supports the study of plasticity in neuromorphic systems.

3.6.2 Exponential SSDP

Learnable scalars.

A_+ (potentiation scale), A_- (depression scale), τ_+ (causal time constant), τ_- (anti-causal time constant).

Inputs.

The module receives per-sample spike-presence indicators $Q_{b,i} \in \{0,1\}$ (pre/input), $P_{b,j} \in \{0,1\}$ (post/output), From these, we form a *synchrony gate* by a broadcasted outer product $\lambda_{b,ji} = P_{b,j}\, Q_{b,i} \in \{0,1\}$, which equals 1 iff both neurons spike at least once within sample b. The module also receives a nonnegative per-sample lag Δt_b (shared across all pairs for that sample).

Update rule.

With the above gate and a decaying exponential in the lag, the batch update implemented by the module is

$$\Delta W_{\text{pot}} = \frac{A_+}{B} \sum_{b=1}^{B} e^{-\Delta t_b/\tau_+}\, \lambda_b, \tag{3.11}$$

$$\Delta W_{\text{dep}} = \frac{A_-}{B} \sum_{b=1}^{B} e^{-\Delta t_b/(\tau_+\tau_-)}\, \lambda_b, \tag{3.12}$$

Here $\lambda_b = P_b\, Q_b^\top$ is the $(N_{\text{out}} \times N_{\text{in}})$ synchrony mask for sample b, and the exponential factors are scalars broadcast to that matrix. *As implemented in code*, the depression kernel uses $e^{-\Delta t_b/(\tau_+\tau_-)}$ (because Δt is first normalised by τ_+ and then divided by τ_- before exponentiation); thus both potentiation and depression are gated by the same λ but decay with different effective time scales.

This exponential variant was only used in tiny sanity tests; all main-text results adopt the symmetric Gaussian SSDP described in the Methods.

3.6.3 Ablation Studies

Different SSDP Variants in SNN-Transformer

We ablate the synchrony update to test whether SSDP is functionally necessary in Table 3.4. On CIFAR-100 with the SNN-Transformer, removing SSDP reduces top-1 accuracy relative to the same model trained with SSDP. And switching from a Gaussian to an Exponential kernel will also reduce the gain. With the symmetric Gaussian window we adopt, accuracy improves from 78.73% to 79.48% under a matched schedule.

Why Gaussian?

SSDP is intended to reward *near-coincident* group activity rather than precise spike ordering. A symmetric Gaussian kernel implements this directly: (i) it maximises the update at zero lag and decays smoothly on both sides, tolerating small timing jitter; (ii) it avoids introducing an arbitrary lead/lag preference, aligning with the rate-controlled evidence of a *symmetric* zero-lag peak Fig. 3.2(a–b); and (iii) it yields bounded, smooth contributions that stabilise late training when combined with clipping. These properties agree with our empirical findings: Fig. 3.3 shows that, with SSDP, the representation becomes more organised along a few shared directions, while spike timing becomes more reliable. In contrast, an asymmetric, order-sensitive decay can overemphasise fine timing differences and is less robust under event noise.

TABLE 3.4: Top-1 classification accuracy (%) on CIFAR-100 with different SSDP variants in SNN-Transformer.

Model Variant	SSDP Type	Accuracy (%)
Baseline (No SSDP)	—	78.73
+ SSDP	Exponential decay	79.12 (+0.39)
+ SSDP	Gaussian decay	**79.48 (+0.75)**

TABLE 3.5: Performance under different A_+ and A_- settings in the SSDP rule.

A_+	A_-	Accuracy (%)
1.5×10^{-4}	5.0×10^{-5}	79.48
1.0×10^{-4}	1.0×10^{-4}	79.03
5.0×10^{-5}	1.5×10^{-4}	79.07
1.0×10^{-2}	1.0×10^{-2}	Divergent

3.6.3.1 Optimal hyperparameter tuning balances synchrony-driven plasticity

We sweep the SSDP strength on the SNN-Transformer (CIFAR-100) over a log-spaced grid of the potentiation scale A_+ and the depression A_- under a matched schedule and fixed seeds; all other hyperparameters are held constant, and each run uses the same epoch budget. The validation top-1 results are summarised in Table 3.5. The accuracy landscape forms a ridge rather than a single sharp optimum. When A_+ is too small, the SSDP contribution is negligible and accuracy regresses toward the no-SSDP baseline. Performance is maximised around $(A_+, A_-) = (1.5{\times}10^{-4},\ 5{\times}10^{-5})$ used in our main models; pushing either parameter much higher quickly destabilises training – once A_+ approaches the 10^{-3} scale together with proportionally large A_-, runs fail to converge. Empirically, settings with $A_+ > A_-$ work best: near-synchronous pairs receive potentiation while asynchronous pairs remain weakly depressed. If A_- approaches A_+, the net effect becomes overly suppressive and learning stalls; if A_+ is made far larger, early or random coincidences can trigger runaway growth. This ridge location—A_+ stronger than A_-—is consistent with our rate-controlled finding that same-time co-activation is positively enriched (Fig. 3.2(a–b)). At settings on the ridge, we also observe the representation spreading mainly along a few shared axes (PCA) and spike-time jitter tightening without forcing rigidity (Fig. 3.3). Off the ridge, these effects weaken, or training becomes unstable. Practically, we recommend starting from $(A_+, A_-) = (1.5{\times}10^{-4},\ 5{\times}10^{-5})$, tuning on a log scale while keeping $0 < A_- < A_+$. If gains vanish, increase A_+ one notch at a similar ratio; if training oscillates or diverges, first reduce A_+ and then A_-.

A related biological observation is that correlated inputs tend to be stabilised, while uncorrelated inputs are weakened (e.g., Song et al. 2000; Katz and Shatz 1996). For example, temporally coordinated synaptic activity can be maintained, whereas asynchronous inputs may undergo activity-dependent weakening in specific preparations (Kehayas and Holtmaat 2015). This picture is consistent with our strength calibration: setting $A_+ > A_-$ lets near-synchronous events produce net potentiation while asynchronous pairs remain weakly depressed.

3.7 Integration into Single Hidden Layer SNNs

We first conducted preliminary evaluations of SSDP on a simple single-hidden-layer SNN without convolutional structures. Experimental results in Table 3.6 showed that our model achieved an accuracy of 89.37% on the static Fashion-MNIST dataset, comparable to certain deep SNN architectures and

TABLE 3.6: Classification accuracy (Top-1) based on SNNs. The proposed model consists of a single hidden layer and a readout classifier, without any convolutional components.

Dataset	Learning rule	Conv. structure	Accuracy (%)
Fashion-MNIST	Sym-STDP (Hao et al. 2020)	✗	85.3
	GLSNN (Zhao et al. 2020)	✗	89.1
	STB-STDP (Dong et al. 2023)	✓	87.0
	R-STDP (Quintana et al. 2022)	✓	83.26
	SSTDP (Liu et al. 2021)	✓	85.16
	S2-STDP (Goupy et al. 2024a)	✓	85.88
	Proposed (1 hidden layer)	✗	**89.37**$_{\pm 0.14}$
CIFAR-10	STB-STDP	✓	32.95
	R-STDP	✓	51.74
	S2-STDP	✓	61.80
	SSTDP	✓	60.80
	NCG-S2-STDP (Goupy et al. 2024a)	✓	66.41
	Proposed (1 hidden layer)	✗	**52.15**$_{\pm 0.23}$
N-MNIST	STDBP	✗	98.74
	Synaptic Kernel Inverse (Cohen et al. 2016)	✗	92.87
	BP-SNN (Perez-Nieves and Goodman 2021)	✗	92.7
	STDP+CDNA-SNN (Saranirad et al. 2024)	✗	98.43
	Proposed (1 hidden layer)	✗	**96.54**$_{\pm 0.11}$

TABLE 3.7: Classification accuracy (Top-1) based on Spiking-ResNet18 (Hu et al. 2021) compared with ANN baselines.

Dataset	Model	Accuracy (%)
CIFAR-10	SpikeResNet18-STDP	76.71
	ResNet20 (ANN) (Sengupta et al. 2019)	89.1
	STBP (Wu et al. 2018)	89.53
	EIHL (Jiang et al. 2023)	90.25
	ResNet20 (ANN) (Han et al. 2020)	91.47
	Proposed	**91.37**$_{\pm 0.37}$
CIFAR-100	SpikeResNet18-STDP	32.66
	STBP	58.48
	EIHL	58.63
	ResNet20 (ANN) (Han et al. 2020)	68.72
	Proposed	**60.97**$_{\pm 0.42}$

TABLE 3.8: Extended evaluation metrics on CIFAR-100 using SNN-Transformer, with and without SSDP.

Metric	Without SSDP	With SSDP
Top-1 Accuracy	78.73%	**79.48%**
Top-5 Accuracy	93.64%	93.58%
Throughput (it/s)	352.75	352.31
MACs	1.226 G	1.226 G
Parameters	10.83 M	10.83 M
Avg SOPs	0.548 G	0.560 G
Power (mJ/sample)	0.4936	0.5043
A/S Power Ratio	11.43	11.19

convolutional STDP-based models. On the more challenging CIFAR-10 dataset, our single-layer SSDP model achieved 52.15% accuracy, slightly outperforming the more complex convolutional R-STDP model but still lagging behind other advanced STDP-based models. Subsequently, we evaluated our model on the event-based N-MNIST dataset; however, its performance did not surpass that of existing STDP-based methods. We provide an in-depth analysis of SSDP's performance on DVS datasets in the following subsection.

3.8 Extended Evaluation of SSDP

To supplement the main results presented in the manuscript, we provide a detailed performance comparison of the SNN-Transformer architecture on the CIFAR-100 dataset, evaluated with and

without the proposed SSDP mechanism. In addition to accuracy metrics, we report throughput, Multiply-accumulate Operations (MACs) (Han et al. 2015; Fang et al. 2021)), Synaptic Operations (SOPs), and average power consumption. These metrics, alongside average power consumption, offer a more comprehensive view of efficiency across architectures. Profiling was conducted using the integrated tools in SpikingJelly (Fang et al. 2023). Table 3.8 summarises the key results.

The integration of SSDP led to a modest but meaningful increase in top-1 accuracy, from 78.73% to 79.48%, corresponding to an approximate 3.5% reduction in classification error. Top-5 accuracy remained comparable, indicating that SSDP enhances first-choice decision precision without degrading overall output diversity. Importantly, this improvement was achieved with negligible impact on inference efficiency: throughput, MACs, and parameter count remained unchanged, while only minimal increases were observed in synaptic operations (SOPs; +2.2%) and per-sample energy consumption (+2.1%). Although the accuracy-to-spike power ratio (A/S Power Ratio) declined slightly from 11.43 to 11.19, the overall energy-performance trade-off remained highly favourable. These results highlight that SSDP offers consistent performance gains with minimal cost, reinforcing its practicality for neuromorphic applications where efficiency is critical.

3.9 Additional qualitative rasters

Fig. 3.5 shows example spike train visualisations from a subset of excitatory neurons in a one-layer SNN on the Fashion-MNIST task. In the non-SSDP-trained network, spikes are distributed irregularly in time across neurons, with relatively few coincident events. In contrast, the SSDP-trained network displays clear episodes of synchronised firing: many neurons fire in unison, forming pronounced vertical bands in the raster plot. These synchronous events occur reliably in response to specific input patterns (e.g., particular digits), suggesting that SSDP causes ensembles of neurons to jointly encode those patterns. This supports the interpretation that SSDP's learning rule reorganises synaptic weights so that neurons that detect the same feature tend to fire together, effectively creating cell assemblies reminiscent of those hypothesised in vivo (Hebb 2005a; Buzsáki 2010). Notably, the phenomenon was observed across network types and layers – for instance, early convolutional layers of the spiking ResNet with SSDP also showed heightened coincident spiking compared to STDP, indicating that synchrony reinforcement is a general effect and not limited to shallow networks.

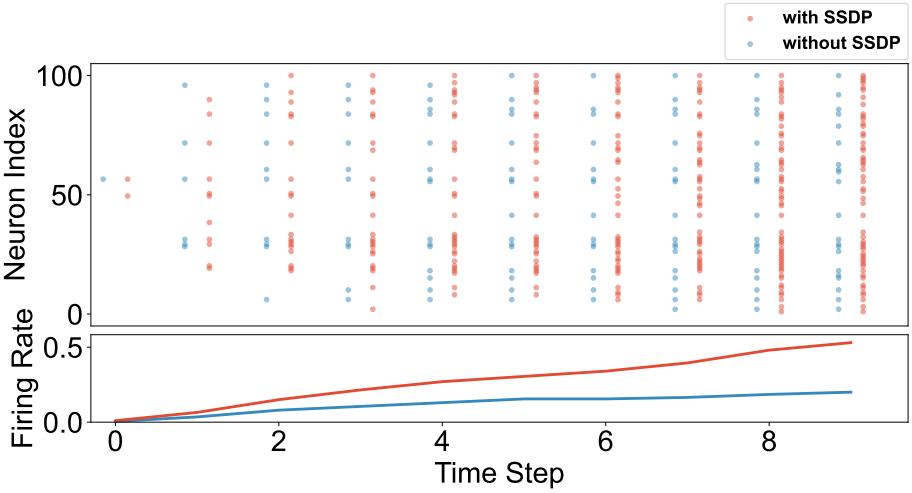


FIGURE 3.5: Synchrony events and average firing rate on Fashion-MNIST. Top: spike rasters for a representative subset of neurons, comparing SSDP (red) and no-SSDP (blue). Each dot marks a binary spike at a timestep. SSDP exhibits more frequent vertically clustered coincident spikes, indicating stronger population synchrony. Bottom: average firing rate over time (mean spikes per neuron per timestep across the shown population and samples). In this setting, SSDP yields a higher rate, consistent with its rule that potentiates near-synchronous pairs and applies depression to asynchronous pairs, thereby increasing the probability of coincident events.

