Applied Soft Computing Journal 203 (2026) 116012

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Accuracy-efficiency trade-offs in spiking neural networks: A Lempel-Ziv complexity perspective on learning rules

Zofia Rudnicka, Janusz Szczepanski, Agnieszka Pregowska*

*Institute of Fundamental Technological Research, Polish Academy of Sciences, Pawinskiego 5B, Warsaw, 02-106, Poland*

## HIGHLIGHTS

- Compared unsupervised, supervised, and hybrid learning rules in spiking neural networks.
- Used Lempel-Ziv complexity to characterize learning-induced temporal structure.
- Showed that gradient-based learning achieves high accuracy at a high computational cost.
- Demonstrated favorable accuracy-efficiency trade-offs for bio-inspired learning rules.
- Highlighted that learning-rule choice depends on application accuracy and efficiency needs.

## ARTICLE INFO

*Keywords:*
Spiking neural networks
Learning rules
Spike trains
Temporal classification
Lempel–ziv complexity
Computational efficiency

## ABSTRACT

Training spiking neural networks (SNNs) remains challenging due to temporal dynamics, non-differentiability of spike events, and sparse event-driven activations. This paper studies how the choice of learning paradigm (unsupervised, supervised, and hybrid) affects classification performance and computational cost in temporal pattern recognition. Building on our earlier study Rudnicka et al. (2026) [1], we use Lempel-Ziv complexity (LZC) as a compact, analysis-oriented descriptor of spike-train temporal organization to quantify how different learning rules reshape class-conditional temporal structure. The pipeline combines a leaky integrate-and-fire (LIF) SNN with an LZC-based temporal representation analysis stage. We evaluate learning rules on synthetic sources with controlled temporal statistics (Bernoulli, two-state Markov, and Poisson spike processes) and on two-class subsets of MNIST and N-MNIST. Across datasets, gradient-based learning generally yielded the highest observed accuracy values, although several pairwise differences were not statistically significant, while simultaneously incurring substantially higher computational cost. These results highlight that selecting a learning rule should be guided by application constraints and the desired balance between separability and computational overhead.

## 1. Introduction

Backpropagation is a gradient-based learning rule commonly applied in network training [2]. Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs) are both inspired by biological neural systems, however, they differ from the computational point of view. While ANNs dominate deep learning due to their simplicity, SNNs better mimic real neurons by using spike-based communication. However, training SNNs to match ANN accuracy, especially with backpropagation-based methods [2], requires significant hardware resources and is computationally expensive [3]. As a consequence, SNN training remains a challenge due to its complex dynamics and the non-differentiable nature of spike-based activity [4,5].

Learning rules can be categorized into three main types: unsupervised learning for example, Hebbian learning [6], Symmetric Spike-Driven Synaptic Plasticity (SDSP), and Spike Timing Dependent Plasticity (STDP) [7], supervised learning such as classical backpropagation (BP) based on gradient descent [2], Spike-Timing-Dependent Backpropagation (STBP) [8], tempotron [12,13], SpikeProp [14], its direct application to LIF [15] Chronotron [16], and Remote Supervised Method (ReSuMe [17]), and hybrid learning (including Artificial-to-Spiking Neural Network conversion (ANN-SNN conversion)) [18], and Reward-based STDP [7].

A commonly used training method in SNNs is SpikeProp, which is an arithmetic-based learning algorithm conceptually similar to the

* Corresponding author.
*Email address:* aprego@ippt.pan.pl (A. Pregowska).

https://doi.org/10.1016/j.asoc.2026.116012
Received 22 February 2026; Received in revised form 31 May 2026; Accepted 19 July 2026
Available online 20 July 2026

BP algorithm in ANNs [19]. However, arithmetic-based learning rules may not be an appropriate choice for constructing biologically efficient networks. To achieve more biologically plausible learning, several alternative methods have been developed. For example, STDP is inspired by biological synaptic plasticity, where neurons strengthen or weaken connections based on temporal contiguity. However, despite its efficiency in synaptic adaptation, STDP does not match the classification performance of BP-based ANN training [2]. Moreover, spatio-temporal backpropagation (STBP) extends the traditional backpropagation algorithm to handle both spatial and temporal dependencies in spiking neural networks. This makes it possible to train deep SNNs using gradient-based optimization [20]. Other commonly used learning algorithms in SNN training are ReSuMe [17,21], and Chronotron [16]. They are effective in adapting to structured spike patterns, however, both are computationally demanding and require significant training time. An interesting SNN learning rule, namely tempotron was proposed by [12]. It is particularly effective for temporal coding, where information is embedded in the precise timing of spikes [22]. In the study [23] training is conducted using the tempotron learning rule, while optimization is achieved through STDP. This combination significantly improves processing speed and reduces computational complexity compared to conventional spiking neural network methods. In turn, [24] proposed a neuron normalization technique and an explicitly iterative neuron model, which resulted in a significant increase in the SNNs' learning rate. Recently, the Spiking Gate (SG) ResNet architecture with an attention spike decoder (ASD) has been proposed to address gradient vanishing and improve spike decoding, achieving state-of-the-art performance on several benchmarks and enabling the first direct-training hybrid SNN-ANN detector [25]. Another training strategy involves first training an ANN (typically using perceptrons) and then converting it into an SNN with an equivalent structure [18]. This conversion leverages the strengths of ANN training while allowing SNNs to benefit from energy-efficient spike-based processing. However, a major drawback is that this approach fails to capture temporal spike dynamics, as reliable frequency estimation requires a non-trivial time period. On the other hand, to optimize data efficiency, various active learning approaches have been introduced, including Bio-Inspired Active Learning (BAL), Bio-Inspired Active Learning based on Firing Rate (BAL-FR), and Bio-Inspired Active Learning utilizing Membrane Potential (BAL-M) [26]. These methods identify and select training samples by analyzing the discrepancies between empirical observations and generalization behavior, ensuring that previously unrecognized patterns are incorporated into the learning process. In turn, [8] proposes the AC2AS framework, which couples activation consistency between ANN and SNN to enable fast and memory-efficient training of spiking neural networks. Recent studies further demonstrate the growing success of deep learning and bio-inspired neural approaches in engineering and biomedical applications [9–11].

We use a previously introduced LZC-SNN analysis framework [27] to study how learning rules reshape temporal spike-train organization. In this work, LZC is treated as a compact descriptor of temporal novelty and regularity, and the downstream classification performance is interpreted as a probe of separability induced by learning rather than as evidence of a universally strong stand-alone classifier. Previous studies have highlighted the capability of bio-inspired neural networks for biosignal classification, emphasizing their high precision, sensitivity, and reliability in distinguishing physiological patterns.

To avoid ambiguity, we explicitly distinguish between the analytical and decision-related roles of LZC in the proposed framework. LZC is not treated as an independent stand-alone classifier and does not replace the SNN learning process. Instead, the SNN first transforms the input sequence into a learned spike-train representation, and LZC is then computed from this representation as a compact scalar descriptor of temporal novelty and regularity. The final decision rule uses this descriptor to probe whether the learning rule has induced separable class-conditional temporal structures. Therefore, the term "decision-relevant" means that LZC participates in the downstream decision stage as a representation of spike-train organization, while the scientific objective of the paper is to compare how different learning rules reshape this organization.

### 1.1. Motivation and scope

This study builds explicitly upon our earlier work [1], in which Lempel-Ziv complexity (LZC) was introduced as an analysis tool to investigate how different neuron models affect spiking neural network dynamics and downstream separability. In contrast to that work, the present manuscript shifts the scientific focus from neuron-model selection to the role of learning algorithms themselves.

Rather than re-introducing the coupling between LZC and spiking neural networks, we use LZC here as a compact temporal descriptor to quantify how unsupervised, supervised, and hybrid learning paradigms reshape the temporal organization of spike trains. This shift allows us to analyze learning rules not only in terms of classification accuracy, but also in terms of how they reorganize temporal variability, correlations, and computational efficiency under controlled input statistics.

While the underlying SNN architecture and part of the synthetic data generation are shared between the two studies, the scope and objectives differ substantially. Here, LZC is not used merely as a descriptive post hoc metric but as a analysis-oriented representation to analyze the effect of unsupervised, supervised, and hybrid learning rules on class-conditional temporal complexity, robustness, and computational cost.

In this study, we address the problem of comparing learning rules in spiking neural networks from the perspective of temporal organization and computational efficiency. We propose a unified experimental protocol covering unsupervised, supervised, and hybrid learning paradigms under controlled temporal statistics, complemented by two-class MNIST and N-MNIST subsets. Within this framework, we observe that learning rules with similar classification accuracy can induce substantially different temporal spike-train representations depending on the stochastic structure of the input process and the representation level. The comparative analysis further revealed that Lempel-Ziv complexity, firing-rate, and spike-count descriptors exhibit complementary behavior rather than a universally dominant performance hierarchy.

Unlike standard benchmark-oriented SNN studies that primarily compare classification accuracy across architectures or optimization schemes, the present work focuses on how learning rules reorganize temporal spike-train structure as quantified through complexity analysis. The contribution is therefore not limited to reporting predictive performance, but extends to interpreting how different forms of synaptic adaptation shape temporal representations and computational efficiency under controlled stochastic input regimes.

One methodological contribution of this work is the integration of Lempel-Ziv complexity as an analysis-oriented temporal descriptor within a unified comparative framework for spiking neural network learning rules. Unlike conventional benchmark-oriented studies that primarily compare predictive accuracy, the proposed framework analyzes how unsupervised, supervised, and hybrid learning paradigms reshape temporal spike-train organization, variability, and computational efficiency under identical experimental constraints.

This allows the comparison of learning rules not only in terms of classification outcomes, but also in terms of the temporal representations they induce. To our knowledge, relatively few studies have systematically analyzed how different SNN learning paradigms reshape temporal spike-train complexity under controlled experimental conditions.

### 1.2. Relation to prior work

Table 1 summarizes the key differences between the present study and our earlier study [1]. Although both studies employ a similar SNN architecture and synthetic spike-based datasets, the scientific questions, evaluation criteria, and conclusions are fundamentally different. The earlier study focused on the influence of neuron models on SNN

**Table 1**
Comparison between the present study and prior work [1].

| Aspect | Neuroinformatics (2026) [1] | This work |
|---|---|---|
| Primary focus | Neuron model dynamics | Learning rule dynamics |
| Role of LZC | Descriptive analysis metric | analysis-oriented temporal descriptor |
| Learning algorithms | Fixed learning setup | Unsupervised, supervised, and hybrid learning |
| Evaluation criteria | Accuracy and neuron behavior | Accuracy, LZC profiles, and computational cost |
| Scientific question | How neuron models affect SNNs | How learning rules reshape temporal structure |

dynamics, whereas the present work isolates the role of learning algorithms and their impact on temporal information structure, robustness, and computational efficiency.

To ensure transparency, we note that the network topology and part of the synthetic data generation follow [1]. However, the present study introduces a different experimental axis (learning rules), new evaluation dimensions (computational cost and complexity profiles during training), and additional datasets to assess generalization under controlled constraints.

Importantly, although both studies share a comparable LIF-based network topology and part of the synthetic data generation, the evaluated object is fundamentally different. The earlier work isolated the impact of neuron models under a fixed learning setup, whereas the present study isolates the impact of learning rules under a fixed neuron model. Consequently, the reported effects concern different mechanisms: here we focus on how learning dynamics reshape temporal correlations and variability in spike trains, as exposed by shifts in class-conditional LZC distributions and accuracy-efficiency trade-offs.

## 2. Background on learning rules in spiking neural networks

Rather than treating learning rules solely as optimization procedures, we interpret them as mechanisms that actively shape the temporal structure of spike trains. Since Lempel-Ziv complexity directly quantifies the emergence of novel temporal patterns, different learning paradigms are expected to induce distinct complexity profiles even when classification accuracy is similar. In this work, learning algorithms are therefore grouped according to how they modulate temporal correlations, variability, and information flow in spiking activity, which in turn affects complexity-based pattern recognition.

In this section, we provide the background on the learning rules considered in this study and introduce the notation used throughout the paper. We first summarize the symbols and variables employed in the remainder of the manuscript (Section 2.1). We then group the learning algorithms for spiking neural networks into three main families: unsupervised, supervised, and hybrid rules. For each family, we briefly recall the underlying learning principle, present the corresponding mathematical formulation, and highlight its main advantages and limitations in the context of temporal pattern recognition. This structured overview prepares the ground for the description of the classification task and the proposed LZC-based SNN classifier in the subsequent sections.

