



OPEN ACCESS

EDITED BY

Huajin Tang,
Zhejiang University, China

REVIEWED BY

Jean-Philippe Thivierge,
University of Ottawa, Canada
Sadique Sheik,
SynSense, Switzerland

*CORRESPONDENCE

Bernard Girau
✉ bernard.girau@loria.fr

RECEIVED 30 June 2023

ACCEPTED 27 October 2023

PUBLISHED 21 November 2023

CITATION

Fois A and Girau B (2023) Enhanced representation learning with temporal coding in sparsely spiking neural networks. *Front. Comput. Neurosci.* 17:1250908. doi: 10.3389/fncom.2023.1250908

COPYRIGHT

© 2023 Fois and Girau. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Enhanced representation learning with temporal coding in sparsely spiking neural networks

Adrien Fois and Bernard Girau*

Université de Lorraine, Centre National de la Recherche Scientifique, Laboratoire lorrain de Recherche en Informatique et ses Applications, Nancy, France

Current representation learning methods in Spiking Neural Networks (SNNs) rely on rate-based encoding, resulting in high spike counts, increased energy consumption, and slower information transmission. In contrast, our proposed method, Weight-Temporally Coded Representation Learning (W-TCRL), utilizes temporally coded inputs, leading to lower spike counts and improved efficiency. To address the challenge of extracting representations from a temporal code with low reconstruction error, we introduce a novel Spike-Timing-Dependent Plasticity (STDP) rule. This rule enables stable learning of relative latencies within the synaptic weight distribution and is locally implemented in space and time, making it compatible with neuromorphic processors. We evaluate the performance of W-TCRL on the MNIST and natural image datasets for image reconstruction tasks. Our results demonstrate relative improvements of 53% for MNIST and 75% for natural images in terms of reconstruction error compared to the SNN state of the art. Additionally, our method achieves significantly higher sparsity, up to 900 times greater, when compared to related work. These findings emphasize the efficacy of W-TCRL in leveraging temporal coding for enhanced representation learning in Spiking Neural Networks.

KEYWORDS

spiking neural networks, temporal code, spike-timing-dependent plasticity, representation learning, visual representations, latency-coding, sparsity, unsupervised learning

1 Introduction

Spiking Neural Networks (SNNs) have been gaining recognition due to their application in supervised (Kheradpisheh and Masquelier, 2020; Lee et al., 2020), reinforcement (Mozafari et al., 2018; Patel et al., 2019), and unsupervised learning tasks (Diehl and Cook, 2015; Kheradpisheh et al., 2018). Compared to Artificial Neural Networks (ANNs) that output continuous values synchronously, SNNs transmit binary outputs sparsely and asynchronously as spikes. This event-based information transmission scheme reduces communication channels, significantly lowers energy requirements, and offers potential energy efficiency gains of up to 1,000 times with neuromorphic processors (Furber, 2016) compared to traditional processors. The massively parallel architecture of neuromorphic processors, where memory (synapses) and computational units (neurons) are co-located, further contributes to this energy efficiency. By utilizing local event-based learning rules, such as Spike-timing-dependent plasticity (STDP), which leverage information from presynaptic and postsynaptic terminals, the principles of parallelism and locality can be fully exploited at the algorithmic level.

Extracting representations from event-based and asynchronous data streams using local rules compatible with neuromorphic processors holds great promise. In this context, King et al. (2013) proposed a spatially local rule learning receptive fields resembling Gabor filters.

Burbank (2015) developed a method that reproduces the behavior of an autoencoder with an STDP rule. This STDP rule learns encoding and decoding weights with a Hebbian and anti-Hebbian rule, respectively. The combination of these two rules allows the approximation of the cost function of an autoencoder. More recently, Tavanaei et al. (2018) proposed an STDP rule, integrating a vector quantization module and a regularization module and provides the best performance in terms of reconstruction error.

However, all these methods operate with spike rate encoding, where information is encoded in the number of emitted spikes. Spike rate encoding incurs high energy costs, and leads to slow information transmission. In contrast, temporal codes, which transmit information based on spike times rather than spike rates, offer a more economical alternative. Temporal encoding using relative latencies achieves significantly higher energy efficiency, reduces the number of spikes required, and enables faster data transmission rates compared to rate coding. This makes temporal codes particularly relevant for neuromorphic computing and learning (Guo et al., 2021).