3.10 Discussion

In this chapter, we introduce a synchrony-sensitive, training-time local rule termed SSDP for spiking networks and integrate it into both shallow models and transformer-style SNNs; to the best of our knowledge, this is the first deployment of a biologically inspired plasticity rule within an SNN-Transformer under matched modern training budgets. Across benchmarks, SSDP improves top-1 accuracy, exhibits a mid-training transition to a lower-variance loss regime, yields broader yet organised feature geometry, and tightens spike-time reliability by shrinking high-jitter tails without enforcing rigidity. Mechanistically, SSDP operates on population co-activation within a short coincidence window: near-synchronous pre-post activity receives potentiation and non-coactive pairs receive weak depression. The update is formed by broadcasted outer products, and adds linear-time memory/compute only during training. Rate-controlled analyses show a symmetric zero-lag excess that is not explained by marginal rates, and representation probes indicate a reallocation of variance toward a few shared directions rather than a diffuse spread; together with tighter jitter distributions, these observations provide a data-consistent account of smoother optimisation and reduced overfitting. In practice, SSDP

functions as a lightweight event-structure regulariser that biases learning toward repeatable, task-relevant spike motifs while preserving convergence behaviour. Overall, injecting an event-structure signal at training time yields more organised spatial codes and more reliable spike timing under realistic budgets, and remains compatible with neuromorphic deployment due to locality and zero-cost inference. SSDP thus offers a practical mechanism for improving SNN learning while providing a testable computational hypothesis for population-level plasticity.

3.11 Data and Code Availability

All datasets used in this study are publicly available benchmark datasets. No proprietary or custom datasets were used or created in this work. The code is available at: https://github.com/NeuroSyd/SSDP.

CHAPTER 4

Synchrony-Gated Plasticity with Dopamine Modulation for Spiking Neural Networks

---

The content presented in this chapter is accepted by *Transactions on Machine Learning Research*, titled *Synchrony-Gated Plasticity with Dopamine Modulation for Spiking Neural Networks*

(1) Y. Tian, S. Tensingh, J. K. Eshraghian, N. D. Truong, and O. Kavehei *Loss-Aware Synchrony Plasticity for Deep Spiking Networks*
Status: Accepted by Transactions on Machine Learning Research (TMLR).

4.1 Abstract

While surrogate backpropagation is useful for training deep Spiking Neural Networks (SNNs), incorporating biologically inspired local signals at a large scale remains challenging. This difficulty stems primarily from the high memory demands of maintaining accurate spike-timing logs and the potential for purely local plasticity adjustments to clash with the supervised learning goal. To effectively leverage local signals derived from spiking neuron dynamics, we introduce Dopamine-Modulated Spike-Synchrony-Dependent Plasticity (DA-SSDP), a synchrony-based rule that is sensitive to loss and serves as a regulariser during training. This helps to enhance backpropagation in spiking transformers. DA-SSDP condenses spike patterns into a synchrony metric at the batch level. An initial brief warm-up phase assesses its relationship to task loss and sets a fixed gate that subsequently adjusts the magnitude of the local update. In cases where synchrony proves unrelated to the task, the gate settles at one, simplifying DA-SSDP to a basic two-factor synchrony mechanism that delivers minor weight adjustments driven by concurrent spike firing and a Gaussian latency function. These small weight updates are only added to the network's deeper layers after the backpropagation phase, and our tests showed that this simplified version didn't degrade performance and sometimes gave a small accuracy

boost. The rule stores only binary spike indicators and first-spike latencies with a Gaussian kernel. Without altering the model structure or optimisation routine, evaluations on benchmarks like Canadian Institute for Advanced Research (CIFAR)-10 (+0.42%), CIFAR-100 (+0.99%), CIFAR10-Dynamic Vision Sensor (DVS) (+0.1%), and ImageNet-1K (+0.73%) demonstrated reliable accuracy gains, accompanied by a minor increase in computational overhead.

4.2 Introduction

SNNs are depicted as the third generation of artificial neural networks (Maass 1997). Driven by the growing demand for energy-efficient computing and the rising interest in brain-inspired architectures, SNNs have gained traction in recent years with neuromorphic hardware and biologically grounded research communities because of their event-driven nature and potential for significantly lowering energy consumption (Amir et al. 2017; Davies et al. 2018b; Friedmann et al. 2016; Merolla et al. 2014). By transmitting information through discrete spikes, SNNs enable sparse, asynchronous processing that closely aligns with the characteristics of brain-inspired hardware (Furber et al. 2014; Merolla et al. 2014). That said, their performance on complex and large-scale benchmarks still lags behind conventional Artificial Neural Networks (ANNs) (Hu et al. 2021; Shi et al. 2024; Zhou et al. 2022). This gap has led researchers to explore whether architectural techniques from ANNs, such as Transformer attention mechanisms, can be adapted to improve performance in spiking models.

Efforts to combine spiking dynamics with Transformer-style attention mechanisms have recently gained momentum (Yao et al. 2024; Zhou et al. 2022). Several research groups have begun reworking components of the transformer architecture, particularly self-attention, to better align with the characteristics of spike-based computation. One early example is Spikformer (Zhou et al. 2022), which replaces operations such as softmax with spike-compatible alternatives, achieving competitive accuracy on image classification tasks. Building on this foundation, (Yao et al. 2024) proposed a fully spike-driven Transformer, while SpikingResformer (Shi et al. 2024) introduced residual convolutional blocks into a spiking attention framework, achieving 79.40% Top-1 accuracy on ImageNet using only four time steps, narrowing the gap between SNNs and standard ANNs.

Despite rapid architectural progress, current spiking transformers are trained almost exclusively with surrogate backpropagation. What is missing is a scalable, structure-aware local signal that can be injected during training without altering the architecture. Conventional regularisers (e.g., weight

decay (Krogh and Hertz 1991), label smoothing (Szegedy et al. 2016), dropout (Srivastava et al. 2014) operate on weights or real-valued activations and largely ignore the event structure that determines computation in SNNs. In contrast, timing-based plasticity rules accurately capture certain neural behaviours. However, when simply adapted to deep, multi-layer networks, they demand detailed temporal tracking and extra memory for states. Moreover, their adjustments lack alignment with the supervised learning signal and this hinders safe, scalable implementation. (Tian et al. 2025). We therefore treat plasticity not as a replacement for backprop but as a training-time regulariser that is memory-efficient, aligned with task loss, and safe to disable when uninformative. Specifically, we employ spike synchrony as a compact proxy for coordinated population activity. We modulate its local impact via a global performance signal. A brief warm-up phase assesses the synchrony-loss correlation, after which the modulation is locked in and simply scales a Hebbian-type update. When synchrony is task-irrelevant, the gate converges to one, and the rule reduces to a benign two-factor baseline applied post-update in late layers. This framing naturally connects to modulated three-factor rules in neuroscience, which are local activity gated by dopamine-like signals, and motivates integrating biologically inspired, self-organising plasticity into spiking transformers without altering their forward dynamics.

In contrast to most artificial systems, biological brains combine both supervised and unsupervised forms of learning (Storrs et al. 2021; Hennig et al. 2021). Synaptic changes are typically driven by local spike activity patterns (Feldman 2012), and these changes can also be shaped by neuromodulatory signals. Dopamine, for instance, plays a central role in linking synaptic modification to reward feedback (Frémaux and Gerstner 2016; Speranza et al. 2021). Inspired by this mechanism, we propose integrating biologically motivated, self-organising plasticity into spiking transformers to enhance their learning capabilities and generalisation. Previous research indicates that Hebbian plasticity (Hebb 2005a) can effectively support continual learning and unsupervised feature extraction in SNNs (Xiao et al. 2024). Yet, current spiking transformer implementations do not incorporate such biologically inspired learning mechanisms.

In this work, we present a novel training strategy for spiking transformers that integrates supervised gradient-based training with an unsupervised, reward-modulated plasticity mechanism. We name this method **D**op**a**mine-modulated **S**pike-**S**ynchrony-**D**ependent **P**lasticity (**DA-SSDP**). The fundamental idea is to monitor spike-timing synchrony during training and dynamically adjust synaptic weights in response to task-specific loss. Essentially, DA-SSDP encourages synapses that contribute to

synchronised spikes correlated with correct predictions, and penalises those associated with poorly timed spikes, mimicking the modulatory role of dopamine observed in shaping neural coordination and learning. Through this additional plasticity mechanism, we aim to (i) capture subtle spike synchronous correlations potentially overlooked by gradient descent, (ii) regularise neural activity to reduce uncoordinated spiking, and (iii) explore how synchrony-based local plasticity learning mechanisms can be integrated with global learning signals to improve generalisation without imposing additional computational cost.

This chapter presents the following primary contributions:

(A) **Loss-modulated synchrony as a training-time regulariser.**
We present DA-SSDP as a regulariser applied during training. It keeps the forward computation graph unchanged. A brief warm-up phase derives a dopamine-inspired scaling factor from synchrony and loss. After that, the gate stays fixed and simply adjusts the size of a Hebbian-style local update for each batch. When synchrony is task-irrelevant, the gate collapses to unity, and the rule safely reverts to the two-factor baseline (no online loss), providing a scalable, structure-aware signal without architectural changes.

(B) **Memory-efficient, scalable implementation for deep SNN-Transformers.**
DA-SSDP requires only binary spike indicators and first-spike latencies, adding only $O(C_{\text{out}}C_{\text{in}})$, where C_{in} refers to the pre-synaptic neurons and C_{out} refers to the post-synaptic neurons in the input and output channels of the module to which DA-SSDP is attached, per-batch element-wise operations, and no inference-time cost. We provide a drop-in implementation that tweaks just the final classification part and a simple late-stage feature projector, keeping the architecture and training schedule (epochs, batch size, optimiser) unchanged. This approach uses much less memory than storing detailed spike-timing records.

(C) **Quantitative validation and robustness analysis.**
Across CIFAR-10/100, ImageNet-1K, and an event stream (CIFAR10-DVS), DA-SSDP shows consistent accuracy gains under the same training setup. On the test set, the median batch synchrony S_b increases from 3×10^{-4} to 1.0×10^{-2} ($\sim 30\times$).

4.3 Related works

Spiking transformers and training-time cost: Recent Spiking Neural Network (SNN)-Transformers (Zhou et al. 2022; Zhou et al. 2023; Shi et al. 2024; Yao et al. 2024; Zhou et al. 2024a; Yao et al. 2023) primarily target inference-time efficiency by exploiting sparse spikes and a few time steps while maintaining competitive accuracy on image benchmarks. However, training remains almost entirely surrogate Backpropagation (BP) (Zhou et al. 2024a; Gygax and Zenke 2025; Hu et al. 2024), whose unrolling, state storage, and memory traffic can be comparable to, or even higher than ANNs counterparts (Zhou et al. 2024b). Architectural progress, therefore, narrows the inference gap but, by itself, does not provide a scalable, structure-aware training signal that interacts with spike statistics.

Local plasticity at scale: Biologically motivated rules range from two-factor timing/trace updates (Spike-Timing-Dependent Plasticity (STDP)-like) to three-factor modulated variants (Mazurek et al. 2025). In deep, multi-layer neural networks, timing-based plasticity rules run into two ongoing problems. First, scalability: keeping super-detailed records of spike timings and assigning credit/blame between pairs of neurons over time, which uses up a huge amount of memory and processing bandwidth. Secondly, the update directions are not coordinated with the supervised loss and may conflict with gradient signals. Consequently, most demonstrations remain on small-scale tasks or in online/RL settings (e.g., Mazurek et al. 2025; Zhou et al. 2024b). Along two useful axes, local signal (pairwise timing vs population synchrony) and global modulation (absent vs reward/prediction-error), prior work largely emphasises pairwise timing, sometimes gated by a neuromodulator, while underusing population-level coordination. Yet neuroscience highlights population synchrony as a description of coordinated activity (Vinck et al. 2023; Majhi et al. 2024), whereas synchrony has rarely been used as a scalable, supervised training-time signal in deep SNNs.

We fill this gap by using batch-level spike synchrony as a low-memory substitute. It relies on binary spike indicators and first-spike latencies with $O(C_{\text{out}}C_{\text{in}})$ per batch. We then adjust its local influence using a dopamine-inspired gate, derived from synchrony and loss during a brief warm-up period. Once set, this modulation remains constant and merely scales a Hebbian-style local update. As a result, the rule enhances backpropagation without altering the forward-pass dynamics. When synchrony is uninformative, the gate converges to one, simplifying the update to a harmless two-factor baseline that's applied after the main updates in deeper layers. This provides the scalable training signal that prior spiking transformers have been missing. The formulation and implementation are detailed next.