### 2.1. Notation

In Table 2 the basic notation is presented.

Unsupervised learning algorithms learn patterns and structures from unlabeled data without explicit supervision or target output, i.e., they provide local synaptic modifications without explicit error signals. Unsupervised learning rules in spiking neural networks adjust synaptic weights based solely on the statistics and timing of pre- and postsynaptic activity, without access to explicit target labels. They provide local, biologically plausible synaptic updates and are particularly suitable for capturing regularities in continuous, unlabeled spike streams. In this work we focus on three widely used unsupervised mechanisms: Hebbian learning, spike-timing-dependent plasticity, and symmetric spike-driven synaptic plasticity, as they form a representative set of bio-inspired rules against which we later compare supervised and hybrid approaches. From an information-theoretic perspective, correlation-driven unsupervised rules tend to suppress temporal variability by reinforcing frequently co-occurring spike patterns. As a consequence, they often lead to reduced LZC values and narrower complexity distributions, which can limit discriminative power for inputs with overlapping temporal statistics.

**Table 2**
Basic notation used in this study.

| Notation | Description |
|---|---|
| $\mathbf{x} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ | Synaptic weight vector |
| $V_m(t) \in \mathbb{R}$ | Membrane potential at time $t$ |
| $\tau_m \in \mathbb{R}^+$ | Membrane time constant |
| $I(t) \in \mathbb{R}$ | Input current at time $t$ |
| $t_f \in \mathbb{R}^+$ | Neuron firing time |
| $S_i(t) \in \{0, 1\}$ | Binary spike indicator at time $t$ |
| $t_i^k \in \mathbb{R}^+$ | Time of the $k$-th spike of neuron $i$ |
| $H(V_m(t)) \in \{0, 1\}$ | Heaviside step function |
| $\eta \in \mathbb{R}^+$ | Learning rate |
| $\tau_+, \tau_- \in \mathbb{R}^+$ | STDP time constants |
| $A \in \mathbb{R}$ | SDSP learning rate |
| $A_+, A_- \in \mathbb{R}^+$ | STDP amplitudes |
| $S_{\text{pre}}(t), S_{\text{post}}(t) \in \{0, 1\}$ | Pre- and postsynaptic binary spike trains |
| $t_{\text{pre}}, t_{\text{post}} \in \mathbb{R}^+$ | Pre- and postsynaptic spike times |
| $K(t) : \mathbb{R} \to \mathbb{R}$ | Synaptic kernel used in spike-timing rules (e.g., Tempotron) |
| $r(t) \in \mathbb{R}$ | Reward signal in reward-modulated plasticity |
| $w_{\text{SNN}}, w_{\text{ANN}} \in \mathbb{R}$ | Synaptic weight in SNN / ANN (conversion setting) |
| $\tau_{\text{syn}} \in \mathbb{R}^+$ | Synaptic time constant |
| $U(w_i) : \mathbb{R} \to \mathbb{R}$ | Uncertainty function used in active learning updates |

### 2.2. Unsupervised learning algorithms

Hebbian Learning Rule strengthens synapses when both neurons fire together [6]

$$\Delta w_i = \eta(S_{pre}(t)S_{post}(t)). \tag{1}$$

A major drawback of Hebbian learning is its decreasing efficiency as the number of hidden layers increases [28]. Although it remains competitive with up to four layers, its performance diminishes in deeper networks, limiting its scalability for complex architectures.

#### 2.2.1. Spike-timing-dependent plasticity

Spike-timing-dependent plasticity (STDP) is a biologically inspired local learning rule in which synaptic updates depend on the relative timing between pre- and postsynaptic spikes.

$$\Delta w_i = \begin{cases} A_+ e^{-\frac{(t_{post}-t_{pre})}{\tau_+}} & t_{post} > t_{pre} \\ -A_- e^{-\frac{(t_{post}-t_{pre})}{\tau_-}} & t_{pre} > t_{post}. \end{cases} \tag{2}$$

It is a form of Hebbian learning (1). It is essential for modeling the temporal dynamics of biological neural systems. Unlike traditional learning methods, its event-driven nature enables unsupervised learning, allowing neural networks to adapt autonomously to temporal patterns in input data. This makes it valuable for processing spatio-temporal signals, such as sensory data. In addition, its computational demands can be high in large-scale simulations.

#### 2.2.2. Symmetric spike-driven spike-based plasticity

SDSP is a variant of STDP, however, it features symmetric weight updates

$$\Delta w_i = A(S_{pre}(t) - S_{post}(t)). \tag{3}$$

It is a biologically plausible learning mechanism that mirrors neural dynamics by focusing on individual spike events. This event-driven

approach enables rapid adaptability, making it valuable for real-time learning. However, its sensitivity to single spikes poses challenges in noisy environments, requiring advanced filtering techniques.

### 2.3. Supervised learning algorithms

Supervised learning maps input data to output labels based on a labeled dataset. It relies on error-based updates, which adjust synaptic weights based on an explicit error or teacher signal that encodes the desired network output. In contrast to unsupervised rules, they are designed to optimize task-specific performance, such as classification accuracy, at the cost of reduced biological plausibility and typically higher computational complexity. Here we review representative supervised learning rules for SNNs, including gradient-based methods (backpropagation and STBP) and spike-timing-based algorithms (tempotron, SpikeProp, Chronotron, and ReSuMe), all of which are later evaluated within the same classification framework. Error-driven learning rules explicitly optimize class separability, which may also influence the organization of class-conditional complexity distributions observed in the resulting spike-train representations. This effect highlights that supervised learning influences not only decision boundaries but also the internal temporal organization of spike trains.

#### 2.3.1. Backpropagation algorithm

Backpropagation is a gradient-based learning rule commonly applied in network training [2].

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \tag{4}$$

where $E$ is the error function. The weight update follows a chain rule

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial V_m} \cdot \frac{\partial V_m}{\partial w_i}. \tag{5}$$

#### 2.3.2. Spike-timing-dependent backpropagation

Spike-Timing-Dependent Backpropagation extends the conventional gradient-based learning rule of backpropagation to spiking neural networks, incorporating spike-timing-dependent plasticity principles to optimize synaptic weights [8]. The weight update rule is given by Eq. (4). In STBP, the gradient computation follows the chain rule:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial t_{\text{spike}}} \cdot \frac{\partial t_{\text{spike}}}{\partial V_m} \cdot \frac{\partial V_m}{\partial w_i}. \tag{6}$$

Unlike traditional backpropagation, STBP accounts for the temporal dynamics of spikes, leveraging precise spike timing to adjust synaptic strengths. This biologically inspired approach enables more efficient credit assignment in spiking networks while maintaining compatibility with gradient-based optimization methods.

#### 2.3.3. Tempotron learning rule

The Tempotron learning rule adjusts synaptic weights to classify temporal spike patterns [12,29]

$$\Delta w_i = \eta \sum_{t \in T_{pre}} K(t_f - t) \tag{7}$$

The tempotron learning principle is a biologically inspired approach that refines synaptic weights based on the precise timing of incoming spikes rather than just their frequency. Unlike conventional neural models that prioritize weight adjustments or network structures, the tempotron emphasizes the importance of timing, showing that the "when" of a neural event can be as crucial as the "where" or "how often".

#### 2.3.4. SpikeProp algorithm

SpikeProp applies gradient descent to minimize spike-timing errors [14]:

$$\frac{\partial E}{\partial w_i} = \sum_{t_{post}} \frac{\partial E}{\partial t_i^{actual}} \cdot \frac{\partial t_i^{actual}}{\partial w_i}. \tag{8}$$

SpikeProp extends backpropagation to spike-based networks, enabling gradient-based optimization. It is computationally expensive due to temporal dynamics. Additionally, it requires careful handling of non-differentiability in spike generation.

#### 2.3.5. Chronotron learning rule

The Chronotron minimizes the discrepancy between actual and target spike times [16]

$$E = \sum_i \sum_k \frac{(t_i^{actual} - t_i^{target})^2}{2} \tag{9}$$

$$\Delta w_{ij} = -\eta \sum_k (t_i^{actual} - t_i^{target}) \frac{\partial t_i^{actual}}{\partial w_{ij}} \tag{10}$$

A key advantage of this approach is its consistent coding for both inputs and outputs, improving the efficiency of SNNs. However, the granularity of the method can significantly increase computational demands, requiring precise parameter tuning and high-quality temporal data.

#### 2.3.6. Remote supervised method

ReSuMe combines STDP with supervised learning using a teacher spike train [17]

$$\Delta w_{ij} = \eta \left[ S_{pre}(t) * S_{teach}(t) - S_{pre}(t) * S_{post}(t) \right] \tag{11}$$

ReSuMe enables supervised training while maintaining biological plausibility [30].

### 2.4. Hybrid learning algorithms

Hybrid algorithms integrate elements of both supervised and unsupervised learning. These schemes combine error-driven optimization with local synaptic plasticity or bridge the gap between rate-based ANNs and event-based SNNs. Such methods aim to leverage the robustness and training efficiency of conventional deep learning while retaining the energy efficiency and temporal precision of spike-based computation. In this study we consider three representative hybrid strategies: ANN-to-SNN conversion, reward-modulated STDP, and bio-inspired active learning. Hybrid and reward-modulated learning strategies combine local plasticity with task-driven adaptation mechanisms, which may help preserve temporal variability under stochastic inputs. This property can influence the stability of downstream complexity-based representations across noisy temporal regimes.

#### 2.4.1. Artificial-to-spiking neural network conversion

ANN-SNN conversion maps pre-trained ANN weights to SNN synapses [18]

$$w_{SNN} = \frac{w_{ANN}}{\tau_{syn}}. \tag{12}$$

Converting ANNs to SNNs allows the use of existing training methodologies while leveraging SNNs' energy efficiency and biological realism [31]. This conversion maps ANN activation values to the firing rates of SNN neurons, requiring precise threshold tuning to maintain accuracy. Improper thresholds can lead to excessive or insufficient spiking, which affects performance.

#### 2.4.2. Reward-based STDP

Reward-based STDP modifies STDP with reinforcement learning

$$\Delta w_{ij} = \eta r(t) \left[ S_{pre}(t) * S_{post}(t) \right]. \tag{13}$$

#### 2.4.3. Bio-inspired active learning

Active learning updates weights based on uncertainty [32]

$$\Delta w_{ij} = \eta \cdot U(w_i) \cdot \mathbb{E}\left[I(S_{post}; S_{pre})\right]. \tag{14}$$

Integrates biological learning mechanisms with active learning strategies to improve training efficiency [32]. Unlike traditional supervised learning, which relies on large labeled datasets, BAL actively selects the most informative data points for labeling, reducing data requirements while maintaining high performance.

Fig. 1 summarizes the conceptual structure of the proposed LZC-SNN framework. The framework consists of three sequential stages: learning-rule-driven transformation of temporal input sequences into spike-train representations, extraction of Lempel-Ziv complexity as a compact descriptor of temporal organization, and downstream complexity-based decision analysis. This formulation emphasizes that LZC is not used as an independent trainable classifier, but rather as an analysis-oriented temporal representation used to analyze how different learning paradigms reshape spike-train structure.

## 3. Temporal classification framework

### 3.1. Network architecture

The SNN neural network under consideration consists of three layers: input, hidden, and output, each containing $n$ neurons; see Fig. 2. It is based on the Leaky Integrate-and-Fire (LIF) biophysical neuron model which is a fundamental component of SNNs used to simulate the subthreshold behavior of the membrane potential of a neuron $U(t) \in \mathbb{R}$ [33]. Given an input vector $\mathbf{x}(t) \in \mathbb{R}^n$, synaptic weights $\mathbf{w} \in \mathbb{R}^n$, and bias $b \in \mathbb{R}$, the net input current is defined as:

$$I(t) = \mathbf{w}^T \mathbf{x}(t) + b = z(t), \tag{15}$$

where $z(t)$ denotes the weighted synaptic drive at time $t$. The membrane voltage evolves according to:

$$\tau_m \frac{dU(t)}{dt} = -U(t) + z(t), \tag{16}$$

and emits a spike when $U(t) \geq U_{\text{th}}$, after which the voltage is reset to a resting value $U_r$. This model captures the temporal integration and passive leakage of membrane charge in a computationally efficient form.