In this paper, we introduce a two-layer SNN that leverages temporal codes, specifically population-based latency coding, where each neuron fires a single spike. Our Weight-Temporally Coded Representation Learning (W-TCRL) model improves the reconstruction performance while increasing the sparsity of the activity in the network. We propose a novel STDP rule that adjusts synaptic weights based on spike times, operating locally in both space and time, thus facilitating a future implementation on neuromorphic processors (Davies et al., 2018). To the best of our knowledge, this is the first presented method for learning representations from a temporal encoding of inputs using an STDP-like rule aimed at achieving low reconstruction error. We evaluate our spiking model using the MNIST dataset and a natural image dataset.¹ Furthermore, we propose a generic parameterization of the model to address the common adaptability issue with data of varying dimensions.

2 Materials and methods

In this section, the various architectural and algorithmic components of the neural model are illustrated, which will be utilized for representation learning. The model is assessed on an image reconstruction task, with a focus on its new STDP rule that targets synaptic weights.

2.1 Spiking neural network

As depicted in Figure 1, our SNN architecture consists of two fully interconnected layers: the first layer encodes input data into relative spike latencies, generating a single spike per neuron, while the second layer extracts representations from the resulting spiking activity.

¹ The code is released: <https://github.com/afois/W-TCRL>.

2.1.1 Neuron and synapse model

The network contains only LIF neurons with current-based (CuBa) synapses. This widely-used model captures the basic behavior of a biological neuron while maintaining low computational cost and ease of analysis. The dynamic of a LIF neuron is given by:

$$\tau_m \frac{dV(t)}{dt} = -V(t) + I(t) + g\eta(t)$$

Neurons face a Gaussian white noise process η , scaled by g ; $g = 0$ (no noise) for most tests. However, we test the robustness of the network on MNIST dataset in Section 3.3 by varying g . We add the output $s(t)$ and the resetting of the membrane potential $V(t)$ when the firing threshold V_θ is reached, $V(t)$ is then maintained at 0 during a refractory period T_{refrac} :

$$\begin{aligned} \text{if } V(t) < V_\theta, & \quad \text{then } s(t) = 0 \\ \text{if } V(t) \geq V_\theta, & \quad \text{then } \begin{cases} s(t) = 1 \\ V(u) = 0 \forall u \in]t, t + T_{\text{refrac}}] \end{cases} \end{aligned}$$

Between the two layers, synaptic transmission is modeled by an exponential synapse model. If at least one presynaptic neuron i fires, the indicator function $s_i(t)$ takes the value 1 (0 otherwise) and the input to the postsynaptic neuron $I_j(t)$ changes instantaneously by an amount equal to the synaptic weight w_{ji} :

$$I_j(t) \leftarrow I_j(t) + \sum_{i=1}^n w_{ji}s_i(t)$$

otherwise $I_j(t)$ exponentially decays over time with a time constant τ_f :