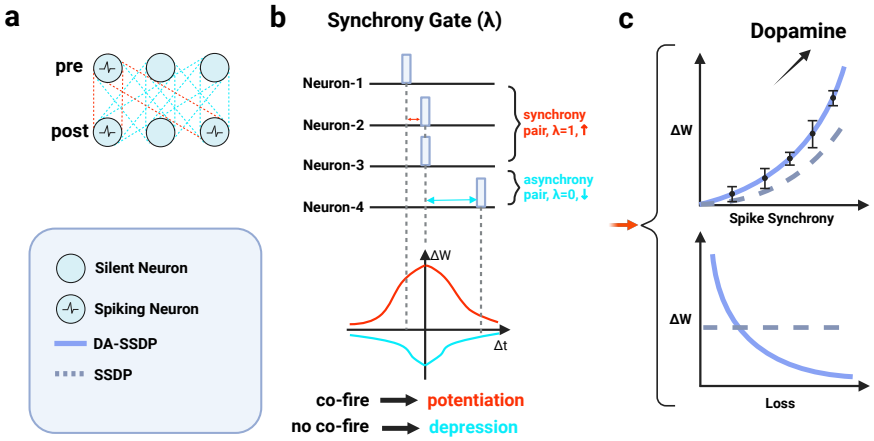


FIGURE 4.1: **Overview of DA-SSDP (a)** Pre–post layer and binary activity indicators from the current mini-batch. **(b) Synchrony gate λ -** A pre/post pair that co-fires in the sample sets $\lambda = 1$ and is potentiated (red), no co-firing sets $\lambda = 0$ and yields depression (cyan). The first-spike lag $|\Delta t|$ is passed through a Gaussian window $g = \exp\big(-\Delta t^2/(2\sigma^2)\big)$, which scales the magnitude only and the rule is order-invariant (synchrony decides the sign, timing sets the strength). **(c) Dopamine modulation -** During a short warm-up, a single scalar slope is fit from the empirical synchrony–loss correlation and after warm-up, the slope is frozen. Thereafter, the gate depends only on batch synchrony and simply rescales the local SSDP update (solid blue: DA-SSDP; dashed: ungated SSDP).

4.4 Method

This section introduces the proposed DA-SSDP mechanism and its role in enhancing the training of spiking transformer models. We begin by outlining the transformer architecture used in our experiments, followed by a detailed explanation of the DA-SSDP rule. We describe a two-stage training strategy that enables loss-aware integration of this local learning rule. Fig. 4.1 summarises DA-SSDP. For each pre/post pair within a sample, a binary synchrony gate $\lambda \in \{0, 1\}$ indicates whether both neurons fired at least once, co-firing ($\lambda = 1$) leads to potentiation and no co-firing ($\lambda = 0$) leads to depression (Fig. 4.1(a,b)). The absolute difference in first-spike times, $|\Delta t|$, feeds into a Gaussian function $g = \exp(-\Delta t^2/(2\sigma^2))$ to adjust only the magnitude of the update. The sign arises from the synchrony itself, ensuring the rule doesn't depend on spike order. Summing up λg across all connections produces a synchrony measure for the entire batch. During a short warm-up, we fit a single negative-slope model that captures the inverse relationship between synchrony and loss, then freeze

it. From then on, this factor serves as a scalar that rescales the local Spike-Synchrony-Dependent Plasticity (SSDP) update without changing its direction (Fig. 4.1(c)).

4.4.1 Spiking Neuron

As outlined in (Shi et al. 2024), Leaky Integrate-and-Fire (LIF)'s dynamics in discrete time are described as

$$v_i[t] = u_i[t] + \frac{1}{\tau}\big(I_i[t] - (u_i[t] - u_{\text{rest}})\big)\ , \tag{4.1}$$

$$s_i[t] = H\big(v_i[t] - u_{\text{th}}\big)\ , \tag{4.2}$$

$$u_i[t{+}1] = s_i[t]\, u_{\text{rest}} + \big(1 - s_i[t]\big)\, v_i[t]\ , \tag{4.3}$$

where $u_i[t]$ and $v_i[t]$ are the membrane potentials of neuron i before and after the charging step, τ is the membrane time constant, $H(\cdot)$ is the Heaviside function, u_{th} is the firing threshold and u_{rest} is the resting potential. The input drive is

$$I_i[t] \ = \ \sum_j w_{i,j}\, s_j[t]\ , \tag{4.4}$$

with $s_j[t] \in \{0, 1\}$ the pre-synaptic spike at time t and $w_{i,j}$ the synaptic weight from neuron j to neuron i. Eq. 4.1 models the charging process with leak toward u_{rest}, Eq. 4.2 emits a spike when the threshold is crossed and Eq. 4.3 resets to u_{rest} on a spike and otherwise keeps $v_i[t]$.

4.4.2 Implementation Details

Integration of DA-SSDP into SpikingResformer: We integrated DA-SSDP into the SpikingResformer architecture, as illustrated in Fig. 4.2. The model follows a hierarchical backbone comprising (i) a front-end down-sampling stem (Prologue) that maps the input image to an initial spike feature map, (ii) Dual Spike Self-Attention (DSSA), (iii) Grouped Convolutional Feed-forward Blocks (GWFFN), and (iv) a spike-based classifier at the output stage. We insert two DA-SSDP modules as light-weight post-update hooks without changing the forward path (i) the linear classifier and (ii) the 1×1 projection in the last DSSA block. At this hook, we consider the DSSA input channels as pre-synaptic and the projection outputs as postsynaptic. In every training iteration, we begin with the usual surrogate-gradient update. Next, using the current mini-batch, we generate binary activity markers and per-channel first-spike timings at the two hook locations, invoke DA-SSDP to calculate an additive weight adjustment

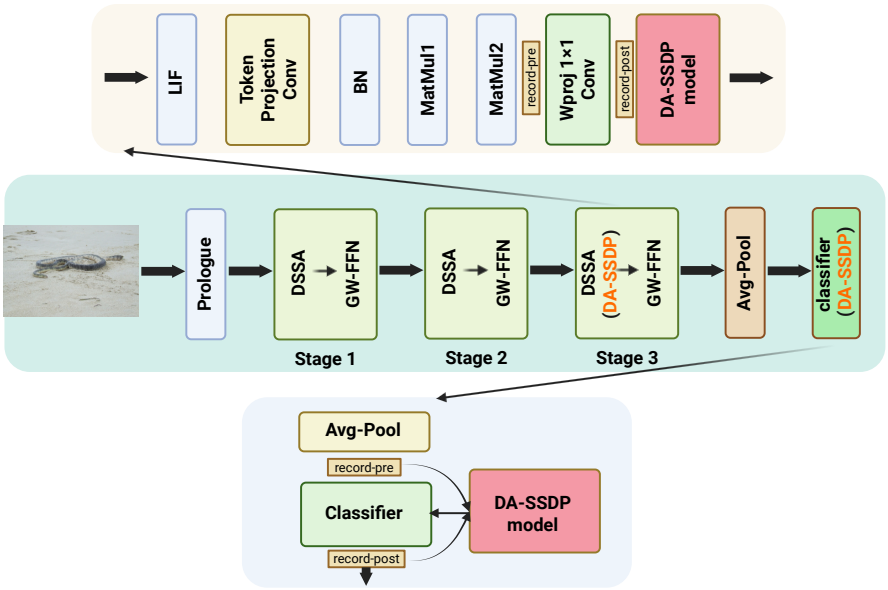


FIGURE 4.2: **DA-SSDP integration points in SpikingResformer** The DA-SSDP module is inserted at two locations (1) the 1×1 projection convolution of the last DSSA block in Stage 3 (2) the linear classifier following global average pooling. During training, each hook records pre/post-spike activity and applies DA-SSDP update after the warm-up phase, operating alongside standard back-propagation.

ΔW, and apply it directly to the relevant weights. During the warm-up phase, $e < E_{\text{warm}}$ where e denotes the current epoch and E_{warm} denotes the maximum epoch of the warm-up phase for computing the dopamine gate, DA-SSDP only records statistics without making any weight modifications. For the classifier hook, we extract pre- or post-activity from the per-step readout activations, and for the DSSA hook, a channel counts as active in a time step if any of its spatial positions are active, with the first-event timing set to the earliest step at which the indicator activates. The two DA-SSDP instances maintain separate parameters and warm-up data. This setup preserves attention maps and all forward-pass operations intact, with DA-SSDP simply providing a minor post-update adjustment driven by batch-level synchrony.

4.4.3 Dopamine-Modulated SSDP Mechanism

We use a mini-batch of size B, where samples are indexed by $b \in \{1, \ldots, B\}$. For a hooked module with C_{in} input channels and C_{out} output channels, we index pre-synaptic channels as $j \in \{1, \ldots, C_{\text{in}}\}$

and postsynaptic channels as $i \in \{1, \ldots, C_{\text{out}}\}$. Let P be the binary pre-synaptic spike matrix of shape $P \in \{0,1\}^{B \times C_{\text{in}}}$ and Q the binary postsynaptic spike matrix of shape $Q \in \{0,1\}^{B \times C_{\text{out}}}$ for a mini-batch of size B (each row resembles a sample, and a column represents a channel). In the implementation, these indicators are obtained by thresholding layer activations during the forward pass and keeping only whether each channel fired at least once in the window, yielding a stable binary co-activation signal that is cheap to cache and process at scale. For every (j, i) we also record the first-spike latency $\Delta t_{b,i,j} = \left| t^{\text{post}}_{b,i} - t^{\text{pre}}_{b,j} \right|$. Only the first spike time is stored, minimising memory use while retaining the core temporal signal. Channels that remain silent throughout the window receive a $t = T$, where T is the window length in time steps. The first spike has been shown to carry most of the critical information, whereas subsequent spikes contribute little additional content and may even introduce redundancy Thorpe et al. 2001; Rullen and Thorpe 2001; Gollisch and Meister 2008. The forward pass provides these timestamps directly as tensors t^{pre} and t^{post}, enabling on-the-spot computation of Δt without rerunning the sequence.

We now outline the fundamental update mechanism of DA-SSDP. For a mini-batch, the weight adjustment between pre-synaptic j and post-synaptic i is given by

$$\Delta W_{i,j} = \text{clip}\left(\frac{1}{B} \sum_{b=1}^{B} G_b \, g(\Delta t_{b,i,j}) \Big(A_+ \lambda_{b,i,j} - A_- (1 - \lambda_{b,i,j}) \Big), -1, \, 1 \right), \tag{4.5}$$

where $\lambda_{b,i,j} = Q_{b,i} P_{b,j} \in \{0,1\}$ signals co-spiking between the post- and pre-synaptic channels, $g(\Delta t) = \exp\big[- \Delta t^2 / (2\sigma^2) \big]$ is a Gaussian kernel with learnable bandwidth $\sigma > 0$, and $A_+, A_- > 0$ control the degrees of potentiation and depression, respectively. The batch-specific scalar gate G_b is detailed further in the warm-up calibration section (Eq. 4.14).

This equation highlights three key elements of DA-SSDP. First, the binary synchrony gate λ ensures that only coincident pre- and post-spike events contribute to plasticity, grounding the rule firmly in spike synchrony. Second, the temporal kernel $g(\Delta t)$ prioritises updates for near synchronous events while suppressing those with larger delays, reflecting how spike timing influences synaptic strength. Third, the Loss-aware modulation G_b injects a loss-dependent signal, enabling the local plasticity rule to remain compatible with global task objectives. Together, these elements produce weight increments that are clipped to a bounded range, preventing instability and ensuring stable training in deep architectures.

4.4.3.1 Instantaneous Potentiation and Depression

Gaussian weighting: For each pre-post pair, we apply a continuous Gaussian kernel

$$g_{b,i,j} \;=\; \exp\!\Big[-\tfrac{\Delta t_{b,i,j}^2}{2\sigma^2}\Big]\,, \tag{4.6}$$

where σ is a learnable bandwidth parameter that's shared across the module. Since g decreases with $|\Delta t|$, updates are strongest for nearly synchronous spikes and taper off as the pre–post latency widens. Using a continuous kernel avoids fragile outcomes tied to rigid thresholds and maintains effective gradients during combined backpropagation training.

Synchrony gate: Co-activation is represented by a binary mask

$$\lambda_{b,i,j} = Q_{b,i} P_{b,j} \in \{0,1\}\,. \tag{4.7}$$

Here λ is computed via an outer product $QP^\top$ for each sample (i.e., for sample b, $Q_b \in \mathbb{R}^{C_{\text{out}}\times 1}$ and $P_b \in \mathbb{R}^{C_{\text{in}}\times 1}$; we transpose P_b to $1 \times C_{\text{in}}$ to form the outer product $Q_b P_b^\top$, yielding a $C_{\text{out}} \times C_{\text{in}}$ mask) at $O(C_{\text{out}}C_{\text{in}})$ complexity and no need for temporal scanning, because first-spike times are computed once per channel (not per pair) and Δt is formed by broadcasting t^{post} and t^{pre}, avoiding any $O(TC_{\text{out}}C_{\text{in}})$ per-pair search. Consequently, the combined factor $g_{b,i,j}\lambda_{b,i,j}$ functions as a gentle synchrony sensor, and it remains strictly positive only when the pair spikes together within the window, and its value adjusts according to temporal closeness through g.

Weight update: The per-sample plasticity change for a given sample b and connection pair (i,j) is defined as

$$\Delta w_{b,i,j} \;=\; g_{b,i,j}\Big((A_+ + A_-)\,\lambda_{b,i,j} - A_-\Big)\,, \tag{4.8}$$

leading to potentiation when λ=1 and depression when λ=0. In cases where both units spike, $\lambda = 1 \Rightarrow \Delta w = +A_+\, g$. Spikes that are closely timed ($\Delta t \approx 0$) produce the strongest potentiation, with bigger delays diminishing it. When only one unit spikes, $\lambda = 0 \Rightarrow \Delta w = -A_-\, g$. For both silent: $\lambda = 0$ and t^{pre}=t^{post}=$T \Rightarrow \Delta t = 0$, so $g = 1$ and $\Delta w = -A_-$. Coactive synapses are reinforced based on their temporal alignment, while inactive pairs are mildly suppressed, limiting accidental co-firing and improving generalisation without maintaining eligibility traces for each synapse.