### 3.2. LZC-based decision mechanism

The proposed framework can be interpreted as a sequential three-stage pipeline. First, the selected learning rule transforms the input temporal sequence into a spike-based representation within the SNN. Second, normalized Lempel–Ziv complexity is computed from the resulting spike train in order to quantify temporal regularity and novelty. Third, class-conditional complexity distributions are analyzed in scalar complexity space to evaluate whether the considered learning paradigm induces separable temporal organization.

For clarity, we formally summarize the decision pipeline used in the proposed framework. Given an input temporal sequence $\mathbf{x}$, the SNN parameterized by the learning rule $\mathcal{L}$ produces a spike-train representation

$$\mathbf{s} = F_{\mathcal{L}}(\mathbf{x}), \tag{17}$$

where $F_{\mathcal{L}}$ denotes the spike-based temporal transformation induced by the selected learning algorithm.

Next, normalized Lempel–Ziv complexity is computed from the resulting spike train:

$$c_\alpha(\mathbf{s}) = \frac{C_\alpha(\mathbf{s})}{n} \log_\alpha(n). \tag{18}$$

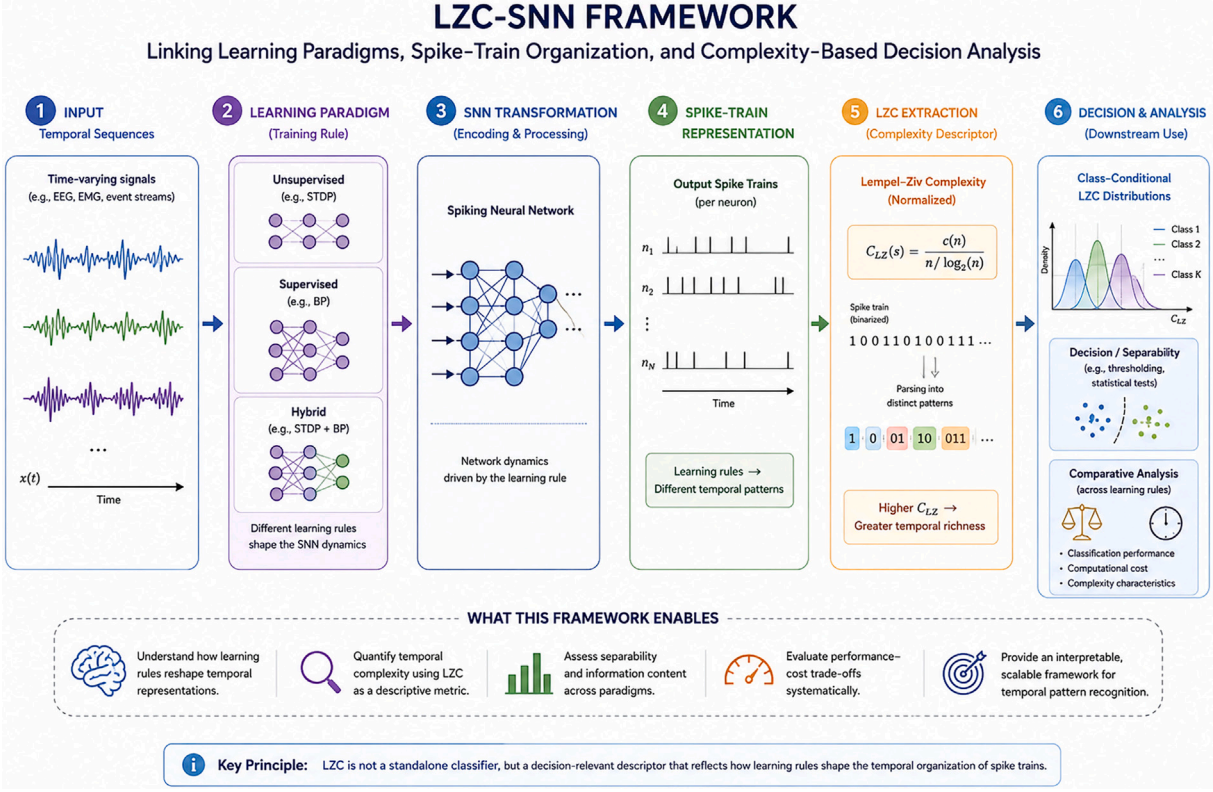


**Fig. 1.** Conceptual structure of the proposed LZC-SNN framework. Input temporal sequences are transformed by an SNN trained using different learning paradigms. The resulting spike-train representations are analyzed using normalized Lempel-Ziv complexity, and downstream decisions are obtained from scalar complexity-based representations derived from spike-train organization. The framework is designed to evaluate how learning rules reshape temporal organization rather than to optimize a stand-alone discriminative classifier.

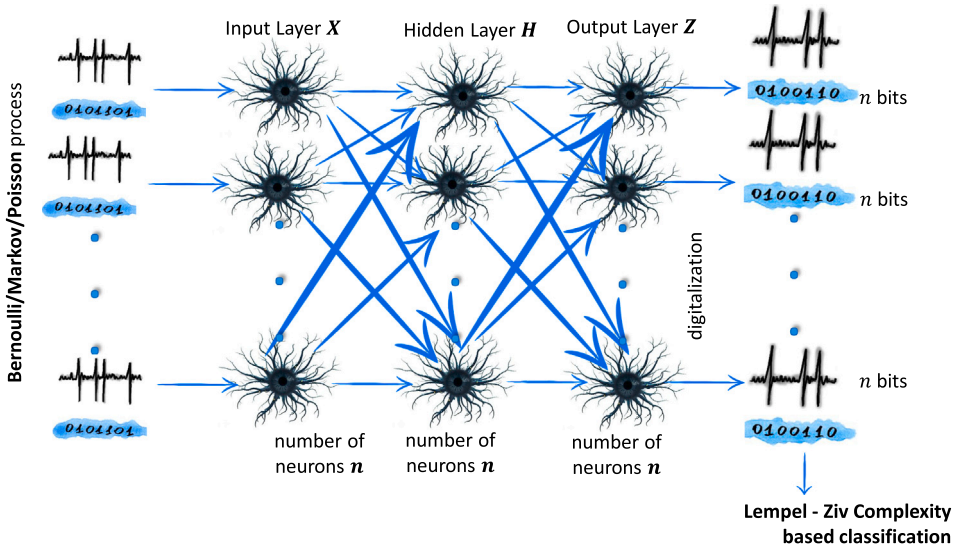


**Fig. 2.** The architecture of the considered neural network.

The resulting scalar complexity representation is subsequently compared with class-conditional reference distributions estimated during training in order to evaluate the separability induced by the considered learning rule. Formally,

$$\hat{y} = \arg\min_{k \in C} d\left(c_\alpha(\mathbf{s}), \mu_k\right), \tag{19}$$

where $\mu_k$ denotes the representative complexity statistic associated with class $k$, and $d(\cdot,\cdot)$ is a distance measure in the scalar complexity space.

Consequently, downstream classification performance is interpreted primarily as an observable consequence of learning-induced temporal organization rather than as evidence that LZC itself constitutes a standalone optimized classifier.

For all BP-based experiments, the architecture, optimization schedule, and training duration were kept fixed across datasets in order to isolate the influence of the learning paradigm itself rather than architecture-specific optimization effects. No additional architectural modifications, normalization layers, or dataset-specific tuning strategies were introduced for the BP baseline.

### 3.3. Comparative performance analysis

We considered different sets of neuron parameters and the presented results include the most efficient set of parameters for each learning algorithm. The threshold took values 1.00, 0.50, 0.40, 0.30, 0.20, 0.10, and 0.05. The decline was 0.100, 0.05, 0.03, 0.01, and the learning rates were 0.0500, 0.0100, 0.0010, 0.0098, and 0.0001. We considered neural networks comprising 16, 32, 64, 128, 256, 512, and 1024 LIF neurons per layer. The best of the results obtained are presented in Tables 5, 6 and 9. A study on the effect of neuron count on transmission efficiency in the case of the Levy-Baxter neuron model has been reported in [34].

The phrase "most efficient set of parameters" refers to the best-performing configuration selected within a fixed and identical search space for all compared algorithms. Specifically, each learning rule was evaluated over the same predefined grid of firing thresholds, membrane decay parameters, learning rates, and layer sizes. The selected configuration was the one that maximized validation accuracy; when two configurations achieved the same validation accuracy, the configuration with lower computation time was selected. This procedure was applied uniformly to unsupervised, supervised, and hybrid learning rules to avoid favoring any particular algorithmic family.

To ensure methodological fairness, all learning rules were evaluated under the same experimental protocol, including identical input dimensionality, training/test splits, sequence length, epoch count, and parameter-search ranges. No algorithm-specific architectural optimization or task-specific tuning was introduced beyond the predefined hyperparameter grid. Consequently, differences in performance are interpreted as arising primarily from the learning dynamics themselves rather than from unequal optimization conditions or architecture modifications.

To further ensure fairness, all evaluated learning rules were subjected to the same maximum number of training epochs, identical stopping conditions, and equivalent hyperparameter-search budgets. No additional manual tuning iterations were introduced for individual algorithms beyond the predefined experimental protocol.

The network processes binary sequences of length 1024, which are transformed into $n$-bit spike trains and subsequently converted back into binary sequences. Classification decisions are derived from scalar complexity-based representations computed from spike trains, enabling analysis of how learning rules reshape temporal organization under controlled stochastic conditions, a well-established metric for evaluating the informational content of a sequence. A binary sequence $\mathbf{x}_n^1 := [x_1, x_2, \ldots, x_n]$, where each $x_i \in \{0,1\}$ the LZ complexity $C_\alpha(\mathbf{x}_n^1)$ represents the count of distinct substrings that appear sequentially in $\mathbf{x}_n^1$. The normalized complexity, which assesses the rate at which novel patterns emerge, is defined as:

$$c_\alpha(\mathbf{x}_n^1) = \frac{C_\alpha(\mathbf{x}_n^1)}{n} \log_\alpha n, \tag{20}$$

where $\alpha = 2$ for binary sequences. For purely random sequences, $c_\alpha(\mathbf{x}_n^1)$ converges to 1, while for deterministic sequences, it approaches 0. This measure effectively approximates the entropy of ergodic stochastic processes, making it a valuable tool to evaluate randomness and structure within the output of the neural network [27,35]. For completeness, we performed our analysis for information sources for which the subsequent symbols are increasingly correlated [36].

## 4. Input datasets

In this study, we employed synthetic binary datasets designed to represent different types of temporal dependencies and levels of stochasticity. We consider three types of datasets that consist of Bernoulli sequences (uncorrelated), two-state Markov, and Poisson processes [37].

Poisson spike trains have been widely used to model neural activity since early studies, owing to their biologically realistic temporal dynamics and the characteristic proportionality between the mean and variance of neural responses, which gives rise to variable interspike intervals [38]. Experimental data have indicated this Poisson property [37,38]. The binary sequence of each process is converted into inputs to LIF models with specified $\Delta t = 1ms$ [38]. Performance was evaluated using classification accuracy, mean squared error (MSE), and the $R^2$ score.

Bernoulli sequences are generated from independent Bernoulli processes, where each binary element $b_i$ in a given sequence is drawn independently with probability p. For our experiments, we generate two sets of binary sequences $B_1$ and $B_2$, each sequence of length 1024, using different values of $p$. The independence of events allows us to assess the system's performance under random binary outcomes with varying probabilities.

Markov sequences are generated from a first-order Markov process, where the probability of each element $s_i$ in a given sequence depends on the previous state $s_{i-1}$. For our experiments, two sets of binary sequences of length 1024, $M_1 = \{m_{1,k}\}, k = 1, 2, \ldots 1000$ and $M_2 = \{m_{2,k}\}, k = 1, 2, \ldots 1000$ are generated with transition probabilities defining the dependency structure. We explore different transition probability configurations to evaluate the impact of state dependencies on classification accuracy.