$$\tau_f \frac{dI_j(t)}{dt} = -I_j(t)$$

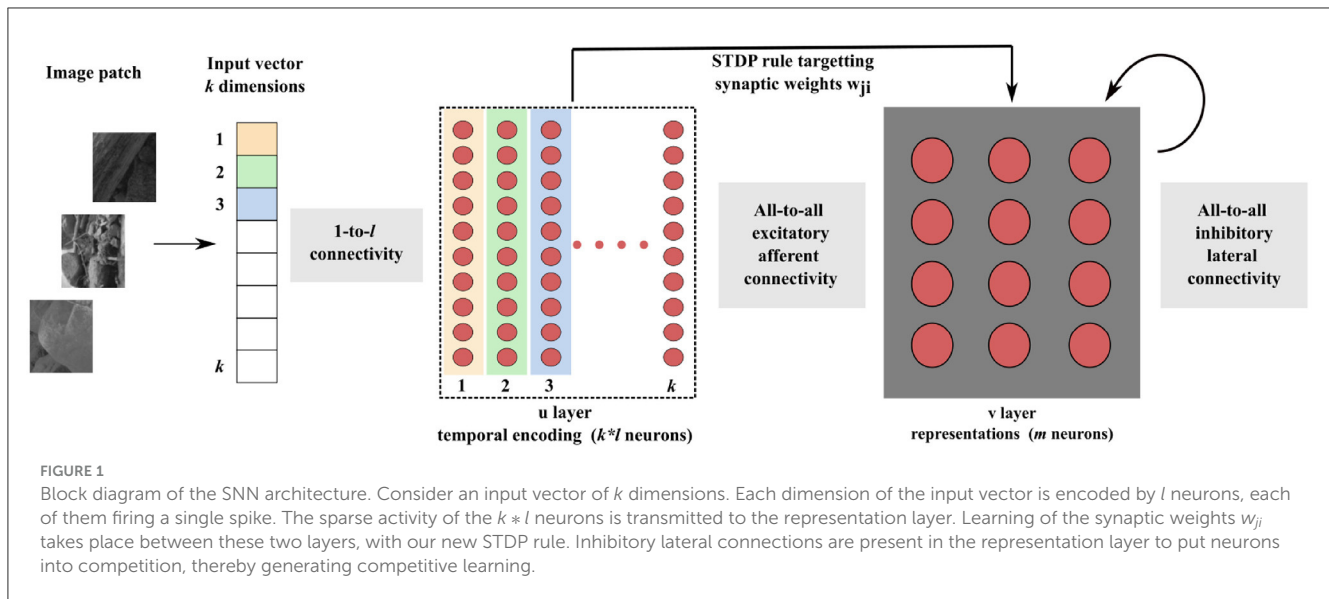
2.1.2 Synaptic traces

Synapses connect the n presynaptic neurons of the encoding layer to the m postsynaptic neurons of the representation layer. Each synapse has access to a local state variable $x_i(t)$ (with $i = 1, 2, \dots, n$) that tracks recent presynaptic spikes. This variable is known as presynaptic trace. It is commonly used for efficient and local implementations of STDP rules, both in simulations (Pfister and Gerstner, 2006) and neuromorphic processors (Davies et al., 2018).

Similarly, a postsynaptic trace $y_j(t)$ (with $j = 1, 2, \dots, m$) accessible to the synapses tracks recent postsynaptic spikes. When a pre (post)-synaptic spike occurs, $x_i(t)$ ($y_j(t)$) jumps to 1 and then decays exponentially to 0 with a time constant τ_x (τ_y):

$$\begin{aligned} x_i(t) & \leftarrow 1, & \text{if } s_i(t) = 1 \\ \tau_x \frac{dx_i(t)}{dt} & = -x_i(t), & \text{otherwise} \end{aligned} \quad (1)$$

$$\begin{aligned} y_j(t) & \leftarrow 1, & \text{if } s_j(t) = 1 \\ \tau_y \frac{dy_j(t)}{dt} & = -y_j(t), & \text{otherwise} \end{aligned} \quad (2)$$



where $s(t)$ is an indicator function returning 1 when a neuron fires a spike at time t , 0 otherwise.

2.2 Encoding input data in relative spike latencies

Let $\mathbf{a} \in [0, 1]^k$ denote a normalized k -dimensional real-valued input vector that we aim to encode into relative spike latencies (or timing) through population coding (Ebitz and Hayden, 2021). This can be achieved by distributing each dimension of the input vector over the population activity of l neurons (Figure 2). We used a population of $l = 10$ neurons to collectively represent one dimension. Each neuron $i \in \{1, 2, \dots, l\}$ of the population has an associated gaussian receptive field in a circular space in range $[0, 1]$, characterized by its preferential value (or center) μ_i and by its width (or standard deviation) σ . The centers μ_i are uniformly spread between 0.05 and 0.95. The standard deviation $\sigma = 0.6$ is the same for all neurons. This broad tuning ensures that any neuron receives high enough activation levels to fire in response to any input value. The resulting receptive fields overlap, covering the entire input space.

A gaussian function $G_{\mu, \sigma} : \mathbf{a} \rightarrow \mathbf{A}$ is then used to transform the input vector $\mathbf{a} \in [0, 1]^k$ into a vector of activations levels $\mathbf{A} \in [0, 1]^{k \times l}$ that feed the $n = k \times l$ LIF neurons of the encoding layer.

To illustrate the encoding process, consider a single entry of the input vector $a \in \mathbf{a}$. The i th neuron—with $i \in \{1, 2, \dots, l\}$ —whose gaussian receptive field center μ_i is the closest to the input value a , gets the highest activation level A_i and will thus fire first. The other neurons fire later. The higher the distance between their centers and the input value, the lower the activation levels they get, and therefore the later they fire. The temporal separation of spikes is accentuated by the non-linear integration of the received activation levels by LIF neurons. With this mechanism, a specific input value is encoded in a specific spike pattern in the spatio-temporal domain. This population-based coding does not lie within

the times to first spikes, but explicitly within the relative latencies of the spikes emitted by each neuron.