4.4.3.2 Batch-level Synchrony Score

We combine spike occurrences into a single scalar synchrony value

$$S_b = \frac{1}{C_{\text{out}} C_{\text{in}}} \sum_{i,j} Q_{b,i} P_{b,j} = \left\langle Q_b P_b^\top \right\rangle , \tag{4.9}$$

which ranges in $[0, 1]$ and measures how synchronous the two populations are on sample b. In the implementation, this is calculated as the average of the outer product over the (i, j) dimensions. The division by $C_{\text{out}} C_{\text{in}}$ ensures that S_b can be meaningfully compared across layers or architectures with varying channel sizes. The definition is permutation-invariant with respect to channel ordering and depends only on binary co-firing, which makes it robust to activation scale and batch normalisation statistics. Relying on co-firing rather than full Δt details, the signal maintains low variance over time windows, while Δt still influences the individual updates through the Gaussian kernel.

4.4.3.3 Warm-up and Gate Calibration

During the E_{warm} epochs, DA-SSDP does not modify weights. Instead, it collects per-batch pairs (S_b, ℓ_b), where S_b is the batch-wise synchrony metric and ℓ_b is the supervised loss for that batch. These pairs are aggregated across all warm-up batches to form two aligned sequences.

Standardisation: Let $\{S_b\}_{b=1}^N$ and $\{\ell_b\}_{b=1}^N$ represent the collected sequences from the warm-up phase (length N). We calculate their means and standard deviations as follows,

$$\mu_S = \frac{1}{N} \sum_{b=1}^{N} S_b , \qquad \sigma_S = \sqrt{\frac{1}{N} \sum_{b=1}^{N} (S_b - \mu_S)^2} , \tag{4.10}$$

$$\mu_\ell = \frac{1}{N} \sum_{b=1}^{N} \ell_b , \qquad \sigma_\ell = \sqrt{\frac{1}{N} \sum_{b=1}^{N} (\ell_b - \mu_\ell)^2} , \tag{4.11}$$

and derive the normalised values

$$\hat{S}_b = \frac{S_b - \mu_S}{\sigma_S} , \qquad \hat{\ell}_b = \frac{\ell_b - \mu_\ell}{\sigma_\ell} . \tag{4.12}$$

Slope fitting: The dopamine slope is set by the negative empirical correlation between standardised synchrony and loss:

$$k \;=\; -\,\mathbb{E}_b[\hat{S}_b\, \hat{\ell}_b] \;= -\,\frac{1}{N} \sum_{b=1}^{N} \hat{S}_b\, \hat{\ell}_b \;, \tag{4.13}$$

where $\mathbb{E}_b[\cdot]$ signifies the empirical mean across the warm-up collection. In essence, a pattern of higher synchrony pairing with reduced loss produces $k > 0$, boosting weights for future high-S_b batches. A feeble connection yields $k \approx 0$, rendering the gate essentially impartial.

Gate at training time: After warm-up, (μ_S, σ_S, k) remain fixed. For each new batch, we compute the scalar gate.

$$G_b \;=\; \mathrm{clip}\Big(1 \;+\; k\,\frac{S_b - \mu_S}{\sigma_S},\; 0,\; 2\Big)\,, \tag{4.14}$$

which is uniformly applied to all synaptic pairs (i, j) within the batch, scaling the per-sample term in Eq. 4.5. The $\mathrm{clip}(x, a, b) := \min\{\max\{x, a\}, b\}$ keeps the modulation bounded and prevents explosive adjustments.

Robustness: If the warm-up data offers limited insight, for example, if the batch synchrony S_b has a very small dynamic range (leading to small σ_S) and/or negligible correlation with loss (so $k \approx 0$), we fall back to a neural gate via k=0 with standardised S_b. This securely deactivates the dopamine component without disrupting the overall training flow. For a practical demonstration, consider the CIFAR10-DVS example in Sec. 4.5.2, where S_b exhibits minimal variation during warm-up, yielding a fitted slope $k \approx 0$ with $G_b \approx 1$.

4.4.3.4 Complexity and Stability

All DA-SSDP computations reduce to broadcast outer products, element-wise exponential for the Gaussian kernel, scalar standardisation, and simple reductions over Boolean tensors. Overall, the effort grows as $\mathcal{O}(B\,C_{\text{out}}C_{\text{in}})$ for each mini-batch, primarily involving efficient vectorised tensor operations that align seamlessly with Graphics Processing Unit (GPU) acceleration. No synapse-specific eligibility traces persist between steps, with only first-spike timings and binary indicators stored from the forward pass; thus, the maximum memory burden consists of the batch-level accumulator ΔW alongside minor per-batch values $(S_b, \mu_S, \sigma_S, k)$. After calibration, the gate parameters (k, μ_S, σ_S) are fixed, whereas A_+, A_-, and σ continue to be trainable. This separation ensures numerical reliability in the dopamine-modulated pathway and prevents fluctuations towards the end of training. Backpropagation remains the core driver of convergence, while DA-SSDP provides additional, task-relevant information by coupling local population synchrony S_b to the global objective via the dopamine gate G_b. This coupling reinforces synchrony only when it predicts lower loss, guiding the network toward more stable firing regimes. Otherwise, the gate neutralises, and the update reduces to the two-factor SSDP baseline.

TABLE 4.1: Top-1 Classification accuracy, **(+)** indicates improvement over the baseline model. Energy is the estimation of energy consumption, same as (Yao et al. 2024; Zhou et al. 2022; Shi et al. 2024) in Appendix (4.6).

Dataset	Method	Architecture	Param (M)	Energy (mJ)	TimeStep	Accuracy (%)
CIFAR10	Spikformer (Zhou et al. 2022)	Spikformer-4-384	9.32	-	4	95.51
	Spikingformer (Zhou et al. 2023)	Spikingformer	-	-	4	95.61
	Spike-driven Transformer (Yao et al. 2024)	Spike-driven Transformer	-	-	4	95.6
	SpikingResformer (Shi et al. 2024)	SpikingResformer-CIFAR	10.83	-	4	95.95
	SpikingResformer+SSDP(w/o DA)	SpikingResformer-CIFAR	**10.83**	-	4	**96.24(+0.29)**
	SpikingResformer+DA-SSDP	SpikingResformer-CIFAR	**10.83**	-	4	**96.37(+0.42)**
CIFAR10-DVS	Spikformer	Spikformer-4-384	9.32	-	16	80.6
	Spikingformer	Spikingformer	-	-	16	81.3
	Spike-driven Transformer	Spike-driven Transformer	-	-	16	80.0
	Transformer	Transformer-4-384	9.32	-	1	81.02
	SpikingResformer	SpikingResformer-CIFAR	17.31	2.403	10	84.4
	SpikingResformer+SSDP(w/o DA)	SpikingResformer-CIFAR	**17.31**	**2.421**	10	**84.5(+0.1)**
	SpikingResformer+DA-SSDP	SpikingResformer-CIFAR	**17.31**	**2.456**	10	**84.5(+0.1)**
CIFAR100	Spikformer	Spikformer-4-384	9.32	-	4	77.86
	Spikingformer	Spikingformer	-	-	4	79.09
	Spike-driven Transformer	Spike-driven Transformer	-	-	4	78.4
	Transformer	Transformer-4-384	9.32	-	1	81.02
	SpikingResformer	SpikingResformer-CIFAR	10.83	0.493	4	78.73
	SpikingResformer+SSDP(w/o DA)	SpikingResformer-CIFAR	**10.83**	**0.504**	4	**79.48(+0.75)**
	SpikingResformer+DA-SSDP	SpikingResformer-CIFAR	**10.83**	**0.494**	4	**79.72(+0.99)**
ImageNet	Spiking ResNet (Hu et al. 2021)	ResNet-50	25.56	70.934	350	72.75
	Transformer (Dosovitskiy 2020)	Transformer-8-512	29.68	38.340	1	80.80
	Spikformer	Spikformer-8-384	16.81	7.734	4	70.24
	Spikformer	Spikformer-8-768	66.34	21.477	4	74.81
	Spikingformer	Spikingformer-8-512	29.68	7.46	4	74.79
	Spikingformer	Spikingformer-8-768	66.34	13.68	4	75.85
	Spike-driven Transformer	Spike-driven Transformer-8-768	66.34	6.09	4	76.32
	SpikingResformer	SpikingResformer-L	60.38	8.76	4	78.77
	SpikingResformer+SSDP(w/o DA)	SpikingResformer-L	**60.38**	**8.89**	4	**79.35(+0.58)**
	SpikingResformer+DA-SSDP	SpikingResformer-L	**60.38**	**8.76**	4	**79.50(+0.73)**

4.5 Experiments

We design our experiments to systematically evaluate the effectiveness of DA-SSDP in improving performance, refining temporal feature encoding, and energy cost in spiking transformer models. All experiments are conducted using publicly available vision datasets. Each experiment is repeated five times for statistical reliability. To maintain fairness, we preserve identical model structures and training regimens across variants, except where otherwise specified.

4.5.1 Experimental Setup

To ensure a fair comparison, we used the same architecture and hyperparameters as the baseline model, training for exactly 600 epochs. DA-SSDP uses a warm-up of E_{warm}=100 (for CIFAR10-DVS, E_{warm}=80) epochs to fit the gate and the fitted gate is then kept fixed for the remaining epochs. Kernel parameters are A_{+}=1.5×10^{-3}, A_{-}=1.0×10^{-4}, and a learnable σ. Backbones follow

Spikingresformer with $T{=}4$ steps on CIFAR-10/100 (Krizhevsky, Hinton et al. 2009) and ImageNet-1k (Deng et al. 2009), and $T{=}10$ on CIFAR10-DVS (Li et al. 2017).

4.5.2 Results

Table 4.1 summarises the results. With identical model structures and comparable training configurations, DA-SSDP yields consistent performance boosts: +0.42% on CIFAR-10, +0.99% on CIFAR-100, and +0.73% on ImageNet-1k. Model size and inference cost are unchanged; the rule adds only train-time computation at the hooked layers. As shown in Fig. 4.5, the median batch synchrony S_b increases from $3{\times}10^{-4}$ to $1{\times}10^{-2}$ ($\approx$33$\times$), indicating stronger co-activation of task-relevant channels.

On CIFAR10-DVS: CIFAR10-DVS is created by showing static CIFAR-10 images to a DVS sensor while moving the screen or camera, which produces event streams dominated by edge-related contrast changes. After the first convolution and pooling layers, the neural activity becomes quite sparse in time, and many pre-post channel pairs never fire together within the T steps, so the synchrony gate λ is often zero. Even when both sides do fire, their first spikes are usually far apart, making $|\Delta t|$ large and the Gaussian weight $g(\Delta t)$ close to zero. Channels that stay silent contribute small negative updates through the $-A_{-}g$ term, which pushes weights down even further. At the batch level, the synchrony score S_b varies only slightly across batches, so its correlation with the loss is weak. This makes the fitted slope k close to zero, and the dopamine gate essentially neutral ($G_b \approx 1$). In practice, DA-SSDP then behaves almost like its basic two-factor version, with only small net updates. This explains why the improvement on CIFAR10-DVS is modest compared to frame-based datasets, and shows that the rule is most effective when the data provide strong, task-relevant spike synchrony.

4.5.3 Baseline Comparison and Ablation

To quantify the contribution of the proposed rule, we compare

- **Baseline:** the spiking transformer trained only with surrogate backprop.
- **SSDP (two–factor):** local synchrony–based updates without dopamine gating.
- **DA-SSDP (three–factor):** the full method with a dopamine-modulated gate.

We evaluate all variants showed in Table 4.1 on CIFAR-10, CIFAR-100, CIFAR10-DVS, and ImageNet-1K. For each dataset, we report top-1 classification accuracy, the number of parameters, energy cost, and the number of time steps.

4.5.3.1 DA-SSDP Integration in the Model

We further examine the rule's optimal placement by applying it to (i) the linear classifier only, (ii) the DSSA block only, or (iii) both components at the same time. No further hyperparameter adjustments are made for each configuration.

TABLE 4.2: Effects of integrating SSDP into different parts of the model on CIFAR-100 dataset. Adding SSDP to the DSSA module yields the largest accuracy gain (+0.58%), whereas integrating it into the Classifier yields a +0.23% improvement. When placed in the Prologue, however, the model fails to converge.

Position	**Accuracy (%)**	**Improvement (%)**
Baseline (No SSDP)	78.73	–
Prologue+SSDP (w/o DA)	–	Divergent
DSSA+SSDP (w/o DA)	79.31	+0.58
DSSA+DA-SSDP	79.60	+0.87
Classifier+SSDP (w/o DA)	78.96	+0.23
DSSA+Classifier+SSDP (w/o DA)	79.48	+0.75
DSSA+Classifier+DA-SSDP	79.72	+0.99

With identical hyperparameters, the baseline achieves 78.73% Top-1 accuracy. Table 4.2 reports the effect of where the local rule is attached. Applying the update in the Prologue convolution will make training unstable in our setting (diverged under the same schedule). By contrast, attaching the rule to the last DSSA block yields the largest gain (+0.58% for two–factor SSDP; +0.87% with the dopamine gate), while adding it only to the linear classifier gives a smaller but positive gain (+0.23%). Using both hooks gives the best result (79.72%, +0.99%). No extra tuning was performed per placement, and the warm-up–fitted gate is kept fixed thereafter.