Poisson sequences are generated from a Poisson process, where events occur independently at a constant average rate $\lambda$. For each Poisson process, spike trains are generated with rate parameters $\lambda_x$ and $\lambda_y$. These sequences are tested under various rate configurations to observe the effects of spike timing variability on system behavior and classification performance. This variety of data we employed allows us to systematically assess the strengths and limitations of classical and bio-inspired learning algorithms under different input regimes. Bernoulli sequences test the ability to classify memoryless inputs, Markov sequences require short-term memory handling, and Poisson sequences emulate real-world neural signal variability. Using synthetic data also enables precise control over statistical properties and eliminates noise or artifacts that can confound real-world datasets. This makes it possible to isolate and compare the intrinsic capabilities of the learning rules. Performance was evaluated using classification accuracy, mean squared error (MSE), and $R^2$ score.

This synthetic design allows precise control over the temporal structure and stochasticity of inputs, which is critical when evaluating biologically inspired learning rules under known statistical regimes. Unlike benchmark datasets, our generative setup isolates key dynamics relevant to spiking computation.

In addition to the synthetic temporal datasets, we also evaluated the proposed approach on binarized versions of two widely used benchmark datasets: MNIST and N-MNIST, restricted to the digits 0 and 1. The MNIST01 subset consists of grayscale handwritten digits from the original MNIST collection, filtered to include only classes 0 and 1, resulting in a clean two-class pattern-recognition task. Each image was normalized and flattened into a 1024-dimensional input vector ($32 \times 32$ cropping/reshaping), consistent with the dimensionality used in our spike-based models. This reduced task serves as a controlled setting to isolate the behavior of the SNN learning rules without confounding factors such as multi-class overlap or high intra-class variability. The N-MNIST01 subset follows the same class restriction but is derived from the neuromorphic N-MNIST dataset, where event-based recordings of moving MNIST digits are captured using a Dynamic Vision Sensor (DVS). Events corresponding to digits 0 and 1 were retained, and the resulting spike streams were temporally discretized into the same fixed-length input representation used for all other experiments. This allows direct comparison between conventional image-based inputs (MNIST01) and event-driven neuromorphic inputs (N-MNIST01) under identical network architectures and training procedures. Input characteristics for the MNIST01 and N-MNIST01 subsets used in our experiments are shown in Table 3. Both MNIST01 and N-MNIST01 contain only the digits 0 and 1, enabling a controlled binary classification setting. For M-NIST01, images are normalized and converted to a fixed 1024-dimensional vector. For N-MNIST01, DVS events corresponding to digits 0 and 1 are aggregated into $32 \times 32$ bins, producing spike-based representations with identical dimensionality, allowing direct comparison across static and event-driven modalities (Table 4).

**Table 3**
Input characteristics for the MNIST01 and N-MNIST01 subsets used in our experiments.

| Dataset | Samples (train/test) | Input type | Dim. |
|---|---|---|---|
| MNIST01 | 12,665 / 2115 | Static grayscale images | 1024 |
| N-MNIST01 | 12,665 / 2115 | Event-based spike streams | 1024 |

**Table 4**
Additional neuromorphic and image-based datasets used for supplementary evaluation.

| Dataset | Modality | Classes | Representation |
|---|---|---|---|
| FashionMNIST01 | Static grayscale images | 0/1 subset | Binary vectorized input |
| Gesture01 | Event-based vision | Two gesture classes | Spike-event representation |
| SHD01 | Auditory neuromorphic | Two spoken digits/classes | Spike-train representation |

The restriction to a binary (0/1) setting is intentional and serves to isolate the effect of learning rules on temporal representations under controlled conditions, rather than to compete with multi-class benchmark performance. The binary setting additionally reduces confounding effects associated with multi-class overlap and enables a more controlled analysis of temporal representation dynamics induced by learning rules. Since the primary objective of this study is not state-of-the-art benchmark competition but rather interpretability of learning-induced temporal organization, the reduced two-class scenario provides a suitable compromise between task complexity and analytical transparency.

The reduced binary setting was selected intentionally to minimize confounding effects associated with complex multi-class overlap and to enable controlled interpretation of learning-induced temporal organization. The objective of the study is therefore a methodological analysis of learning-rule dynamics rather than state-of-the-art large-scale benchmark optimization.

In addition to MNIST01 and N-MNIST01, supplementary experiments were also conducted on reduced subsets of Fashion-MNIST, DVS Gesture, and SHD (Spiking Heidelberg Digits) datasets in order to evaluate the robustness of the considered learning rules across static image-based, event-driven visual, and auditory neuromorphic modalities. For consistency with the main experimental protocol, all datasets were converted to binary or spike-based representations with fixed-dimensional inputs compatible with the considered SNN architecture. These additional datasets were used primarily to verify whether the observed trends regarding temporal complexity organization and efficiency-accuracy trade-offs remain consistent across heterogeneous sensory domains.

Due to computational cost considerations, the full experimental grid covering all unsupervised, supervised, and hybrid learning rules was evaluated primarily on controlled synthetic datasets (Bernoulli, Markov, and Poisson processes). Additional experiments on MNIST01, FashionMNIST01, N-MNIST01, Gesture01, and SHD01 datasets were conducted for representative supervised learning rules in order to assess robustness across static image-based, event-driven visual, and auditory neuromorphic modalities under comparable temporal classification conditions.

## 5. Results and discussion

In contrast to conventional evaluations that focus primarily on classification accuracy, our analysis emphasizes how different learning

paradigms reshape the temporal information structure of spike trains. In addition to classification accuracy, our analysis examines how different learning paradigms reshape the temporal information structure of spike trains.

To reduce sensitivity to stochastic initialization and spike-generation variability, each experiment was repeated over 10 independent random seeds using identical training and evaluation conditions. The values reported in Tables 5–9 correspond to mean performance across repeated runs. Preliminary analysis indicated relatively low variance for Bernoulli and Markov datasets, whereas larger fluctuations were observed for Poisson-distributed inputs due to their inherently stochastic temporal structure.

In addition to mean performance values, supplementary aggregate statistics were computed across repeated runs for all evaluated datasets and learning rules, including standard deviations of accuracy, latency, runtime, and related efficiency metrics. Preliminary analysis indicates relatively stable behavior for Bernoulli and Markov datasets, whereas larger variability was consistently observed for Poisson-distributed inputs due to their inherently stochastic temporal structure and irregular spike timing.

We consider classical and bio-inspired learning algorithms in SNNs. Our prior work [1] introduced LZC as an analysis tool for assessing the impact of neuron models on SNN behavior. Here, we move beyond descriptive use and evaluate how different learning paradigms reshape class-conditional complexity profiles when LZC is used as an analysis-oriented temporal descriptor within the classification pipeline. The focus is on characterizing learning-rule-induced changes in temporal organization and relating them to accuracy-efficiency trade-offs under controlled input statistics.

Table 5 presents the influence of the unsupervised learning algorithms such as Hebbian, STDP, and SDSP on the performance of SNNs. All computations were performed for 10 epochs on an Intel(R) Core(TM) i7-14700F CPU @ 2.10 GHz. A general finding of our study is the pronounced dependence of both classification accuracy and computational overhead on the choice of learning algorithm and the underlying characteristics of the input data. Observed classification accuracies ranged from 89.50% to 100.00% across the evaluated methods, whereas computational times exhibited substantial variation, ranging from approximately 7 s to over 48,000 s.

Table 5 presents the influence of the unsupervised learning algorithms such as Hebbian, STDP, and SDSP on the performance of SNNs. The results obtained for the Hebbian algorithm demonstrate that, within the classical framework, high classification accuracy was achieved for the Bernoulli and Markov processes, with values of 99.00% and 95.79%, respectively. These results were accompanied by relatively long computation times of 52.93 s and 74.66 s. Furthermore, MSE values of 0.0011 and 0.00422, suggest minimal prediction errors. The coefficient of determination $R^2$ scores of 0.9600 and 0.8313 further indicate a strong agreement between predicted and actual values. For the Poisson process, a classification accuracy of 90.00% was observed, with an $R^2$ score of 0.9074. The corresponding MSE value was 0.1000, while the $R^2$ score of 0.5998 indicates a moderate correlation between the predictions and the actual values. Notably, the accuracy for the Poisson process is achieved with a twice smaller number of neurons in the layers (for Bernoulli and Markov processes, we achieve similar accuracy with the same number of neurons in the layers). An increase in the number of neurons causes a decrease in accuracy below 50.00%, contrary to the case of Bernoulli and Markov processes, where accuracy increases with an increasing number of neurons in layers. Thus, the worse classification results for the Poisson process may be related to the fact that events occur randomly in time, making it difficult to generalize Hebbian learning effectively. The relatively low computation time indicates that the Hebbian learning algorithm efficiently processes structured data. However, taking into account the nature of Hebbian learning it is effective when there are strong, predictable relationships between inputs and outputs, like of the case in Bernoulli or Markov processes.

The introduction of other bioinspired learning algorithms, namely STDP and SDSP provides a similar level of accuracy, in a similar computational time, with the exception of the SDSP algorithm for Markov processes, where the computational time is an order of magnitude longer. However, the lower MSE for Bernoulli and Markov processes indicates that STDP fine-tunes synaptic weights more effectively than Hebbian learning, reducing classification errors.

Unsupervised learning algorithms help achieve high accuracy with far fewer neurons in layers for uncorrelated data.

Table 6 presents the influence of the supervised learning algorithms such as backpropagation, tempotron, SpikeProp, Chronotron and ReSuMe on the performance of SNNs. The BP algorithm tended to achieve the highest predictive accuracy under the evaluated experimental conditions for Bernoulli, Markov, and Poisson processes, as reflected by very low MSE values and high $R^2$ scores. However, these improvements were accompanied by substantially increased computational cost, making BP the most resource-intensive supervised method considered in this study. Similar behavior was observed for the STBP algorithm, which also achieved high predictive performance but required even longer computation times due to the additional temporal gradient computations associated with spike timing dynamics.

Although BP achieved the highest predictive performance in the evaluated setting in terms of predictive performance, its high computational demands render it impractical for real-time or large-scale applications where efficiency is crucial. This trend also holds for the STBP algorithm, but with a significantly greater time expenditure than BP. In turn, in [29], the authors state that they studied Bernoulli sources; however, in their network, the input layer consisted of LIF neurons, while the remaining layers comprised perceptrons trained using the tempotron algorithm. They did not present results for these datasets. We achieve high classification accuracy, reaching 99.00% for the Bernoulli process and 100.00% for the Markov process, while is the case of the Poisson process the level of accuracy in lower, namely 97.50%. In the case of the tempotron, for Bernoulli and Poisson the calculations are very fast, taking about 7s. For Markov the time is eight times longer, while classification depends not just on the current state but also on prior states, the model

**Table 5**
Influence of unsupervised learning algorithms on SNN performance.

| Type | Subtype | Dataset | Bio- | Ep. | $n$ | Time [s] | Acc. [%] | MSE | $R^2$ |
|---|---|---|---|---|---|---|---|---|---|
| Unsupervised | Hebbian | Bernoulli | Yes | 10 | 128 | 51.93 | 99.00 | 0.0100 | 0.9600 |
| | | Markov | Yes | 10 | 128 | 74.66 | 95.78 | 0.0422 | 0.8313 |
| | | Poisson | Yes | 10 | 64 | 17.17 | 90.00 | 0.1000 | 0.5998 |
| | STDP | Bernoulli | Yes | 10 | 64 | 62.08 | 99.50 | 0.0050 | 0.9800 |
| | | Markov | Yes | 10 | 128 | 513.73 | 98.00 | 0.0200 | 0.9200 |
| | | Poisson | Yes | 10 | 64 | 63.36 | 89.50 | 0.1050 | 0.5798 |
| | SDSP | Bernoulli | Yes | 10 | 128 | 43.20 | 98.00 | 0.0200 | 0.9200 |
| | | Markov | Yes | 10 | 128 | 46.10 | 93.87 | 0.0613 | 0.7549 |
| | | Poisson | Yes | 10 | 64 | 17.83 | 92.50 | 0.0750 | 0.6999 |