Note that in the scenario of continuous input presentation, this circuit will disrupt the temporal representation of the analog inputs. Inputs need to be somewhat reset to achieve spike time reproducibility. However, it is straightforward to replace this circuit with others such as the one proposed by Rumbell et al. (2014). This method enables continuous input presentation to the network by producing oscillatory firing through an inhibitory mechanism. Thus, it generates an internal time reference through oscillations. Both encoding methods utilize a population of Gaussian receptive fields and yield identical spiking patterns. Rumbell’s method offers greater flexibility at the cost of a more complex encoding circuit.

2.3 W-TCRL: STDP rule for learning representations in weights

Non-weight-dependent Spike-Timing-Dependent Plasticity (NSTDP) rules can produce stable receptive fields; however, they are bistable, leading to synaptic weights saturating at minimum and maximum values (Song et al., 2000). Hence, the potential range of learning parameters, the synaptic weights in this case, is not fully utilized under NSTDP rules. This limitation significantly impedes the expressive capacity of synaptic weights, diminishing their capability to finely represent external states, such as environmental states.

On the other hand, Weight-dependent Spike-Timing-Dependent Plasticity (WSTDP) rules rectify this issue by generating unimodal and continuous distributions of synaptic weights, thereby leveraging the entire range of weight variation. However, these rules fall short in generating stable receptive fields (Billings and van Rossum, 2009).

Our STDP rule provides the dual advantages of stability and the complete utilization of the synaptic weight range for representation storage. The interplay between our STDP rule and the unimodal temporal code results in a stable and unimodal