These observations are consistent with the mechanism of DA-SSDP. The rule acts as a post-update, synchrony-based local adjustment whose benefit depends on the presence of task-aligned co-activation. Such co-activation is more reliably expressed in mid-to-late representations than in very early feature extractors, where activity is sparser and more input-driven, which explains why the last DSSA benefits most while the Prologue is brittle under the same hyperparameters. Across placements, the dopamine

gate further improves over the two-factor variant, indicating that scaling the local update by the synchrony–loss relationship learned during warm-up is helpful for learning. Overall, DA-SSDP enhances model performance by injecting local synchrony signals into deeper layers and scaling them based on their correlation with the global backpropagation loss.

TABLE 4.3: Performance under different A_+ and A_- settings. The highest improvement is achieved with $A_+ = 0.00015$ and $A_- = 0.00005$.

DA	A_+	A_-	**Accuracy (%)**
✗	0.0015	0.0005	78.86
✗	0.00015	0.00005	79.48
✗	0.00010	0.00010	79.03
✗	0.001	0.001	Divergent
✓	0.0015	0.0005	**79.72**
✓	0.00015	0.00005	**79.40**
✓	0.00015	0.0001	**79.22**
✓	0.001	0.001	**79.21**

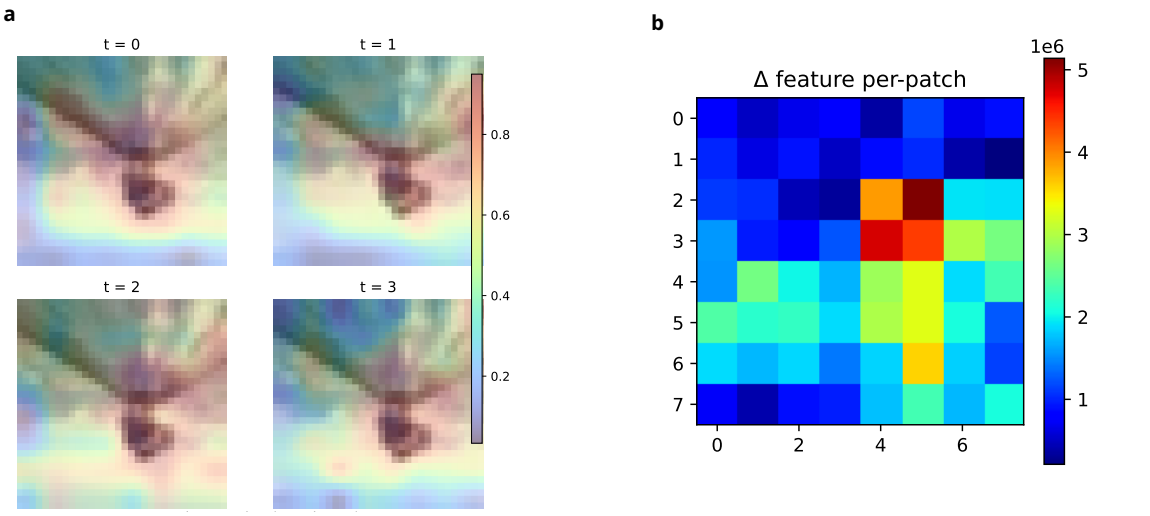
4.5.3.2 Impact of SSDP parameters on Convergence and Performance

To better understand how DA-SSDP handles parameter variation, we compared it against both the baseline and a variant using SSDP alone. We vary the potentiation/depression amplitudes (A_+, A_-) and compare the two–factor SSDP to the three–factor DA-SSDP with a fixed gate (learned during warm-up and then frozen). No additional hyperparameter retuning is performed per setting.

Hyperparameter sensitivity Sensitivity to (A_+, A_-): Table 4.3 highlights three key patterns (i) under high amplitudes ($A_+{=}A_-{=}10^{-3}$), the two-factor SSDP fails to converge, but DA-SSDP achieves 79.21% accuracy, demonstrating better stability with extreme hyperparameter; (ii) at moderate amplitudes, DA-SSDP matches or slightly exceeds SSDP (e.g., 79.48% vs. 79.40% at $1.5{\times}10^{-4}/5{\times}10^{-5}$) (iii) the best accuracy is obtained with a larger potentiation setting ($A_+{=}1.5{\times}10^{-3}$, $A_-{=}5{\times}10^{-4}$), where DA-SSDP reaches **79.72%** versus **78.86%** for SSDP.

Interpretation: These findings align with the dopamine gate's function in our implementation. After warm-up, it delivers a stable, batch-specific rescaling $G_b{=}1 + k\,\hat{S}_b$, for the local adjustment, which compresses the per-batch update magnitude and reduces sensitivity to overly large (A_+, A_-) without changing the update direction or the forward computation. When $k{\approx}0$ (weak synchrony-loss correlation), the method reduces to the two-factor baseline. Overall, DA-SSDP enables safe use of

larger potentiation/depression amplitudes and achieves equivalent or better accuracy within an identical training configuration.



Baseline attention maps (t=1∼4) Per-patch feature gain Δ induced by DA-SSDP

FIGURE 4.3: **Spatial effect of DA-SSDP in the last DSSA stage.** (a) Baseline attention maps (averaged over heads; $t = 1{\sim}4$) on the same test image. (b) Per-patch feature-gain $\Delta(u, v)$, computed as the absolute change of the post-projection activation between the DA-SSDP and baseline checkpoints and averaged over time and channels (brighter = larger change). Hotspots in Δ align with high-attention areas from (a), suggesting a re-weighting of already attended tokens rather than a wholesale redirection.

4.5.4 Visualisation and Activity Analysis

To explore the effect of DA-SSDP on temporal feature organisation, we used several types of analysis. A qualitative t-distributed Stochastic Neighbour Embedding (t-SNE) view of the test embeddings, with setup and plots, is provided in Appendix 4.7.

Spatial feature gain analysis: We examine how DA-SSDP modifies spatial evidence within the final DSSA block using a typical test image. Fig. 4.3(a) shows the baseline attention maps (averaged across heads and time steps). Fig. 4.3(b) depicts a per-patch feature gain map, calculated as the absolute difference in post-projection activations between the final DA-SSDP model and the baseline version,

as detailed in Eq. 4.15.

$$\Delta(u,v) = \frac{1}{TC}\sum_{t,c}\Big|\left(W_{\text{proj}}^{\text{DA}} A_t^{\text{DA}} V_t^{\text{DA}}\right)_c(u,v) - \left(W_{\text{proj}}^{\text{Base}} A_t^{\text{Base}} V_t^{\text{Base}}\right)_c(u,v)\Big| \,. \tag{4.15}$$

Hotspots in $\Delta(u,v)$ correspond to high attention zones in Fig. 4.3(a), aligning with our design. The local update connects to the attention layer, thereby preferentially recalibrating tokens with already prominent signals.

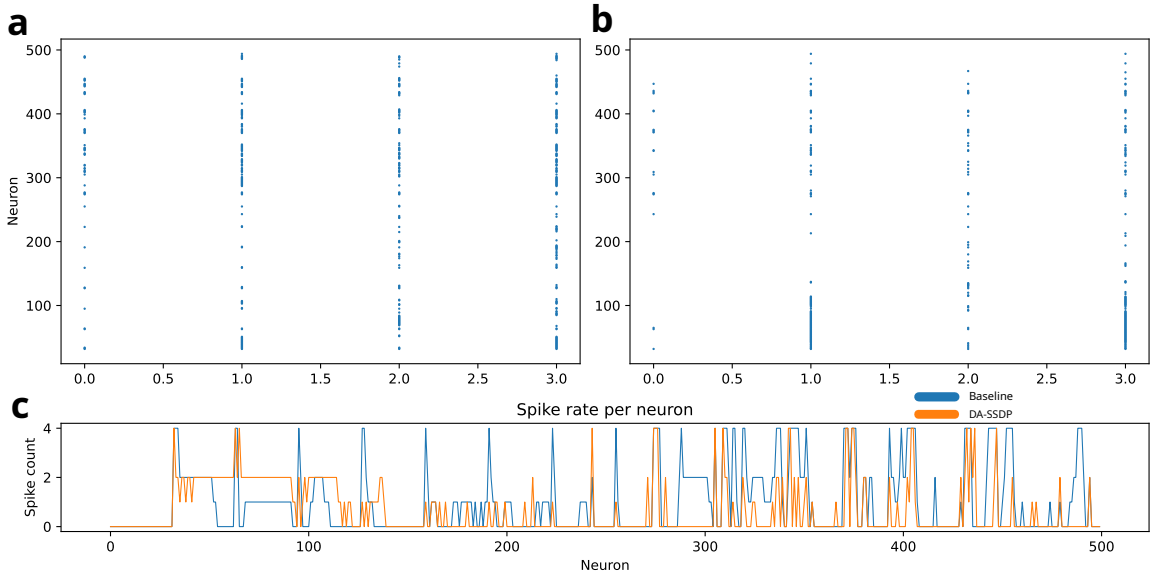


FIGURE 4.4: **Temporal spike patterns shaped by DA-SSDP.** Raster plots (top) and per–neuron spike counts (bottom) for the first 500 LIF/PLIF channels over $T = 4$ time steps, evaluated on a representative CIFAR-100 sample. Baseline activity is dense and widely spread, with nearly every channel firing at every step. After DA-SSDP training, a compact synchronous burst emerges around indices 30–120, whereas most channels emit at most a single spike or remain silent, yielding a markedly sparser profile.

Spike Activity and Rate Analysis: Fig. 4.4 demonstrates how DA-SSDP reorganises the network's spiking behaviours. In the baseline network, spike frequencies vary widely, and most neurons fire at every time step (4 spikes in total). After introducing DA-SSDP, neurons crucial for decisions become more prone to simultaneous activation, and as a result, these channels are reinforced and maintain elevated spiking rates, while more neurons emit at most one spike or stay inactive. Taken together with the results in Table 4.1, DA-SSDP strengthens neurons that convey useful information while suppressing those offering minimal value or just adding noise.

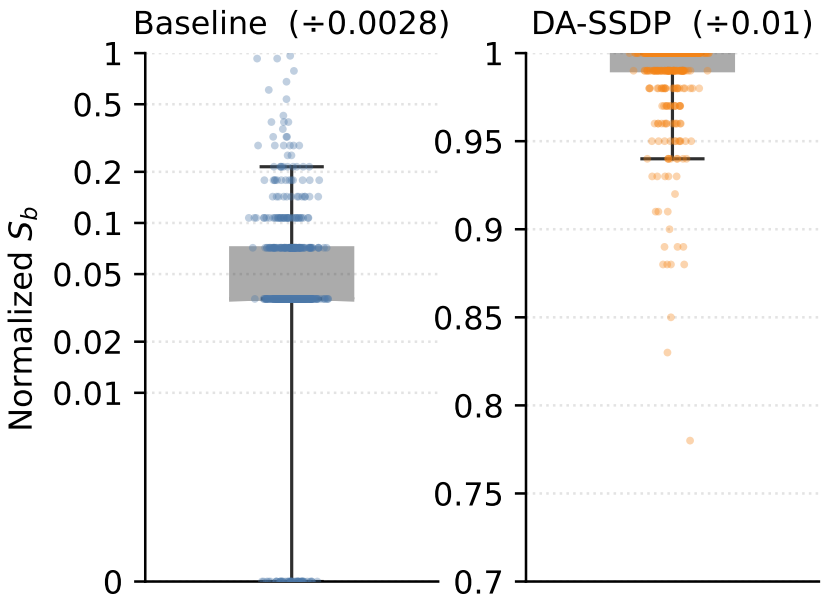


FIGURE 4.5: **Batch-level synchrony boxplot.** Each point summarises one mini-batch. For a batch b, we compute a binary spike indicator per channel over the temporal window T. The synchrony score S_b is then defined as the fraction of pre/post channel pairs that are simultaneously active within T. Compared with the vanilla baseline, **DA-SSDP** markedly shifts the distribution upward, raising the typical (median) synchrony from $\sim 3{\times}10^{-4}$ to $\sim 10^{-2}$ (about $30\times$) while tightening dispersion, consistent with the design goal that the loss-aware gate G_b enhances task-aligned spike coincidences and suppresses asynchronous activity.

These results confirm that the proposed plasticity rule not only reshapes temporal dynamics but also does so in a manner aligned with the model's performance.

Loss-aligned modulation via synchrony covariance: Using the gate G_b in Eq. 4.14 with the slope k from Eq. 4.13, DA-SSDP is expected to upweight batches whose synchrony S_b predicts lower loss and to stay neutral otherwise. Fig. 4.5 provides the mechanism-level signature of this behaviour. Relative to the vanilla model, the entire distribution of S_b on the test set shifts upward and contracts, with the median increasing from 3×10^{-4} to 1.0×10^{-2} (about $30\times$). This pattern indicates (i) more frequent task-aligned co-activation and (ii) reduced batch-to-batch variability in synchrony. Across random seeds, we observe the same qualitative shift. Taken together, the plot validates that the learned gate injects population-level information into the update stream and steers the network toward temporally stable firing regimes.

4.6 Computation of MACs, SOPs, and Estimated Energy

We follow the standard accounting used in prior SNN works (Yao et al. 2024; Zhou et al. 2022; Shi et al. 2024). All quantities are reported per image, per inference, and T denotes the number of simulation steps.