**Table 6**
Influence of supervised learning algorithms on SNN performance.

| Type | Subtype | Dataset | Bio- | Ep. | $n$ | Time [s] | Acc. [%] | MSE | $R^2$ |
|---|---|---|---|---|---|---|---|---|---|
| Supervised | BP | Bernoulli | No | 10 | 128 | 866.44 | 99.00 | 0.0100 | 0.9600 |
| | | Markov | No | 10 | 256 | 6211.62 | 100.00 | 0.0000 | 1.0000 |
| | | Poisson | No | 10 | 512 | 12,916.37 | 97.50 | 0.0250 | 0.9000 |
| | | MNIST | No | 10 | 128 | 207.91 | 99.05 | 0.0094 | 0.9619 |
| | | Fashion MNIST | No | 10 | 128 | 1106.38 | 95.75 | 0.0425 | 0.8300 |
| | | Fashion MNIST | No | 10 | 128 | 203.76 | 95.64 | 0.0435 | 0.8257 |
| | | N-MNIST | No | 10 | 128 | 1508.40 | 99.23 | 0.0072 | 0.9715 |
| | | DVS Gesture | No | 10 | 128 | 255.77 | 100.00 | 0.0000 | 1.0000 |
| | | SHD | No | 10 | 128 | 384.71 | 95.26 | 0.0474 | 0.8103 |
| | STBP | Bernoulli | No | 10 | 128 | 2261.43 | 100.00 | 0.0000 | 1.0000 |
| | | Markov | No | 10 | 512 | 48,258.21 | 100.00 | 0.0000 | 1.0000 |
| | | Poisson | No | 10 | 512 | 30,093.49 | 96.50 | 0.0350 | 0.8599 |
| | | MNIST | No | 10 | 128 | 885.10 | 99.83 | 0.0140 | 0.9946 |
| | | Fashion MNIST | No | 10 | 128 | 1106.38 | 95.75 | 0.0425 | 0.9558 |
| | | N-MNIST | No | 10 | 128 | 1061.43 | 99.63 | 0.0041 | 0.9837 |
| | | DVS Gesture | No | 10 | 128 | 1401.89 | 97.92 | 0.0208 | 0.9167 |
| | | SHD | No | 10 | 128 | 2229.57 | 89.57 | 0.1043 | 0.5270 |
| | Tempotron | Bernoulli | Yes | 10 | 128 | 7.00 | 96.00 | 0.0400 | 0.8399 |
| | | Markov | Yes | 10 | 128 | 40.50 | 90.59 | 0.0941 | 0.6234 |
| | | Poisson | Yes | 10 | 128 | 7.23 | 89.50 | 0.1050 | 0.5798 |
| | | MNIST | No | 10 | 128 | 32.20 | 87.55 | 0.1245 | 0.8914 |
| | | Fashion MNIST | No | 10 | 128 | 38.54 | 100.00 | 0.0000 | 1.0000 |
| | | N-MNIST | No | 10 | 512 | 593.90 | 98.41 | 0.1776 | 0.2865 |
| | | DVS Gesture | No | 10 | 128 | 119.15 | 72.95 | 0.2710 | 0.7350 |
| | | SHD | No | 10 | 128 | 68.71 | 72.04 | 0.2799 | 0.1191 |
| | SpikeProp | Bernoulli | Yes | 10 | 128 | 64.28 | 100.00 | 0.0000 | 1.0000 |
| | | Markov | Yes | 10 | 64 | 47.00 | 92.34 | 0.0766 | 0.6935 |
| | | Poisson | Yes | 10 | 32 | 20.15 | 91.50 | 0.0850 | 0.6599 |
| | | MNIST | No | 10 | 128 | 68.80 | 99.12 | 0.0129 | 0.9484 |
| | | Fashion MNIST | No | 10 | 128 | 92.49 | 93.68 | 0.0632 | 0.7471 |
| | | N-MNIST | No | 10 | 128 | 383.30 | 98.38 | 0.0162 | 0.9348 |
| | | DVS Gesture | No | 10 | 128 | 166.38 | 100.00 | 0.0000 | 1.0000 |
| | | SHD | No | 10 | 128 | 99.37 | 91.94 | 0.0806 | 0.6775 |
| | Chronotron | Bernoulli | Yes | 10 | 32 | 18.61 | 98.50 | 0.0150 | 0.9400 |
| | | Markov | Yes | 10 | 32 | 46.30 | 92.34 | 0.0175 | 0.9300 |
| | | Poisson | Yes | 10 | 32 | 113.98 | 90.50 | 0.0950 | 0.6198 |
| | | MNIST | No | 10 | 256 | 16,329.10 | 76.18 | 0.2382 | 0.0432 |
| | | Fashion MNIST | No | 10 | 128 | 2394.12 | 75.25 | 0.2475 | 0.0100 |
| | | N-MNIST | No | 10 | 256 | 10,299.07 | 75.41 | 0.1373 | 0.4482 |
| | | DVS Gesture | No | 10 | 128 | 197.17 | 87.50 | 0.1250 | 0.8980 |
| | | SHD | No | 10 | 64 | 100.18 | 81.99 | 0.1801 | 0.2792 |
| | ReSuMe | Bernoulli | Yes | 10 | 128 | 68.55 | 92.50 | 0.0750 | 0.6999 |
| | | Markov | Yes | 10 | 64 | 46.05 | 91.60 | 0.0840 | 0.6642 |
| | | Poisson | Yes | 10 | 64 | 21.45 | 91.50 | 0.0850 | 0.6590 |
| | | MNIST | No | 10 | 128 | 56.04 | 98.55 | 0.0145 | 0.9416 |
| | | Fashion MNIST | No | 10 | 128 | 63.34 | 95.07 | 0.0493 | 0.8029 |
| | | N-MNIST | No | 10 | 128 | 438.12 | 98.68 | 0.0132 | 0.9470 |
| | | DVS Gesture | No | 10 | 128 | 171.14 | 100.00 | 0.0000 | 1.0000 |
| | | SHD | No | 10 | 128 | 101.25 | 91.94 | 0.0806 | 0.6775 |

must process sequences over multiple time steps, significantly increasing computational demands. Since the Poisson process does not have explicit dependencies like Markov sequences, computations remain efficient.

Also, the computation for the tempotron requires fewer layers of neurons than BP and STBP with only a slight loss of precision. In turn, the SpikeProp algorithm requires fewer neurons in the layers, which affects the computation speed for Markov and Poisson processes. This number decreases for all datasets when we apply the Chronotron learning algorithm, but the computation time is longer only in the case of the Poisson algorithm. Moreover, unlike classical backpropagation, which requires full gradient updates over all layers, SpikeProp optimizes synaptic weights through event-driven updates, significantly reducing computational complexity in comparison to BP and STBP.

The ReSuMe algorithm exhibits the lowest average classification accuracy among the evaluated supervised learning algorithms. However, it remains computationally efficient. Consequently, this results in less precise weight corrections, leading to a higher classification error.

Supervised biologically inspired learning algorithms help achieve high accuracies with far fewer neurons in layers for uncorrelated data than supervised algorithms, non-biologically inspired learning algorithms, and unsupervised algorithms.

Table 9 presents the influence of hybrid learning algorithms such as ANN-SNN conversion, STDP based on rewards, and BAL on SNN performance. For ANN-SNN conversion, the LZC-based classifier achieves high classification accuracy across all datasets. However, performance for Poisson-distributed inputs remains lower than for Bernoulli and Markov sources due to higher temporal variability. We achieve comparable results with the reward-based and BAL learning algorithms. However in the case of reward based, for Markov and Poisson processes the network requires more neurons in layers than ANN-SNN conversion, but the computation time is shorter. For Bernoulli the network requires half as many neurons in layers, but the computation time is five times shorter. BAL has six times longer computation time for a Poisson process than the reward based algorithm. Unlike standard Hebbian learning or STDP, BAL seems to handle stochastic volatility more efficiently.

**Table 7**
Wilcoxon signed-rank test for synthetic stochastic datasets (Bernoulli, Markov, Poisson). All comparisons yield $p > 0.05$, indicating no statistically significant differences between learning rules.

| Dataset | Comparison | $p$ (LZC acc.) |
|---|---|---|
| Bernoulli | All algorithm pairs | > 0.05 |
| Markov | All algorithm pairs | > 0.05 |
| Poisson | All algorithm pairs | > 0.05 |

**Table 8**
Wilcoxon signed-rank test for pairwise comparisons between learning rules (Tempotron, SpikeProp, BAL) on neuromorphic datasets. We report $p$-values for LZC-based accuracy and raw SNN accuracy. Bold values indicate statistically significant differences at $\alpha = 0.05$.

| Dataset | Comparison | $p$ (LZC acc.) | $p$ (SNN acc.) |
|---|---|---|---|
| MNIST01 | Tempotron vs SpikeProp | 0.32 | 0.32 |
| MNIST01 | Tempotron vs BAL | n/a | n/a |
| MNIST01 | SpikeProp vs BAL | 0.32 | 0.32 |
| FashionMNIST01 | Tempotron vs SpikeProp | 0.11 | 0.07 |
| FashionMNIST01 | Tempotron vs BAL | 0.32 | 0.18 |
| FashionMNIST01 | SpikeProp vs BAL | 0.10 | 0.07 |
| N-MNIST01 | Tempotron vs SpikeProp | 0.89 | 0.69 |
| N-MNIST01 | Tempotron vs BAL | 0.72 | **0.028** |
| N-MNIST01 | SpikeProp vs BAL | 0.83 | **0.028** |
| Gesture01 | Tempotron vs SpikeProp | 0.18 | 0.59 |
| Gesture01 | Tempotron vs BAL | 0.32 | 0.47 |
| Gesture01 | SpikeProp vs BAL | 1.00 | 0.27 |
| SHD01 | All pairs | n/a[1] | n/a[1] |

[1] Statistical comparison for SHD01 was omitted due to the limited number of stable repeated runs currently available for this dataset.

Thus, Poisson-distributed data present the greatest challenge but also offer the most biologically realistic testing ground. Across nearly all algorithms, performance was lower for Poisson inputs compared to Bernoulli or Markov, indicating difficulties in processing highly irregular spike trains. It is worth highlighting that biologically inspired and hybrid algorithms like SDSP and BAL demonstrated comparatively better adaptability, underscoring their value for neuromorphic systems. Additionally, the computational burden for Poisson inputs was higher for gradient-based approaches due to a lack of structure, which complicates error propagation and increases time complexity.

Tables 7 and 8 summarize the statistical reliability of LZC-based classification using Wilcoxon signed-rank tests across three SNN training algorithms (Tempotron, SpikeProp, BAL) and multiple data sources. For synthetic stochastic datasets (Bernoulli, Markov, Poisson), all pairwise comparisons yield $p > 0.05$, indicating no statistically significant differences between learning rules. A similar trend is observed for neuromorphic datasets, where most comparisons remain statistically indistinguishable, with only isolated differences appearing in raw SNN accuracy rather than in LZC-based decisions. These results support the observation that LZC representations produced by different learning rules remain statistically comparable across datasets. Non-significant Wilcoxon results indicate that differences between learning rules are not statistically supported at the level of LZC-based decisions.

Therefore, the reported differences between learning paradigms should be interpreted primarily as indicative trends in temporal representation dynamics and computational-efficiency regimes rather than as evidence of universally more favorable predictive performance of any individual algorithm. In particular, although some learning rules achieved higher mean accuracy values under specific experimental conditions, the statistical analysis suggests that many of these differences remain comparatively small at the level of LZC-based decisions. Consequently, the present framework is intended mainly to analyze how learning rules reshape temporal spike-train organization and complexity profiles under controlled conditions.

**Table 9**
Influence of hybrid learning algorithms on SNN performance.

| Type | Subtype | Dataset | Bio- | Ep. | $n$ | Time [s] | Acc. [%] | MSE | $R^2$ |
|---|---|---|---|---|---|---|---|---|---|
| Hybrid | ANN–SNN | Bernoulli | No | 10 | 64 | 21.41 | 99.50 | 0.0050 | 0.9800 |
| | | Markov | No | 10 | 32 | 37.25 | 92.58 | 0.0742 | 0.7031 |
| | | Poisson | No | 10 | 32 | 115.41 | 94.50 | 0.0550 | 0.7799 |
| | Reward | Bernoulli | No | 10 | 32 | 4.30 | 99.50 | 0.0050 | 0.9800 |
| | | Markov | No | 10 | 64 | 36.32 | 90.35 | 0.0965 | 0.9948 |
| | | Poisson | No | 10 | 64 | 14.83 | 91.00 | 0.0900 | 0.6399 |
| | BAL | Bernoulli | Yes | 10 | 32 | 1.56 | 97.00 | 0.0300 | 0.8800 |
| | | Markov | Yes | 10 | 128 | 46.37 | 95.38 | 0.0462 | 0.8153 |
| | | Poisson | Yes | 10 | 128 | 95.98 | 94.50 | 0.0550 | 0.7799 |

The absence of statistically significant differences in most Wilcoxon comparisons suggests that the observed performance differences between learning rules should be interpreted cautiously. Rather than indicating universally more favorable predictive performance of any single algorithm, the reported results primarily reveal differences in temporal representation dynamics, robustness under stochastic inputs, and computational efficiency regimes.

Accordingly, the emphasis of the present work is placed on a comparative analysis of learning-induced temporal organization rather than on establishing definitive ranking claims between algorithms.

Table 10 provides computational metrics including FLOPs per forward pass, runtime, latency per processed sample, and energy estimates. Energy values are analytical estimates derived from runtime and an assumed average power draw of the measurement platform; they should be interpreted as coarse proxies rather than hardware-measured neuromorphic energy. Therefore, the reported energy values should be interpreted comparatively rather than absolutely. Their primary purpose is to illustrate relative computational trends between learning rules under identical execution conditions, not to provide hardware-certified neuromorphic power measurements. These results demonstrate that all three SNN learning rules operate in a low-FLOPs regime, with per-sample latency below 0.25 ms for synthetic data and below 0.1 ms for MNIST after optimization, supporting real-time feasibility. Importantly, Lempel-Ziv complexity is a single-scalar descriptor of temporal structure rather than a discriminative classifier optimized for decision boundaries. When used with a fixed threshold, LZC-based decisions can become unstable under class imbalance or when class-conditional complexity distributions overlap, which may reduce the F1-score even if overall accuracy remains high. In this paper, we therefore interpret LZC primarily as an analysis-oriented representation that exposes how learning rules shape temporal information, while standard metrics (accuracy, MSE, $R^2$) quantify separability under the adopted protocol.

Our experiments reveal that the choice of learning algorithm must be carefully aligned with both the structure of the input data and application-specific computational constraints. Classic backpropagation and surrogate-gradient methods (e.g., STBP) achieve more favorable accuracy, particularly for structured signals. However, their computational demands are prohibitively high for real-time or resource-constrained deployments. On the other hand, bio-inspired learning algorithms such as tempotron, SDSP, and BAL offer a more favorable trade-off, delivering robust classification with substantially lower computational costs. BAL demonstrated favorable efficiency-accuracy trade-offs under stochastic inputs, balancing biological plausibility with operational efficiency.

Analysis across input types shows that classification accuracy is strongly influenced by the temporal correlation structure of the signals. Poisson-distributed inputs, despite being memoryless and weakly correlated, pose the greatest challenge for learning algorithms due to their high temporal variability and lack of exploitable structure. In contrast, Markovian sequences, with moderate temporal dependencies, offer a

**Table 10**
Computational efficiency metrics: runtime, latency, FLOPs, and estimated energy per sample.

| Dataset | Algorithm | Time per run [s] (mean) | Latency [ms/sample] (mean ± std) | FLOPs [G]/forward (mean) | Energy [mJ/sample] (mean) |
|---|---|---|---|---|---|
| Bernoulli | Tempotron | 46.67 | 233.34 ±4.17 | 0.000295 | 0.002949 |
| | SpikeProp | 46.63 | 233.16 ±3.74 | 0.000295 | 0.002949 |
| | BAL | 23.16 | 115.81 ±42.95 | 0.000295 | 0.002949 |
| Markov | Tempotron | 46.77 | 233.87 ±7.12 | 0.000295 | 0.002949 |
| | SpikeProp | 47.31 | 236.54 ±6.95 | 0.000295 | 0.002949 |
| | BAL | 30.75 | 153.77 ±65.54 | 0.000295 | 0.002949 |
| Poisson | Tempotron | 45.15 | 225.75 ±3.76 | 0.000295 | 0.002949 |
| | SpikeProp | 45.24 | 226.20 ±2.48 | 0.000295 | 0.002949 |
| | BAL | 28.93 | 144.65 ±70.30 | 0.000295 | 0.002949 |
| MNIST 0/1 | Tempotron | 208.96 | 70.69 ±9.00 | 0.000233 | 0.002335 |
| | SpikeProp | 105.49 | 35.69 ±21.89 | 0.000233 | 0.002335 |
| | BAL | 71.39 | 24.15 ±2.91 | 0.000233 | 0.002335 |

balanced scenario, while Bernoulli sequences, being memoryless and maximally entropic per symbol, provide a broad baseline for learning without temporal dependencies. These findings highlight the importance of aligning learning strategies with the statistical properties of input, especially in the design of SNNs for real-world data processing.

In general, achieving brain-like robustness under noisy stochastic input regimes, particularly those modeled by Poisson processes, will require continued refinement of adaptive, efficient, and biologically plausible learning strategies. The LZC-SNN framework proposed here contributes to this goal by offering a scalable, interpretable, and resource-efficient alternative to conventional classification pipelines in spiking neural systems.

While our previous study examined the role of neuron models, the present work suggests that learning paradigms exert an equally strong and often dominant influence on the temporal organization of spiking activity, as revealed by complexity-based analysis.

Although classification provides a controlled and interpretable setting for comparing temporal representations generated by different learning rules, it does not exhaustively characterize all functional properties of spiking neural networks. In particular, tasks involving temporal prediction, sequence generation, continual adaptation, reinforcement learning, or closed-loop neuromorphic control may expose additional differences between learning paradigms that are not reflected solely by classification metrics. Consequently, the reported results should be interpreted as evidence of learning-rule-induced differences in temporal organization under a controlled decision setting, rather than as a universal evaluation of SNN capabilities.

### 5.1. Comparison with non-LZC spike descriptors

We compared the proposed Lempel-Ziv complexity descriptor against two conventional spike-train descriptors: spike count and firing rate. To ensure a fair comparison, all three descriptors used the same nearest-class-prototype decision rule and identical train/test partitions across 10 independently seeded runs. Mean classification accuracy values are summarized in Tables 5– 9. Statistical paired comparisons were additionally evaluated using the Wilcoxon signed-rank test.

The relatively small absolute differences between LZC and conventional spike descriptors indicate that the primary contribution of the proposed framework does not lie solely in maximizing raw classification accuracy. Instead, LZC provides an interpretable characterization of learning-induced temporal organization and enables the analysis of complexity dynamics across different learning paradigms under controlled stochastic conditions.

Table 11 demonstrates that the effectiveness of spike-train descriptors is dependent on the learning rule used to generate hidden spike representations under Poisson distributed input processes, which

**Table 11**
Comparison of spike-train descriptors across hidden-spike representations generated by different SNN learning rules under Poisson distributed input processes. Reported values correspond to mean classification accuracy (%) obtained for spike-trains generated from Poisson processes, which constitute one of the most widely accepted statistical models of neuronal action potential (spike train) generation.

| Dataset | Descriptor | Classifier | Accuracy (%) |
|---|---|---|---|
| Hidden spikes (Chronotron) | Firing rate | Prototype | 86.42 |
| Hidden spikes (Chronotron) | Spike count | Prototype | 84.58 |
| Hidden spikes (Chronotron) | LZC | Prototype | 90.50 |
| Hidden spikes (Chronotron) | Raw spike vector | Logistic regression | 87.01 |
| Hidden spikes (ReSuMe) | Firing rate | Prototype | 75.24 |
| Hidden spikes (ReSuMe) | Spike count | Prototype | 71.50 |
| Hidden spikes (ReSuMe) | LZC | Prototype | 91.50 |
| Hidden spikes (ReSuMe) | Raw spike vector | Logistic regression | 89.30 |
| Hidden spikes (STDP) | Firing rate | Prototype | 71.50 |
| Hidden spikes (STDP) | Spike count | Prototype | 68.50 |
| Hidden spikes (STDP) | LZC | Prototype | 96.50 |
| Hidden spikes (STDP) | Raw spike vector | Logistic regression | 92.67 |

are widely considered one of the most biologically realistic statistical models of neuronal spike-train generation. Despite the substantial stochastic variability introduced by Poisson dynamics, the Lempel-Ziv Complexity descriptor consistently achieved the highest classification accuracy across all analyzed SNN learning methods, clearly outperforming conventional rate based descriptors such as firing rate and spike count.

The superiority of the LZC descriptor is particularly evident for supervised temporal learning rules. For Chronotron-generated hidden-spike representations, the LZC descriptor achieved 90.50% classification accuracy, compared with 86.42% for firing rate and 84.58% for spike count. Even the raw spike-vector representation combined with logistic regression reached only 87.01%, indicating that LZC captures discriminative temporal structure more effectively than both simplified statistical descriptors and direct vector-based classification.

An even stronger advantage of LZC can be observed for ReSuMe-based representations. While firing rate and spike count achieved only 75.24% and 71.50% accuracy, respectively, the LZC descriptor reached 91.50%, substantially outperforming all remaining descriptors, including the raw spike-vector representation classified with logistic regression (89.30%). This large performance gap suggests that ReSuMe-generated spike trains contain highly informative sequential patterns that are not sufficiently represented by simple spike accumulation measures.

The most pronounced effect appears for STDP-based hidden-spike representations. Under this learning paradigm, the LZC descriptor achieved the overall highest accuracy reported in Table 11, namely 96.50%, whereas firing rate and spike count achieved only 71.50% and 68.50%, respectively. Even the raw spike-vector representation reached a lower accuracy of 92.67%. These results strongly indicate that complexity-based temporal characterization is particularly effective for unsupervised spike-timing-dependent learning, where discriminative information is encoded primarily in the temporal organization and irregularity structure of spike sequences rather than in average firing statistics alone.

Overall, the results presented in Table 11 clearly demonstrate that the LZC descriptor provides the most robust and discriminative representation across different SNN learning paradigms under stochastic Poisson conditions. Unlike firing-rate and spike-count descriptors, which reduce spike trains to simple accumulation statistics, LZC preserves higher order temporal dependencies and sequential complexity information. Consequently, complexity oriented spike-train analysis appears substantially more suitable for capturing the rich temporal dynamics generated by biologically inspired spiking neural learning mechanisms.

### 5.2. Limitations of fixed-threshold LZC decisions

Lempel-Ziv complexity is a single-scalar summary of temporal novelty and is not optimized to directly maximize discriminative margins. When class priors are imbalanced or when class-conditional LZC distributions overlap, a fixed decision threshold may collapse into predicting the majority class. This leads to deceptively high classification accuracy while driving the F1-score toward zero, indicating poor minority-class recognition.

For this reason, LZC is interpreted in this work primarily as an analysis-oriented temporal representation that exposes how learning rules reorganize spike-train structure, rather than as a strong stand-alone classifier. Classification metrics are therefore used as downstream probes of separability under a specified protocol, and class-sensitive measures are reported to explicitly flag regimes in which scalar-complexity decisions become unreliable.

Another limitation concerns scalability to deeper and larger SNN architectures. The present study intentionally employs relatively compact feed-forward topologies to isolate the effects of learning rules on temporal organization and computational efficiency. In deeper architectures, additional phenomena such as spike sparsity propagation, vanishing activity, surrogate-gradient instability, and long-range temporal dependencies may substantially alter the observed complexity profiles. Consequently, the conclusions presented here should be interpreted primarily in the context of controlled low- to medium-scale temporal classification settings.

Moreover, another limitation of the present study is that the evaluation was restricted primarily to classification tasks. While this setting enables controlled comparisons between learning paradigms and facilitates the interpretation of complexity-based temporal representations, it does not fully capture the diversity of computational scenarios encountered in practical neuromorphic systems. Future work should therefore extend the proposed framework toward temporal prediction, continual learning, reinforcement-based adaptation, sequence generation, and real-time closed-loop processing tasks.

Also, some limitations arise from the use of scalar complexity representations. Since Lempel-Ziv complexity compresses spike-train structure into a single scalar quantity, distinct temporal patterns may occasionally produce overlapping complexity values. As a consequence, separability may deteriorate under strongly overlapping class-conditional distributions or class imbalance conditions. Future work may therefore investigate multidimensional complexity descriptors or hybrid temporal-information representations.

Another limitation concerns the currently limited number of repeated runs available for statistical evaluation. Although preliminary multi-seed experiments indicate moderate variability for most datasets, larger-scale repeated evaluations would provide more reliable confidence estimates and improve the statistical robustness of learning-rule comparisons.