Synaptic operations. For a block or layer l in *SpikingResformer*, the synaptic operation count is related to its static arithmetic cost and the activity level of its inputs:

$$\mathrm{SOPs}(l) \;\approx\; \bar{f}_r(l) \;\times\; T \;\times\; \mathrm{MACs}(l), \tag{4.16}$$

where $\bar{f}_r(l)$ is the average input firing rate of block l, T is the simulation length, and $\mathrm{MACs}(l)$ is the multiply–accumulate count of l obtained by structural profiling (`thop` with custom counters). In our implementation, $\mathrm{SOPs}(l)$ are measured dynamically by a forward-hook monitor, which is consistent with the scaling in Eq. 4.16.

Model-level estimated energy. Assuming per-operation energy coefficients on a 45 nm process, $E_{\mathrm{MAC}} = 4.6\,\mathrm{pJ/MAC}$ and $E_{\mathrm{AC}} = 0.9\,\mathrm{pJ/SOP}$, the energy estimate of *SpikingResformer* is

$$\begin{aligned} E_{\textit{SpikingResformer}} = {} & E_{\mathrm{MAC}} \cdot \mathrm{MACs}\big(\mathrm{FL}^{1}_{\text{SNN-Conv}}\big) \\ & + E_{\mathrm{AC}} \cdot \left(\sum_{n=2}^{N} \mathrm{SOPs}^{n}_{\text{SNN-Conv}} \;+\; \sum_{m=1}^{M} \mathrm{SOPs}^{m}_{\text{SNN-FC}} \;+\; \sum_{\ell=1}^{L} \mathrm{SOPs}^{\ell}_{\text{DSSA}} \right), \end{aligned} \tag{4.17}$$

where $\mathrm{FL}^{1}_{\text{SNN-Conv}}$ denotes the first convolution that converts the static Red, Green, and Blue (RGB) image into spikes; the subsequent spiking convolution blocks, spiking fully-connected blocks, and DSSA blocks contribute activity-dependent Synaptic Operations (SOPs).

TABLE 4.4: SOPs and estimated (Est) energy per image (Img) on SpikingResformer-CIFAR. Estimated energy follows the mapping described in Appendix 4.6.

Method	**SOPs (G)**	**Est. energy per image (μJ/Img)**	$\Delta(\%)$**(μJ/Img)** *vs.* **Base**
Baseline	0.54846	493.62	–
DA-SSDP	0.56425	494.82	+0.24

Block-wise form. For an arbitrary block b,

$$\text{Estimated energy}_{\text{ANN}}(b) \;=\; 4.6\,\text{pJ} \times \text{MACs}(b), \qquad \text{Estimated energy}_{\text{SNN}}(b) \;=\; 0.9\,\text{pJ} \times \text{SOPs}(b), \tag{4.18}$$

which become millijoules when $\text{MACs}(b)$ or $\text{SOPs}(b)$ are expressed in billions.

4.7 Additional Visualisation: t-SNE of Test Embeddings

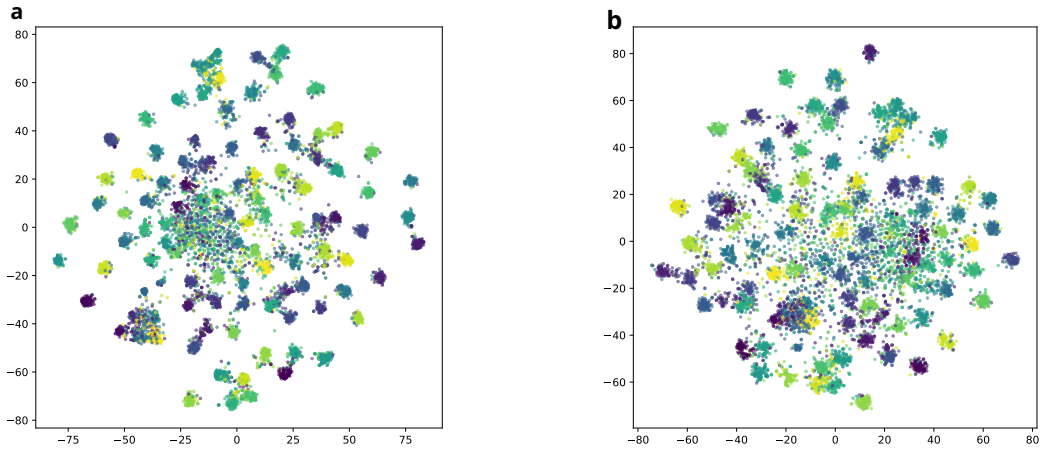
Fig. 4.6 compares the hidden state embeddings across the two models. In the baseline model, each class clusters into a compact, circular hub, resulting in wide, vacant spaces between hubs. It results in a configuration that commonly signals overfitting to the "simpler" examples near class centroids. On the other hand, in DA-SSDP, after applying the adaptive gate in Eq. 4.13 and Eq. 4.14, the original compact clusters extend into wider, sometimes stretched shapes. Samples belonging to the same class remain grouped, but sparse connections occasionally form between similar categories, creating smoother, rather than sharp class boundaries. Both panels use the same t-SNE configuration (identical hyperparameters and random seed) applied to the same test features, and thus the differences reflect the learned representations rather than projection randomness.

The reason is that the gate only adjusts the strength of the original SSDP update, enhancing potentiation for samples with higher synchrony and lower loss ($S_b > \mu_S$ and $k > 0$), while for samples with below-average synchrony, it reduces both potentiation and depression. As a result, the network reinforces typical firing patterns without forcing all data points into overly tight clusters. Samples near decision boundaries or with unusual features receive gentler updates, creating a more flexible decision boundary. Although the resulting geometry appears less organised in two dimensions, it corresponds to the improved Top-1 accuracy achieved by DA-SSDP (79.66% vs. 78.73% baseline). In essence, this seeming drop in visual tightness embodies a synchrony-driven regularisation that enhances the model's ability to generalise.

4.8 Discussion and Conclusion

Loss-modulated synchrony as a scalable training-time signal

DA-SSDP incorporates a group-level signal into the training updates. Batch synchrony S_b serves as a rough gauge of task-relevant simultaneous firing and impacts training solely through the gate G_b, and



Baseline t-SNE at test accuracy (78.73%). DA-SSDP t-SNE at test accuracy (79.66%).

FIGURE 4.6: (a) t-SNE visualisation of hidden-layer embeddings after the final attention stage for the baseline model, exhibiting compact, well-separated clusters. (b) t-SNE visualisation for the DA-SSDP model, exhibiting broader intra-class manifolds and smooth, low-density "bridges" that connect semantically related classes, indicating softer inter-class boundaries. Each dot represents a single sample from a two-dimensional t-SNE projection, and its colour indicates the class label. A cluster is a patch of dots of the same colour that lie close together. The cluster area measures how far samples of a given class spread in the feature space (small area = high tightness). The gap between clusters reflects the distance between the class centres (a large gap = clear separation).

is inherently tuned to the supervised goal. As such, backpropagation continues as the core optimiser, whereas DA-SSDP adds a supportive inclination towards reliable, low-noise spiking dynamics in channels prone to join activation on well-predicted examples. This inclination disappears at inference time, leaving the forward-pass operations intact.

Effect size tracks task-aligned co-activation in deeper layers

The rule is most effective in mid to late representations where task-aligned co-activation is reliably expressed. Applying the update to very early layers can be brittle under the same hyperparameters, consistent with sparser, input-driven activity in those stages. The dopamine-like gate, fitted during warm-up, rescales the local update per batch, improving stability and enabling the safe use of larger potentiation and depression amplitudes.

Boundary conditions and stability under weak synchrony

The gains rely on the availability of synchrony linked to class-specific patterns. When the synchrony-loss correlation is weak, as observed on CIFAR10-DVS, the fitted slope collapses toward zero, the

gate becomes neutral, and DA-SSDP reduces to its two-factor baseline with minimal effect. These properties clarify where the mechanism proves useful and ensure a safe degradation when synchrony offers little insight.

Limitations and future directions

While DA-SSDP shows promising improvements in accuracy, temporal coding, and estimated energy efficiency, several limitations remain. We outline some limitations of our proposed DA-SSDP, which we are currently investigating and improving.

(A) While our experiments in this chapter demonstrate performance gains on standard vision benchmarks, a broader validation is needed. In particular, evaluations on neuromorphic hardware platforms like Loihi or SpiNNaker could provide stronger evidence of generalisation and practical utility.

(B) On datasets such as DVS, spike counts are extremely sparse and often distributed across wide temporal spans. Reliable performance typically requires a larger T to capture sufficient information. In such regimes, the synchrony signal of DA-SSDP is suppressed by the Gaussian kernel, leading to near-zero incremental benefit over weights.

(C) We acknowledge that while DA-SSDP is motivated by biological plausibility, the introduction of a direct loss-derived global signal diminishes strict biological alignment. An interesting direction for future work is to explore alternative modulatory signals inspired by biological dopaminergic pathways, which may better balance biological compatibility with task-driven optimisation.

Future work includes extending synchrony-aware plasticity to self-supervised or pretraining regimes, exploring additional hook placements in attention pathways, and designing curriculum schedules that shape synchrony over time. A second direction is to couple DA-SSDP with online neuromodulatory signals in continual or reinforcement learning scenarios. Finally, we plan to quantify hardware effects on real accelerators by measuring event traffic, latency, and energy under fixed-weight deployment.

CHAPTER 5

Conclusion

This thesis investigated brain-inspired learning signals for deep spiking neural networks with a focus on population-level synchrony. We introduced Spike-Synchrony-Dependent Plasticity (SSDP), a rule that strengthens synapses when pre- and post-synaptic neurons co-fire within the setting timestep. SSDP relies only on binary spike indicators and first-spike latencies, requires only a single accumulator pass over the weight matrix, and leaves the forward computation graph unchanged. We then proposed Dopamine-Modulated Spike-Synchrony-Dependent Plasticity (DA-SSDP), which gates the synchrony update using a task-level feedback signal. After a brief warm-up to estimate the correlation between batch synchrony and the loss, the gate is fixed and thereafter rescaled to the local update without altering the training schedule.

Across static vision, event-driven vision, and auditory time-series tasks, SSDP and DA-SSDP improved accuracy and convergence stability relative to backpropagation-only baselines while preserving model size and inference cost. Gains were observed in both recurrent Spiking Neural Networks (SNNs) and spiking transformers. Analyses indicated that synchrony-based updates reduce spike-time jitter and increase class-relevant co-firing.

5.1 Mechanistic Insights and Broader Implications

By integrating the findings across the preceding chapters, several key insights emerge regarding the role of event-structure regularisers in spiking models.

5.1.1 Loss-Modulated Synchrony as a Scalable Signal

Mechanistically, SSDP operates on population co-activation within a short coincidence window: near-synchronous pre-post activity receives potentiation, and non-coactive pairs receive weak depression.

DA-SSDP elevates this by incorporating a group-level signal into the training updates. Batch synchrony S_b serves as a rough gauge of task-relevant simultaneous firing and impacts training solely through the gate G_b, inherently tuned to the supervised goal. As such, backpropagation remains the core optimiser, whereas DA-SSDP provides a supportive bias towards reliable, low-noise spiking dynamics in channels prone to joint activation on well-predicted examples. Ultimately, injecting an event-structure signal at training time yields more organised spatial codes and more reliable spike timing under realistic budgets.

5.1.2 Boundary Conditions

The proposed rule is most effective in mid-to-late representations where task-aligned co-activation is reliably expressed. Applying the update to very early layers can be brittle under the same hyperparameters, consistent with sparser, input-driven activity in those stages. The dopamine-like gate, fitted during warm-up, rescales the local update per batch, improving stability and enabling the safe use of larger potentiation and depression amplitudes.

Crucially, the gains rely on the availability of synchrony linked to class-specific patterns. When the synchrony-loss correlation is weak, as observed on highly sparse datasets like Canadian Institute for Advanced Research (CIFAR)10-Dynamic Vision Sensor (DVS), the fitted slope collapses toward zero, the gate becomes neutral, and DA-SSDP gracefully reduces to its two-factor baseline with minimal effect. These properties clarify where the mechanism proves useful and ensure safe degradation when synchrony offers little actionable insight.

5.2 Limitations

While SSDP and DA-SSDP show promising improvements in accuracy, temporal coding, and estimated energy efficiency, several limitations have been identified:

(A) **Kernel and Window Sensitivity**. The quality of the synchrony signal heavily depends on the window choice and kernel shape. When synchrony is weakly related to the objective, the ungated rule brings limited benefit.

(B) **Sparsity on Event Streams**. On datasets such as DVS, spike counts are extremely sparse and often distributed across wide temporal spans. Reliable performance typically requires a larger

T to capture sufficient information. In such regimes, the synchrony signal of DA-SSDP is suppressed by the Gaussian kernel, leading to near-zero incremental benefit over weights.

(C) **Assumption of Distribution Stability**. DA-SSDP requires a short calibration stage and assumes that the learned gate remains valid throughout the remainder of training. A strong distribution shift later in the training process can violate this assumption.

(D) **Biological Alignment** *vs.* **Optimisation**. While DA-SSDP is motivated by biological plausibility, the introduction of a direct loss-derived global signal diminishes strict biological alignment in favour of task-driven optimisation.

5.3 Future Research Directions

Although the present study demonstrates that SSDP and DA-SSDP enhance stability and accuracy in modern Spiking Neural Network (SNN) architectures, the mechanisms are not yet well-established drop-in solutions. The next step is to make synchrony and its gate truly adaptive: learn the effective window or kernel from data, allow layer-specific forms, and co-optimise an online gate with the task loss via Backpropagation (BP). Beyond the datasets tested above, evaluation should target richer temporal structures, including deeper spiking Transformers, multi-task encoders, and cross-modal problems that combine event-based vision with audio or Electroencephalogram (EEG). Given its event-driven nature, SSDP is also a candidate driver for online and continual learning on streams, where single-pass updates, label scarcity, sensor noise, and distributional shift are the norm. A clear question is whether synchrony updates can act as the primary learning signal rather than only an auxiliary signal alongside surrogate-gradient BP.

Progress further depends on algorithm-hardware co-design. Building on this need, as introduced in Chapter 4, DA-SSDP serves as a training-time regulariser, ideally suited for offline use on Graphics Processing Unit (GPU) to generate fixed weights ready for deployment. Parts of DA-SSDP are highly hardware-friendly. Neuromodulators like dopamine can be implemented as simple scalar multipliers or broadcast operations on Application-Specific Integrated Circuits (ASICs), avoiding complex per-synapse computations. Local aspects, such as detecting spike coincidences, operate on a per-synapse or per-neuron-pair basis using straightforward binary spike markers and time offsets. These can be processed efficiently on neuromorphic processors, capitalising on their built-in event-triggered synaptic modification capabilities. By concentrating spikes into fewer, more synchronised channels and

reducing incidental activity, the learned weights indirectly improve inference efficiency on event-driven hardware.

From a neuroscience standpoint, our use of a Gaussian-shaped kernel is a modelling choice encoding the simple hypothesis that "closer synchrony yields stronger potentiation". Testing this shape against physiological data remains a future task. Real neural synchrony involves more than coincident spikes; phase relations, oscillatory context, dendritic nonlinearities, and subthreshold dynamics are likely to matter. Future work should therefore explore alternative modulatory signals inspired by biological dopaminergic pathways, mapping them to plasticity rules that remain compatible with efficient digital or mixed-signal implementations. Pursuing these directions will clarify when and why synchrony-based learning helps, extend this rule to more demanding temporal problems, and ultimately move the approach toward robust, real-time, energy-efficient neuromorphic systems.

Bibliography

Amir, Arnon et al. (2017). 'A low power, fully event-based gesture recognition system'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252.

Anisimova, Margarita et al. (2023). 'Spike-timing-dependent plasticity rewards synchrony rather than causality'. In: *Cerebral Cortex* 33.1, pp. 23–34.

Apolinario, Marco Paul E and Kaushik Roy (2023). 'S-TLLR: STDP-inspired Temporal Local Learning Rule for Spiking Neural Networks'. In: *Transactions on Machine Learning Research*.

Arenas, Alex et al. (2008). 'Synchronization in complex networks'. In: *Physics Reports* 469.3, pp. 93–153.

Averbeck, Bruno B, Peter E Latham and Alexandre Pouget (2006). 'Neural correlations, population coding and computation'. In: *Nature Reviews Neuroscience* 7.5, pp. 358–366.

Baruchi, Itay et al. (2008). 'The emergence and properties of mutual synchronization in in vitro coupled cortical networks'. In: *European Journal of Neuroscience* 28.9, pp. 1825–1835.

Bekolay, Trevor et al. (2014). 'Nengo: A Python Tool for Building Large-Scale Functional Brain Models'. In: *Frontiers in Neuroinformatics* 7, p. 48. DOI: 10.3389/fninf.2013.00048.

Bellec, Guillaume et al. (2020a). 'A Solution to the Learning Dilemma for Recurrent Networks of Spiking Neurons'. In: *Nature Communications* 11, p. 3625. DOI: 10.1038/s41467-020-17236-y.

Bellec, Guillaume et al. (2020b). 'A solution to the learning dilemma for recurrent networks of spiking neurons'. In: *Nature Communications* 11, p. 3625. DOI: 10.1038/s41467-020-17236-y.

Benna, Marcus K. and Stefano Fusi (2016). 'Computational principles of synaptic memory consolidation'. In: *Nature Neuroscience* 19, pp. 1697–1706. DOI: 10.1038/nn.4401.

Bi, Guo-qiang and Mu-ming Poo (1998). 'Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type'. In: *The Journal of Neuroscience* 18.24, pp. 10464–10472. DOI: 10.1523/JNEUROSCI.18-24-10464.1998.

Bohte, S. M., J. N. Kok and H. La Poutre (2002). 'Error-Backpropagation in Temporally Encoded Networks of Spiking Neurons'. In: *Neural Computation* 14.6, pp. 1527–1546. DOI: 10.1162/089976602320264033.

Bottou, Léon (2010). 'Large-scale machine learning with stochastic gradient descent'. In: *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, pp. 177–186.

Brette, Romain (2012). 'Computing with neural synchrony'. In: *PLoS Computational Biology* 8.6, e1002561.

— (2015). 'Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain'. In: *Frontiers in Systems Neuroscience* 9, p. 151. DOI: 10.3389/fnsys.2015.00151.

Buzsáki, György (2010). 'Neural syntax: cell assemblies, synapsembles, and readers'. In: *Neuron* 68.3, pp. 362–385.

Buzsáki, György and Andreas Draguhn (2004). 'Neuronal Oscillations in Cortical Networks'. In: *Science* 304.5679, pp. 1926–1929. DOI: 10.1126/science.1099745.

Cao, Yongqiang, Yang Chen and Deepak Khosla (2015). 'Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition'. In: *International Journal of Computer Vision*.

Caporale, Natalia and Yang Dan (2008). 'Spike Timing–Dependent Plasticity: A Hebbian Learning Rule'. In: *Annual Review of Neuroscience* 31, pp. 25–46. DOI: 10.1146/annurev.neuro.31.060407.125639.

Clopath, Claudia et al. (2010). 'Connectivity reflects coding: a model of voltage-based STDP with homeostasis'. In: *Nature Neuroscience* 13.3, pp. 344–352.

Cohen, Gregory K et al. (2016). 'Skimming digits: neuromorphic classification of spike-encoded images'. In: *Frontiers in Neuroscience* 10, p. 184.

Cramer, Benjamin et al. (2020a). 'The heidelberg spiking data sets for the systematic evaluation of spiking neural networks'. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7, pp. 2744–2757.

— (2020b). 'The heidelberg spiking data sets for the systematic evaluation of spiking neural networks'. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7, pp. 2744–2757.

Cramer, Benjamin et al. (2022). 'Surrogate gradients for analog neuromorphic computing'. In: *Proceedings of the National Academy of Sciences* 119.4, e2109194119.

Davies, Mike et al. (2018a). 'Loihi: A Neuromorphic Manycore Processor with On-Chip Learning'. In: *IEEE Micro* 38.1, pp. 82–99. DOI: 10.1109/MM.2018.112130359.

Davies, Mike et al. (2018b). 'Loihi: A Neuromorphic Manycore Processor with On-Chip Learning'. In: *IEEE Micro* 38.1, pp. 82–99.

Dayan, Peter and L. F. Abbott (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press.

Deng, Jia et al. (2009). 'Imagenet: A large-scale hierarchical image database'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, pp. 248–255.

Diehl, Peter U et al. (2015). 'Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing'. In: *2015 International joint conference on neural networks (IJCNN)*. ieee, pp. 1–8.

Diehl, Peter U. and Matthew Cook (2015). 'Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity'. In: *Frontiers in Computational Neuroscience* 9, p. 99. DOI: 10.3389/fncom.2015.00099.

Dong, Yiting et al. (2023). 'An unsupervised STDP-based spiking neural network inspired by biologically plausible learning rules and connections'. In: *Neural Networks* 165, pp. 799–808.

Dosovitskiy, Alexey (2020). 'An image is worth 16x16 words: Transformers for image recognition at scale'. In: *ArXiv preprint arXiv:2010.11929*.

Engel, Andreas K, Pascal Fries and Wolf Singer (2001). 'Dynamic predictions: oscillations and synchrony in top–down processing'. In: *Nature Reviews Neuroscience* 2.10, pp. 704–716.

Eshraghian, Jason K et al. (2023). 'Training spiking neural networks using lessons from deep learning'. In: *Proceedings of the IEEE* 111.9, pp. 1016–1054.

Esser, Steven K. et al. (2016). 'Convolutional networks for fast, energy-efficient neuromorphic computing'. In: *Proceedings of the National Academy of Sciences* 113.41, pp. 11441–11446.

Fang, Wei et al. (2021). 'Deep residual learning in spiking neural networks'. In: *Advances in Neural Information Processing Systems* 34, pp. 21056–21069.

Fang, Wei et al. (2023). 'Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence'. In: *Science Advances* 9.40, eadi1480.

Feldman, Daniel E (2012). 'The spike-timing dependence of plasticity'. In: *Neuron* 75.4, pp. 556–571.

Florian, Răzvan V (2007). 'Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity'. In: *Neural Computation* 19.6, pp. 1468–1502.

Frémaux, Nicolas and Wulfram Gerstner (2015). 'Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules'. In: *Frontiers in Neural Circuits* 9, p. 85. DOI: 10.3389/fncir.2015.00085.

— (2016). 'Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules'. In: *Frontiers in Neural Circuits* 9, p. 85. DOI: 10.3389/fncir.2015.00085.

Frey, Uwe and Richard GM Morris (1997). 'Synaptic tagging and long-term potentiation'. In: *Nature* 385.6616, pp. 533–536.

Friedmann, Simon et al. (2016). 'Demonstrating hybrid learning in a flexible neuromorphic hardware system'. In: *IEEE Transactions on Biomedical Circuits and Systems* 11.1, pp. 128–142.

Fries, Pascal (2005). 'A mechanism for cognitive dynamics: neuronal communication through neuronal coherence'. In: *Trends in Cognitive Sciences* 9.10, pp. 474–480.

Fries, Pascal (2015). 'Rhythms for Cognition: Communication through Coherence'. In: *Neuron* 88.1, pp. 220–235. DOI: 10.1016/j.neuron.2015.09.034.

Furber, Steve B et al. (2014). 'The SpiNNaker project'. In: *Proceedings of the IEEE* 102.5, pp. 652–665.

Gallego, Guillermo et al. (2020). 'Event-based Vision: A Survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. early access.

Gerstner, Wulfram and Werner M. Kistler (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.

Gollisch, T. and M. Meister (2008). 'Rapid neural coding in the retina with relative spike latencies'. In: *Science* 319.5866, pp. 1108–1111.

Goupy, Gaspard, Pierre Tirilly and Ioan Marius Bilasco (2024a). 'Neuronal competition groups with supervised STDP for spike-based classification'. In: *Advances in Neural Information Processing Systems*.

— (2024b). 'Paired competing neurons improving STDP supervised local learning in Spiking Neural Networks'. In: *Frontiers in Neuroscience* 18, p. 1401690.

Gütig, Robert and Haim Sompolinsky (2006). 'The tempotron: a neuron that learns spike timing–based decisions'. In: *Nature Neuroscience* 9.3, pp. 420–428.

Gygax, Julia and Friedemann Zenke (2025). 'Elucidating the Theoretical Underpinnings of Surrogate Gradient Learning in Spiking Neural Networks'. In: *Neural Computation* 37.5, pp. 886–925. DOI: 10.1162/neco_a_01752.

Han, Bing, Gopalakrishnan Srinivasan and Kaushik Roy (2020). 'Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13558–13567.

Han, Song et al. (2015). 'Learning both weights and connections for efficient neural network'. In: *Advances in Neural Information Processing Systems* 28.

Hao, Yunzhe et al. (2020). 'A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule'. In: *Neural Networks* 121, pp. 387–395.

Harris, Alexander Z and Joshua A Gordon (2015). 'Long-range neural synchrony in behavior'. In: *Annual Review of Neuroscience* 38.1, pp. 171–194.

Harris, Kenneth D (2005). 'Neural signatures of cell assembly organization'. In: *Nature Reviews Neuroscience* 6.5, pp. 399–407.

Hebb, Donald Olding (2005a). *The organization of behavior: A neuropsychological theory*. Psychology Press.

— (2005b). *The organization of behavior: A neuropsychological theory*. Psychology press.

Hennig, Jay A et al. (2021). 'How learning unfolds in the brain: toward an optimization view'. In: *Neuron* 109.23, pp. 3720–3735.

Hu, Yangfan, Huajin Tang and Gang Pan (2021). 'Spiking deep residual networks'. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.8, pp. 5200–5205.

Hu, Yangfan et al. (2024). 'Toward Large-scale Spiking Neural Networks: A Comprehensive Survey and Future Directions'. In: *arXiv preprint arXiv:2409.02111*. URL: https://arxiv.org/abs/2409.02111.

Huguenard, John R and David A McCormick (2007). 'Thalamic synchrony and dynamic regulation of global forebrain oscillations'. In: *Trends in Neurosciences* 30.7, pp. 350–356.

Indiveri, Giacomo and Shih-Chii Liu (2015a). 'Memory and information processing in neuromorphic systems'. In: *Proceedings of the IEEE* 103.8, pp. 1379–1397. DOI: 10.1109/JPROC.2015.2444094.

— (2015b). 'Memory and information processing in neuromorphic systems'. In: *Proceedings of the IEEE* 103.8, pp. 1379–1397.

Izhikevich, Eugene M (2007a). 'Solving the distal reward problem through linkage of STDP and dopamine signaling'. In: *Cerebral Cortex* 17.10, pp. 2443–2452.

— (2007b). 'Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling'. In: *Cerebral Cortex* 17.10, pp. 2443–2452. DOI: 10.1093/cercor/bhl152.