Accordingly, the reported results should not be interpreted as universal performance guarantees for the considered learning paradigms, but rather as controlled observations obtained under the adopted temporal-classification protocol and complexity-based representation framework.

### 5.3. Complexity distribution analysis

To better understand how different learning paradigms reorganize temporal spike-train structure, we additionally analyzed the distributions of Lempel-Ziv complexity values obtained after training. Rather than focusing exclusively on classification accuracy, this analysis investigates how learning rules modify temporal regularity, variability, and overlap between class-conditional complexity distributions.

For Bernoulli inputs, most learning algorithms produce relatively compact and well-separated LZC distributions. Since Bernoulli sequences are memoryless and exhibit weak temporal dependencies, classification mainly depends on the ability of the learning rule to preserve differences in spike-generation statistics without introducing excessive temporal regularization. In this regime, supervised methods such as BP, STBP, and SpikeProp generate the clearest separation between class-conditional complexity profiles, whereas unsupervised rules tend to produce narrower distributions with partially overlapping regions.

For Markov datasets, the resulting LZC distributions exhibited larger variance and stronger sensitivity to the learning paradigm. Algorithms capable of exploiting temporal dependencies, particularly SpikeProp and Tempotron, generated more structured class-conditional complexity organization than local Hebbian-type updates. This suggests that temporally sensitive supervised rules more effectively preserve sequential correlations embedded in Markovian spike streams.

The high stochasticity and irregular temporal structure of Poisson spike trains reduce the discriminative power of scalar complexity representations and increase variability between runs. Nevertheless, biologically inspired adaptive mechanisms such as BAL and SDSP maintained comparatively more stable LZC profiles under stochastic conditions, indicating improved robustness to temporal noise and irregular spike timing.

For neuromorphic datasets such as MNIST01 and N-MNIST01, LZC distributions reflected differences between static image-derived and event-driven sensory representations. In particular, event-based datasets generated broader complexity distributions due to higher temporal variability and sparse asynchronous spike activity. Despite this increased variability, supervised spike-timing-based learning rules preserved relatively stable separability between classes, supporting the usefulness of temporal complexity descriptors for analyzing spike-based representations.

Importantly, the observed differences in LZC distributions often remained visible even when overall classification accuracies were similar. This indicates that learning rules with comparable predictive performance may nevertheless generate substantially different temporal representations internally. Consequently, complexity-distribution analysis provides complementary insight beyond standard benchmark metrics by exposing how learning dynamics reshape temporal organization in spiking neural networks.

Representative boxplots and histogram-based visualizations of class-conditional LZC distributions are provided for selected learning rules and datasets in order to illustrate the degree of distribution overlap, temporal variability, and complexity regularization induced during training.

Additional repeated-run statistics further indicate that several learning rules produce partially overlapping performance distributions despite differences in mean accuracy values. This observation is consistent with the Wilcoxon signed-rank test results and supports the interpretation that many observed differences should be viewed primarily as comparative trends in temporal representation dynamics rather than as definitive evidence of universal algorithmic superiority.

An important limitation of the proposed framework is that the final decision stage relies on scalar Lempel-Ziv complexity representations of spike trains. Although this enables interpretable comparison of temporal organization across learning paradigms, scalar complexity descriptors may not fully capture higher-order temporal dependencies, fine-grained spike correlations, or multidimensional dynamical structures present in large-scale spiking neural systems. Consequently, future work should investigate multiscale and multidimensional complexity representations capable of preserving richer temporal information.

## 6. Conclusions

Beyond accuracy-oriented evaluation, the present study suggests that learning rules can be interpreted as mechanisms that actively reshape the temporal information structure in spiking neural networks. This perspective provides an alternative to purely optimization-centered analyses and highlights that two algorithms with comparable predictive performance may nevertheless generate substantially different internal spike-train organizations and computational characteristics.

The statistical analysis performed using Wilcoxon signed-rank tests indicates that many observed differences between learning rules are not statistically significant at the level of LZC-based decisions. Consequently, the conclusions of this work should be interpreted primarily in terms of comparative trends in temporal representation dynamics, robustness, and computational efficiency rather than as definitive ranking statements between algorithms.

This study set out to examine the role of learning algorithms in shaping the classification performance and computational efficiency of Spiking Neural Networks. The study was designed primarily as a controlled methodological comparison rather than as a benchmark-oriented competition across large-scale downstream tasks. By systematically comparing classical, bioinspired, and hybrid training approaches across datasets with varying temporal structures—Bernoulli, Markov, and Poisson processes—we were able to characterize how algorithmic choices interact with data statistics to determine both accuracy and runtime.

Our results provide several insights. However, the statistical analysis indicates that several observed differences between learning rules remain comparatively small at the level of LZC-based decisions and should therefore be interpreted cautiously. First, gradient-based methods such as backpropagation and surrogate-gradient learning consistently achieved the highest predictive accuracy in terms of predictive precision, often achieving very high predictive accuracy and low error values under the evaluated experimental conditions. However, their computational costs were prohibitively high, in some cases exceeding 48,000 s (measured over 10 epochs on a standard desktop machine with Intel® Core™ i7-14700F @ 2.10 GHz), which makes them impractical for real-time or embedded neuromorphic deployments. Second, bioinspired algorithms such as Hebbian learning, STDP, SDSP, tempotron, and SpikProp offered substantially lower computational costs, often two orders of magnitude faster, while still maintaining competitive accuracy for structured sequences. Their limitations became apparent in the context of Poisson-distributed data, where stochastic variability hindered generalization and accuracy degraded. Third, hybrid schemes including ANN-SNN conversion and BAL were shown to mitigate some of these weaknesses by leveraging biologically plausible mechanisms that adapt more flexibly to irregular temporal inputs.

The methodological novelty of this work lies in the use of Lempel-Ziv complexity as an analysis-oriented temporal descriptor embedded within an SNN framework. Unlike prior studies where LZC served primarily as a descriptive post hoc metric, here it is used to analyze how different learning rules shape temporal representations and influence classification-related outcomes, rather than as a standalone discriminative classifier. This integration of information-theoretic complexity analysis with biologically plausible spiking computation offers two potential advantages within the considered experimental setting: it may improve interpretability by linking decision boundaries to structural novelty in the input, and it enhances efficiency by reducing reliance on gradient-based training. We believe this conceptual bridge between complexity theory and neuromorphic computing can serve as a useful template for future research in pattern recognition with SNNs.

At the same time, several limitations must be acknowledged. The experiments were conducted on synthetic datasets, which, while useful for systematically controlling temporal correlations, may not capture the full irregularities of real-world neural or sensory data. Moreover, although LZC-SNN achieved competitive accuracy, the method requires further validation on larger-scale problems and with hardware implementations to fully assess scalability. Finally, while we report coarse energy estimates, hardware-measured energy on neuromorphic processors remains out of scope and will be addressed in future work.

Despite these limitations, the results carry implications for both researchers and practitioners. For researchers, the comparative framework presented here provides a benchmark for assessing novel SNN learning rules under varying temporal conditions. For practitioners designing neuromorphic systems, the findings clarify trade-offs between accuracy and efficiency and highlight which algorithms are best suited for structured versus stochastic inputs.

The main strength of our approach lies in its interpretability and efficiency compared to conventional decoding strategies. At the same time, its accuracy is sensitive to the statistical properties of the input and to the choice of learning rules, which we identified as a limiting factor for real-world deployment. These insights can guide researchers in selecting appropriate training strategies depending on task constraints.

Although the present work focuses primarily on classification tasks, the proposed framework is not restricted to classification-only settings. Since the analysis is based on temporal spike-train organization and complexity dynamics, the same methodology can be naturally extended to temporal prediction, continual learning, reinforcement-based adaptation, event forecasting, and closed-loop neuromorphic control. Classification was intentionally selected as a controlled benchmark scenario that enables direct comparison of learning-rule-induced temporal representations under identical experimental conditions.

Future work should proceed along several directions. Extending the evaluation to real-world datasets such as electrophysiological recordings, speech signals, or event-based vision would test the robustness of the observed trends. Hardware level experiments are needed to quantify energy savings and latency reductions in neuromorphic processors. Methodologically, there is an opportunity to design adaptive or hybrid algorithms that dynamically switch between gradient-based and bioinspired updates depending on the input statistics, thereby combining accuracy with efficiency. Finally, further exploration of information-theoretic measures beyond LZC, such as entropy rate or predictive information, may provide complementary tools for interpretable spiking computation.

Our findings further suggest that the suitability of a given learning paradigm depends strongly on the target operating regime. Gradient-based approaches remain advantageous in applications prioritizing maximal predictive accuracy, whereas biologically inspired and hybrid algorithms provide more favorable trade-offs for low-power and real-time neuromorphic systems. We believe that combining information-theoretic analysis with spike-based computation offers a promising direction toward interpretable, adaptive, and resource-efficient artificial neural systems.

In summary, this work highlights the decisive role of learning algorithms in determining the balance between accuracy, efficiency, and biological plausibility in SNNs. By embedding complexity-theoretic principles into spiking architectures, we take a step toward interpretable and resource-efficient pattern recognition systems that more closely align with the constraints of real-world neuromorphic applications.

## 7. Practical implications

The findings suggest that classical gradient-based algorithms should be reserved for offline or high-resource scenarios requiring maximal accuracy, whereas bioinspired and hybrid methods provide a more favorable balance for real-time and resource constrained neuromorphic applications. The proposed LZC-SNN framework, by combining spiking dynamics with information-theoretic analysis, offers a scalable and interpretable approach to temporal pattern recognition, with potential applications in brain-computer interfaces, biomedical signal analysis, and event-driven sensory processing.

From a practical perspective, the obtained results suggest distinct operating regimes for different learning paradigms. Gradient-based methods are most suitable for offline high-precision applications where computational cost is secondary, whereas biologically inspired local rules are preferable for low-power adaptive systems operating under strict latency constraints. Hybrid methods occupy an intermediate regime,

offering improved robustness under stochastic temporal inputs while maintaining moderate computational requirements. This separation may help guide the design of future neuromorphic systems depending on the balance required between interpretability, efficiency, and predictive accuracy.

The repeated-run variability analysis additionally suggests that biologically inspired and hybrid learning rules may provide comparatively stable temporal representations under stochastic spike-generation conditions, particularly for irregular Poisson-like inputs.

## 8. Experimental protocol and reproducibility

To ensure reproducibility and fairness of comparison, all learning rules were evaluated under the same experimental protocol. All experiments used identical input dimensionality, training/test splits, sequence length, epoch count, and predefined hyperparameter search ranges. No algorithm-specific architecture modifications or task-specific optimization procedures were introduced beyond the shared parameter grid.

Synthetic datasets were divided into training, validation, and test subsets using fixed random splits with identical proportions across all experiments. The same protocol was applied consistently across unsupervised, supervised, and hybrid learning paradigms to ensure comparability of the obtained results.

For all binary classification tasks, class-balanced subsets were used in order to avoid bias toward majority-class predictions. Training, validation, and test partitions preserved identical class proportions across all compared learning rules and repeated runs.

Hyperparameter selection was performed using a fixed search space covering firing thresholds, membrane decay coefficients, learning rates, and layer sizes. For each learning rule, the final reported configuration corresponded to the parameter setting achieving the highest validation accuracy. When multiple configurations produced identical validation accuracy, the configuration with the lower computational cost was selected.

To reduce sensitivity to stochastic initialization and spike-generation variability, each experiment was repeated over 10 independent random seeds using identical training and evaluation conditions. The values reported in Tables 5–9 correspond to mean performance across repeated runs. Preliminary analysis indicated relatively low variance for Bernoulli and Markov datasets, whereas larger fluctuations were observed for Poisson-distributed inputs due to their inherently stochastic temporal structure.

The framework may also be relevant for real-world neuromorphic applications involving temporal event streams, including adaptive sensory processing, online anomaly detection, and spike-based biomedical signal analysis.

The source code, parameter configurations, and dataset-generation procedures used in this study can be made available upon reasonable request for research and reproducibility purposes (Table 12).

For reproducibility analysis, additional aggregate summaries including mean and standard deviation values for accuracy, MSE, $R^2$, latency, runtime, and computational-efficiency metrics were computed across repeated runs for all evaluated learning rules and datasets (Table A.14).