Jiang, Tingting et al. (2023). 'Adaptive deep spiking neural network with global-local learning via balanced excitatory and inhibitory mechanism'. In: *The Twelfth International Conference on Learning Representations*.

Katz, Larry C and Carla J Shatz (1996). 'Synaptic activity and the construction of cortical circuits'. In: *Science* 274.5290, pp. 1133–1138.

Kehayas, Vassilis and Anthony Holtmaat (2015). 'Dissonant synapses shall be punished'. In: *Neuron* 87.2, pp. 245–247.

Kembay, Assel, Karina Aguilar and Jason Eshraghian (2025). 'A quantitative analysis of catastrophic forgetting in quantized spiking neural networks'. In: *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, pp. 1–5.

Kheradpisheh, Saeed R. et al. (2018). 'STDP-based spiking deep neural networks for object recognition'. In: *Neural Networks* 99, pp. 56–67.

Kingma, Diederik P and Jimmy Ba (2014). 'Adam: A method for stochastic optimization'. In: *ArXiv preprint arXiv:1412.6980*.

Krizhevsky, Alex, Geoffrey Hinton et al. (2009). 'Learning multiple layers of features from tiny images'. In.

Krogh, Anders and John Hertz (1991). 'A simple weight decay can improve generalization'. In: *Advances in Neural Information Processing Yystems* 4.

KRUEGER, JAMES M and FERENC OBÄL JR (1993). 'A neuronal group theory of sleep function'. In: *Journal of Sleep Research* 2.2, pp. 63–69.

Lapicque, Louis (1907). 'Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation'. In: *Journal de Physiologie et de Pathologie Générale* 9, pp. 620–635.

— (2007). 'Quantitative investigations of electrical nerve excitation treated as polarization. 1907'. In: *Biological Cybernetics* 97.5-6, pp. 341–349.

Lee, Jun Haeng, Tobi Delbruck and Michael Pfeiffer (2016). 'Training deep spiking neural networks using backpropagation'. In: *Frontiers in Neuroscience* 10, p. 508.

Lewicki, Michael S (2002). 'Efficient coding of natural sounds'. In: *Nature Neuroscience* 5.4, pp. 356–363.

Li, Hongmin et al. (2017). 'CIFAR10-DVS: an event-stream dataset for object classification'. In: *Frontiers in Neuroscience* 11, p. 309.

Lichtsteiner, Patrick, Christoph Posch and Tobi Delbruck (2008). 'A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor'. In: *IEEE Journal of Solid-State Circuits* 43.2, pp. 566–576.

Liu, Fangxin et al. (2021). ‘SSTDP: Supervised spike timing dependent plasticity for efficient spiking neural network training’. In: *Frontiers in Neuroscience* 15, p. 756876.

Loshchilov, Ilya and Frank Hutter (2016). ‘Sgdr: Stochastic gradient descent with warm restarts’. In: *ArXiv preprint arXiv:1608.03983*.

Maass, Wolfgang (1997). ‘Networks of spiking neurons: the third generation of neural network models’. In: *Neural Networks* 10.9, pp. 1659–1671.

Majhi, Soumen et al. (2024). ‘Patterns of neuronal synchrony in higher-order networks’. In: *Physics of Life Reviews*.

Malyshev, Aleksey et al. (2013). ‘Energy-efficient encoding by shifting spikes in neocortical neurons’. In: *European Journal of Neuroscience* 38.8, pp. 3181–3188.

Markram, Henry et al. (1997). ‘Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs’. In: *Science* 275.5297, pp. 213–215.

Masquelier, Timothée and Simon J. Thorpe (2007). ‘Unsupervised Learning of Visual Features Through Spike Timing Dependent Plasticity’. In: *PLoS Computational Biology* 3.2, e31. DOI: 10.1371/journal.pcbi.0030031.

Mazurek, Szymon et al. (2025). ‘Three-Factor Learning in Spiking Neural Networks: An Overview of Methods and Trends from a Machine Learning Perspective’. In: *arXiv preprint arXiv:2504.05341*.

Merolla, Paul A. et al. (2014). ‘A Million Spiking-Neuron Integrated Circuit With a Scalable Communication Network and Interface’. In: *Science* 345.6197, pp. 668–673. DOI: 10.1126/science.1254642.

Mostafa, Hesham (2017). ‘Supervised Learning Based on Temporal Coding in Spiking Neural Networks’. In: *Neural Computation* 29.8, pp. 2151–2177. DOI: 10.1162/neco_a_00961.

Neftci, Emre O, Hesham Mostafa and Friedemann Zenke (2019). ‘Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks’. In: *IEEE Signal Processing Magazine* 36.6, pp. 51–63.

Olshausen, Bruno A and David J Field (1996). ‘Emergence of simple-cell receptive field properties by learning a sparse code for natural images’. In: *Nature* 381.6583, pp. 607–609.

Pascanu, Razvan, Tomas Mikolov and Yoshua Bengio (2013). 'On the Difficulty of Training Recurrent Neural Networks'. In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1310–1318.

Paszke, Adam et al. (2019). 'Pytorch: An imperative style, high-performance deep learning library'. In: *Advances in neural information processing systems* 32.

Patel, Adeetya, Michael Eickenberg and Eugene Belilovsky (2023). 'Local learning with neuron groups'. In: *arXiv preprint arXiv:2301.07635*.

Perez-Nieves, Nicolas and Dan Goodman (2021). 'Sparse spiking gradient descent'. In: *Advances in Neural Information Processing Systems* 34, pp. 11795–11808.

Perez-Nieves, Nicolas et al. (2021). 'Neural heterogeneity promotes robust learning'. In: *Nature Communications* 12.1, p. 5791.

Pfeiffer, Michael and Thomas Pfeil (2018). 'Deep learning with spiking neurons: Opportunities and challenges'. In: *Frontiers in Neuroscience* 12, p. 409662.

Pfister, Jean-Pascal and Wulfram Gerstner (2006). 'Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity'. In: *Journal of Neuroscience* 26.38, pp. 9673–9682. DOI: 10.1523/JNEUROSCI.1425-06.2006.

Popovych, Oleksandr V, Serhiy Yanchuk and Peter A Tass (2013). 'Self-organized noise resistance of oscillatory neural networks with spike timing-dependent plasticity'. In: *Scientific Reports* 3.1, p. 2926.

Quintana, Fernando M, Fernando Perez-Pena and Pedro L Galindo (2022). 'Bio-plausible digital implementation of a reward modulated STDP synapse'. In: *Neural Computing and Applications* 34.18, pp. 15649–15660.

Rathi, Nitin et al. (2020). 'Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation'. In: *arXiv preprint arXiv:2005.01807*.

Rieke, Fred et al. (1997). *Spikes: Exploring the Neural Code*. MIT Press.

Rossbroich, Julian, Julia Gygax and Friedemann Zenke (2022). 'Fluctuation-driven initialization for spiking neural network training'. In: *Neuromorphic Computing and Engineering* 2.4, p. 044016.

Roy, Kaushik, Akhilesh Jaiswal and Priyadarshini Panda (2019). 'Towards spike-based machine intelligence with neuromorphic computing'. In: *Nature* 575.7784, pp. 607–617.

Rueckauer, Bodo et al. (2017). 'Conversion of Continuous-Valued Deep Networks to Efficient Spiking Neural Networks'. In: *Frontiers in Neuroscience* 11, p. 682. DOI: 10.3389/fnins.2017.00682.

Rullen, R. van and S. Thorpe (2001). 'Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex'. In: *Neural Computation* 13.6, pp. 1255–1283.

Saranirad, Vahid et al. (2024). 'CDNA-SNN: A New Spiking Neural Network for Pattern Classification Using Neuronal Assemblies'. In: *IEEE Transactions on Neural Networks and Learning Systems*.

Schemmel, Johannes et al. (2010). 'A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling'. In: *2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN.2010.5596480.

Sengupta, Abhronil et al. (2019). 'Going deeper in spiking neural networks: VGG and residual architectures'. In: *Frontiers in Neuroscience* 13, p. 95.

Sharifshazileh, Mohammadali et al. (2021). 'An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG'. In: *Nature Communications* 12.1, p. 3095.

Shi, Xinyu, Zecheng Hao and Zhaofei Yu (2024). 'SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5610–5619.

Singer, Wolf (1999a). 'Neuronal synchrony: a versatile code for the definition of relations?' In: *Neuron* 24.1, pp. 49–65.

— (1999b). 'Neuronal synchrony: a versatile code for the definition of relations?' In: *Neuron* 24.1, pp. 49–65.

Song, Sen, Kenneth D Miller and Larry F Abbott (2000). 'Competitive Hebbian learning through spike-timing-dependent synaptic plasticity'. In: *Nature Neuroscience* 3.9, pp. 919–926.

Speranza, Luisa et al. (2021). 'Dopamine: the neuromodulator of long-term synaptic plasticity, reward and movement control'. In: *Cells* 10.4, p. 735.

Srivastava, Nitish et al. (2014). 'Dropout: a simple way to prevent neural networks from overfitting'. In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Stimberg, Marcel, Romain Brette and Dan F. M. Goodman (2019). 'Brian 2, an Intuitive and Efficient Neural Simulator'. In: *eLife* 8, e47314. DOI: 10.7554/eLife.47314.

Storrs, Katherine R, Barton L Anderson and Roland W Fleming (2021). 'Unsupervised learning predicts human perception and misperception of gloss'. In: *Nature Human Behaviour* 5.10, pp. 1402–1417.

Stuijt, Jan et al. (2021). 'μBrain: An event-driven and fully synthesizable architecture for spiking neural networks'. In: *Frontiers in Neuroscience* 15, p. 664208.

Szegedy, Christian et al. (2016). 'Rethinking the inception architecture for computer vision'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pp. 2818–2826.

Tavanaei, Amirhossein et al. (2019). 'Deep learning in spiking neural networks'. In: *Neural Networks* 111, pp. 47–63.

Thorpe, S., A. Delorme and R. van Rullen (2001). 'Spike-based strategies for rapid processing'. In: *Neural Networks* 14.6, pp. 715–725.

Tian, Yuchen et al. (2025). 'Beyond Pairwise Plasticity: Group-Level Spike Synchrony Facilitates Efficient Learning in Spiking Neural Networks'. In: *arXiv preprint arXiv:2505.14841*.

Turrigiano, Gina G. (2012). 'Homeostatic Synaptic Plasticity: Local and Global Mechanisms for Stabilizing Neuronal Function'. In: *Cold Spring Harbor Perspectives in Biology* 4.1, a005736. DOI: 10.1101/cshperspect.a005736.

Uhlhaas, Peter et al. (2009). 'Neural synchrony in cortical networks: history, concept and current status'. In: *Frontiers in Integrative Neuroscience* 3, p. 543.

Vinck, Martin et al. (2023). 'Principles of large-scale neural interactions'. In: *Neuron* 111.7, pp. 987–1002. DOI: 10.1016/j.neuron.2023.03.015.

Wu, Yujie et al. (2018). 'Spatio-temporal backpropagation for training high-performance spiking neural networks'. In: *Frontiers in Neuroscience* 12, p. 331.

Xiao, Mingqing et al. (2024). 'Hebbian learning based orthogonal projection for continual learning of spiking neural networks'. In: *arXiv preprint arXiv:2402.11984*.

Yao, Man et al. (2023). 'Spike-driven transformer'. In: *Advances in neural information processing systems* 36, pp. 64043–64058.

— (2024). ‘Spike-driven transformer’. In: *Advances in Neural Information Processing Systems* 36.

Yin, Bojian, Federico Corradi and Sander M Bohté (2021). ‘Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks’. In: *Nature Machine Intelligence* 3.10, pp. 905–913.

Zenke, Friedemann and Surya Ganguli (2018). ‘SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks’. In: *Neural Computation* 30.6, pp. 1514–1541. DOI: 10.1162/neco_a_01086.

Zenke, Friedemann and Tim P Vogels (2021). ‘The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks’. In: *Neural Computation* 33.4, pp. 899–925.

Zenke, Friedemann, Tim P. Vogels and Wulfram Gerstner (2017). ‘The temporal paradox of Hebbian learning and homeostatic plasticity’. In: *Current Opinion in Neurobiology* 43, pp. 166–176. DOI: 10.1016/j.conb.2017.03.015.

Zhao, Dongcheng et al. (2020). ‘GLSNN: A multi-layer spiking neural network based on global feedback alignment and local STDP plasticity’. In: *Frontiers in Computational Neuroscience* 14, p. 576841.

Zheng, Hanle et al. (2024). ‘Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics’. In: *Nature Communications* 15.1, p. 277.

Zhou, Chenlin et al. (2023). ‘Spikingformer: Spike-driven residual learning for transformer-based spiking neural network’. In: *arXiv preprint arXiv:2304.11954*.

Zhou, Chenlin et al. (2024a). ‘Qkformer: Hierarchical spiking transformer using qk attention’. In: *Advances in Neural Information Processing Systems*.

Zhou, Shangdong et al. (2022). ‘Spikformer: When Spiking Neural Network Meets Transformer’. In: *The Eleventh International Conference on Learning Representations*.

Zhou, Yihan et al. (2024b). ‘Direct training high-performance deep spiking neural networks: a review of theories and methods’. In: *Frontiers in Neuroscience* 18, p. 1383844. DOI: 10.3389/fnins.2024.1383844.

Zucker, Robert S. and Wade G. Regehr (2002). 'Short-Term Synaptic Plasticity'. In: *Annual Review of Neuroscience* 25, pp. 355–405. DOI: 10.1146/annurev.neuro.25.112701.142017.