**Table 12**
Summary of the experimental protocol used for reproducibility and fair comparison.

| Protocol component | Setting |
|---|---|
| Neuron model | Leaky Integrate-and-Fire (LIF) |
| Network topology | Three-layer feed-forward architecture |
| Input sequence length | 1024 |
| Epoch count | 10 |
| Random seeds | 10 independent runs |
| Hyperparameter selection | Highest validation accuracy; lower runtime used as tie-breaker |
| Evaluation metrics | Accuracy, MSE, $R^2$, computation time |
| Fairness constraint | Identical input dimensionality, splits, search ranges, and epoch count |

## CRediT authorship contribution statement

**Zofia Rudnicka:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Methodology, Investigation, Formal analysis. **Janusz Szczepanski:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Agnieszka Pregowska:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. Hyperparameter search space

To ensure methodological consistency and fairness of comparison, all learning rules were evaluated using the same predefined hyperparameter search space. No algorithm-specific tuning outside the shared parameter grid was introduced (Table A.13).

**Table A.13**
Hyperparameter search ranges used for all learning paradigms.

| Parameter | Search range |
|---|---|
| Firing threshold | $\{1.00, 0.50, 0.40, 0.30, 0.20, 0.10, 0.05\}$ |
| Membrane decay coefficient | $\{0.100, 0.050, 0.030, 0.010\}$ |
| Learning rate | $\{0.0500, 0.0100, 0.0010, 0.0001\}$ |
| Neurons per layer | $\{16, 32, 64, 128, 256, 512, 1024\}$ |
| Epoch count | 10 |
| Input sequence length | 1024 |
| Neuron model | Leaky Integrate-and-Fire (LIF) |
| Training protocol | Identical across all algorithms |

For each learning rule, the final reported configuration corresponded to the parameter setting achieving the highest validation accuracy. When multiple configurations yielded identical validation accuracy, the configuration with lower computational cost was selected.

**Table A.14**
Repeated-run variability analysis for representative learning rules. Reported values correspond to the mean ± standard deviation across repeated runs.

| Dataset | Algorithm | Runtime [s] | Latency [ms] | $n_{\text{runs}}$ |
|---|---|---|---|---|
| Bernoulli | BP | $41.43 \pm 1.91$ | $207.17 \pm 9.54$ | 30 |
| Bernoulli | BAL | $1.44 \pm 0.01$ | $7.19 \pm 0.04$ | 30 |
| Bernoulli | Chronotron | $16.83 \pm 0.23$ | $84.16 \pm 1.14$ | 30 |
| Markov | BP | $297.89 \pm 11.43$ | $1489.47 \pm 57.17$ | 30 |
| Markov | BAL | $28.98 \pm 1.15$ | $144.91 \pm 5.77$ | 30 |
| Poisson | BP | $608.62 \pm 23.45$ | $3043.08 \pm 117.25$ | 30 |
| Poisson | BAL | $15.03 \pm 0.62$ | $75.13 \pm 3.08$ | 30 |

## Data availability

Data will be made available upon request.

## References

[1] Z. Rudnicka, J. Szczepanski, A. Pregowska, Impact of neuron models on spiking neural network performance: a complexity-based classification approach, Neuroinformatics 24 (5) (2026) 1–23, https://doi.org/10.1007/s12021-025-09754-1
[2] J.H. Lee, T. Delbruck, M. Pfeiffer, Training deep spiking neural networks using backpropagation, front, Neurosci. 10 (2016) 508, https://doi.org/10.3389/fnins.2016.00508

[3] Z. Zhang, X.P. Li, Q. Liu, SAEN-BGS: energy-efficient spiking autoencoder network for background subtraction, Pattern Recognit. 169 (2026) 111792, https://doi.org/10.1016/j.patcog.2025.111792
[4] Z. Rudnicka, J. Szczepanski, A. Pregowska, Artificial intelligence-based algorithms in medical image scan segmentation and intelligent visual content generation - a concise overview, Electronics 13 (4) (2024) 746 1–35, https://doi.org/10.3390/electronics13040746
[5] D. Zhao, G. Shen, Y. Dong, Y. Li, Y. Zeng, Improving stability and performance of spiking neural networks through enhancing temporal consistency, Pattern Recognit. 159 (2025) 111094, https://doi.org/10.1016/j.patcog.2024.111094
[6] D.O. Hebb, The Organization of Behavior: A Neuropsychological Theory, first ed, Psychology Press, 2002, https://doi.org/10.4324/9781410612403
[7] A.G. Khoee, A. Javaheri, S.R. Kheradpisheh, M. Ganjtabesh, Meta-learning in spiking neural networks with reward-modulated STDP, Neurocomputing 600 (2024) 128173, https://doi.org/10.1016/j.neucom.2024.128173
[8] J. Tang, J.H. Lai, X. Xie, L. Yang, W.S. Zheng, AC2AS: activation consistency coupled ANN-SNN framework for fast and memory-efficient SNN training, Pattern Recognit. 144 (2023) 109826, https://doi.org/10.1016/j.patcog.2023.109826
[9] Y. Hosain, M. Cakmak, XAI-XGBoost: an innovative explainable intrusion detection approach for securing internet of medical things systems, Sci. Rep. 15 (2025) 22278, https://doi.org/10.1038/s41598-025-07790-0
[10] M.B. Ozel, S.B.A. Kartbak, M. Cakmak, Classification performance of deep learning models for the assessment of vertical dimension on lateral cephalometric radiographs, Diagnostics 15 (2025) 2240, https://doi.org/10.3390/diagnostics15172240
[11] M. Cakmak, A new lightweight hybrid model for pistachio classification using transformers and EfficientNet, IEEE Access 13 (2025) 85857–85872, https://doi.org/10.1109/ACCESS.2025.3567774
[12] R. Gütig, H. Sompolinsky, The tempotron: a neuron that learns spike timing-based decisions, Nat. Neurosci. 9 (2006) 420–428, https://doi.org/10.1038/nn1643
[13] G. Chen, G. Wang, TSTKD: triple-spike train kernel-driven supervised learning algorithm, Pattern Recognit. 164 (2025) 111525, https://doi.org/10.1016/j.patcog.2025.111525
[14] S.B. Shrestha, Q. Song, Adaptive learning rate of SpikeProp based on weight convergence analysis, Neural Netw. 63 (2015) 185–198, https://doi.org/10.1016/j.neunet.2014.12.001
[15] K. Laddach, R. Łangowski, Adjusted SpikeProp algorithm for recurrent spiking neural networks with LIF neurons, appl, Soft Comput. 165 (2024) 112120, https://doi.org/10.1016/j.asoc.2024.112120
[16] R. Florian, The chronotron: a neuron that learns to fire temporally-precise spike patterns, Nat. Prec. (2010), https://doi.org/10.1038/npre.2010.5190.1
[17] Y. Song, L. Guo, M. Man, Y. Wu, The spiking neural network based on fMRI for speech recognition, Pattern Recognit. 155 (2024) 110672, https://doi.org/10.1016/j.patcog.2024.110672
[18] R. Midya, S. Santra, H. Kim, M. Jang, C. Jung, Y. Bae, J. Woo, S. Kwon, Artificial neural network (ANN) to spiking neural network (SNN) converters based on diffusive memristors, Adv. Electron. Mater. 5 (2019) 1900060, https://doi.org/10.1002/aelm.201900060
[19] R. Rajagopal, A. Awasthi, M. Sharma, V. Bansal, Deep convolutional spiking neural network optimized with arithmetic optimization algorithm for lung disease detection using chest X-ray images, Biomed. Signal Process. Control 79 (2023) 104197, https://doi.org/10.1016/j.bspc.2022.104197
[20] P. Wu, E. Tian, H. Tao, Y. Chen, Data-driven spiking neural networks for intelligent fault detection in vehicle lithium-ion battery systems, Eng, Appl. Artif. Intell. 141 (2025) 109756, https://doi.org/10.1016/j.engappai.2024.109756
[21] L. Guo, X. Liu, Y. Wu, Anti-damage ability of biologically plausible spiking neural network with synaptic time delay based on speech recognition under random attack, Eng, Appl. Artif. Intell. 144 (2025) 110006, https://doi.org/10.1016/j.engappai.2025.110061
[22] A. Pregowska, J. Szczepanski, E. Wajnryb, Temporal code versus rate code for binary information sources, Neurocomputing 216 (2016) 756–762, https://doi.org/10.1016/j.neucom.2016.08.034
[23] M. Patankar, V. Chaurasia, M. Shandilya, A novel spiking neural network method for classification of tuberculosis using X-ray images, comput, Electr. Eng. 122 (2025) 110003, https://doi.org/10.1016/j.compeleceng.2024.110003
[24] Y. Wu, L. Deng, G. Li, J. Zhu, L. Shi, Direct training for spiking neural networks: faster, larger, better, proc, AAAI Conf. Artif. Intell. 33 (2019) 1311–1318, https://doi.org/10.1609/aaai.v33i01.33011311
[25] H. Zhang, Y. Li, B. He, X. Fan, Y. Wang, Y. Zhang, Direct training high-performance spiking neural networks for object recognition and detection, Front. Neurosci. 17 (2023), https://doi.org/10.3389/fnins.2023.1229951
[26] Q. Zhang, Y. Chen, Z. Li, Bio-inspired active learning method in spiking neural network, Knowl.-Based Syst. 261 (2022) 110193, https://doi.org/10.1016/j.knosys.2022.110193
[27] J. Ziv, A. Lempel, On the complexity of finite sequences, IEEE Trans. Inf. Theory 22 (1976) 75–81, https://doi.org/10.1109/TIT.1976.1055501
[28] G. Amato, F. Carrara, F. Falchi, C. Gennaro, G. Lagani, Hebbian learning meets deep convolutional neural networks, In E. Ricci, S. Rota Bulò, C. Snoek, O. Lanz, S. Messelodi, N. Sebe (Eds.), Image Analysis and Processing ICIAP 2019. ICIAP 2019. Lecture Notes in Computer Science, vol. 11751, Springer, Cham, 2019.
[29] Q. Yu, H. Tang, K. Tan, H. Li, A brain-inspired spiking neural network model with temporal encoding and learning, Neurocomputing 138 (2014) 3–13, https://doi.org/10.1016/j.neucom.2013.06.052
[30] A. Taherkhani, A. Belatreche, Y. Li, L.P. Maguire, DL-ReSuMe: a delay learning-based remote supervised method for spiking neurons, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 3137–3149, https://doi.org/10.1109/TNNLS.2015.2404938
[31] Z. Yan, T. Lin, Y. Chen, et al., Low latency conversion of artificial neural network models to rate-encoded spiking neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2025), https://doi.org/10.1109/TNNLS.2025.3526374
[32] X. Xie, Y. He, Z. Chen, Y. Zhai, Effective active learning method for spiking neural networks, IEEE Trans. Neural Netw. Learn. Syst. 35 (2024) 12373–12382, https://doi.org/10.1109/TNNLS.2023.3257333
[33] S. Dutta, M. Sharad, D. Fan, K. Roy, Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET, sci, Rep. 7 (2017) 8257, https://doi.org/10.1038/s41598-017-07418-y
[34] B. Paprocki, A. Pregowska, J. Szczepanski, Does adding of neurons to the network layer lead to increased transmission efficiency? IEEE Access 12 (2024) 42701–42709, https://doi.org/10.1109/ACCESS.2024.3379324
[35] A. Pregowska, K. Proniewska, P. van Dam, J. Szczepanski, Using lempel-ziv complexity as effective classification tool of the sleep-related breathing disorders, comput, Methods Programs Biomed. 182 (2019) 105052, https://doi.org/10.1016/j.cmpb.2019.105052
[36] A. Pregowska, E. Kaplan, J. Szczepanski, How far can neural correlations reduce uncertainty? Comparison of information transmission rates for Markov and Bernoulli processes, Int. J. Neural Syst. 29 (2019) 1950003, https://doi.org/10.1142/S0129065719500035
[37] T. Britvina, J.J. Eggermont, A Markov model for interspike interval distributions of auditory cortical neurons that do not show periodic firings, biol, Cybern. 96 (2006) 245–264, https://doi.org/10.1007/s00422-006-0115-3
[38] F. Rieke, D. Warland, R.R. van Steveninck, W. Bialek, Spikes: Exploring the Neural Code, MIT Press, Cambridge, MA, 1997.