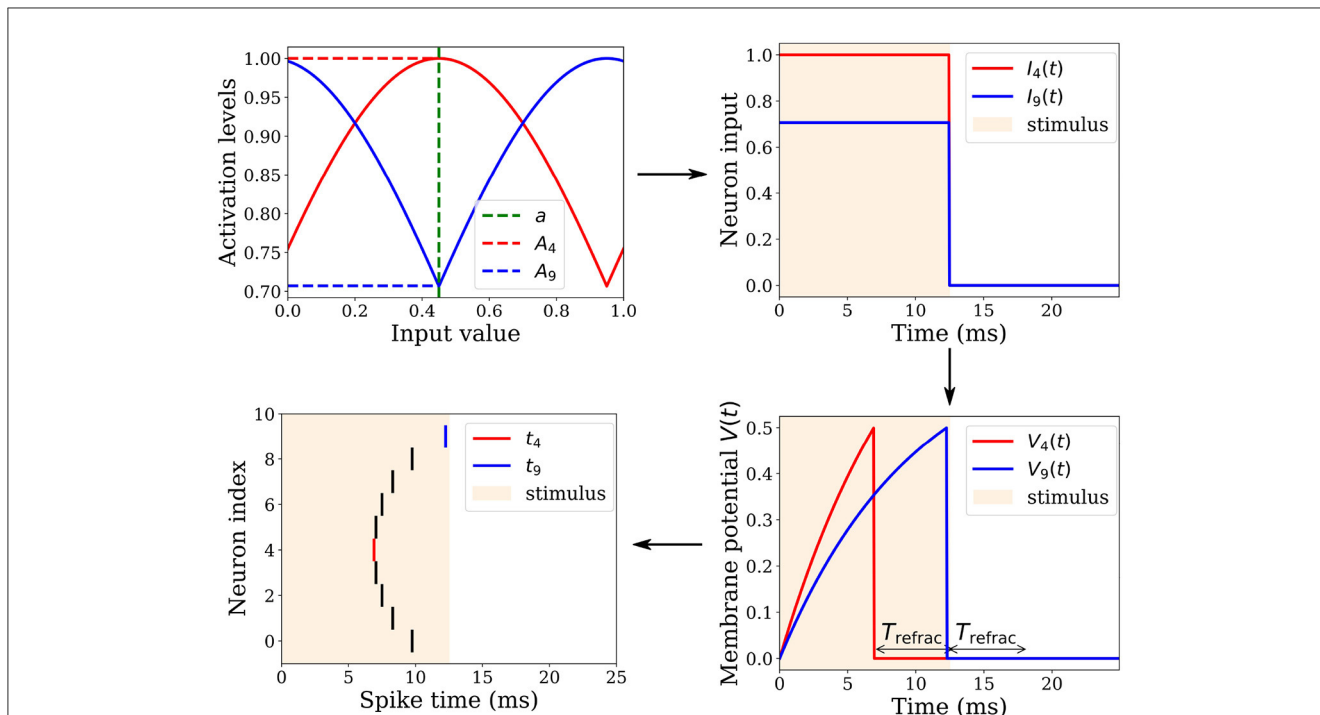


FIGURE 2 Spatio-temporal spike pattern encoding an input value of $a = 0.45$. **Top left:** The input is received by a population of $l = 10$ neurons, each having a Gaussian tuning curve centered on a preferential value μ_i . For instance, Neuron 4 and 9 possess $\mu_4 = 0.45$ and $\mu_9 = 0.95$, respectively. Their tuning curves and corresponding activation levels A_4 and A_9 for the input value are depicted in red and blue, respectively. **Top right:** These activation levels remain constant over a 12.5 ms interval, followed by an equal duration of quiescence. **Bottom right:** The neuronal inputs are integrated into membrane potentials V_4 and V_9 . With a higher activation level, V_4 grows faster than V_9 , enabling it to reach its firing threshold V_θ sooner. Upon crossing this threshold, the neuron fires a spike, resets its membrane potential, and enters a refractory period T_{refrac} . **Bottom left:** A neuron fires a spike when it crosses its threshold V_θ , thereby distributing an input value across a neuronal population to produce a specific spatio-temporal spike pattern.

distribution of synaptic weights (see Figure 4). Here, the magnitude of a synaptic weight is function of the temporal distance between a presynaptic and a postsynaptic spike. In the encoding layer u , faster firing neurons carry a more accurate representation of the input, correspondingly leading to a higher value of the synaptic weight. This, in turn, boosts the neuron’s causal influence over the emission of a postsynaptic spike.

Our novel STDP rule, derived from a vector quantization criterion, operates both spatially and temporally. Spatially, it focuses on a single pair of presynaptic and postsynaptic neurons i and j , and temporally, it operates within a defined temporal window.

The change in synaptic weight Δw_{ji} is calculated based on the presynaptic trace $x_i(t) \in]0, 1]$ and the postsynaptic trace $y_j(t) \in]0, 1]$. The synaptic weight $w_{ji} \in [0, 1]$ is then updated by this change $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$. Our new STDP rule is defined as follows (see Figure 3):

$$\Delta w_{ji} = \begin{cases} +\alpha^+(1 - x_i(t) - w_{ji} + w_{offset}), & \text{if } s_j(t) = 1 \text{ and } x_i(t) > \epsilon \\ -\alpha^-(1 - y_j(t)), & \text{if } s_i(t) = 1 \text{ and } y_j(t) > \epsilon \end{cases} \quad (3)$$

where α^+ and α^- are the learning rates. This STDP rule works in combination with the introduction of hard bounds that restrict

synaptic weights to the range $0 \leq w_{ji} \leq 1$:

$$w_{ji} \leftarrow \min(1, \max(0, w_{ji} + \Delta w_{ji}))$$

2.3.1 First case of synaptic weight adaptation

The first adaptation case (Figure 3A) is (1) triggered eventwise by a postsynaptic spike indicated by $s_j(t = t_{post}) = 1$ and (2) operates locally in time since the presynaptic trace $x_i(t)$ is sampled at time $t = t_{post}$ and must be greater than ϵ , corresponding to a temporal window since the time of the presynaptic spike t_{pre} . Thus, postsynaptic spikes arriving outside this temporal window do not trigger the rule. This rule addresses the case of causal interactions, i.e., presynaptic spikes emitted before or at the time of the postsynaptic spike, i.e., $\Delta_t \geq 0$.

This first adaptation case integrates a vector quantization module aimed at learning the distribution of relative spike times in the distribution of synaptic weights, and thus minimizing the reconstruction error because the neural encoding is temporal. We can observe, by analyzing the equilibrium solution $\Delta w_{ji} = 0$, that the weight w_{ji} converges to a value dependent on $x_i(t)$ and thus on the time interval $\Delta_t = t_{post} - t_{pre}$ between pre- and

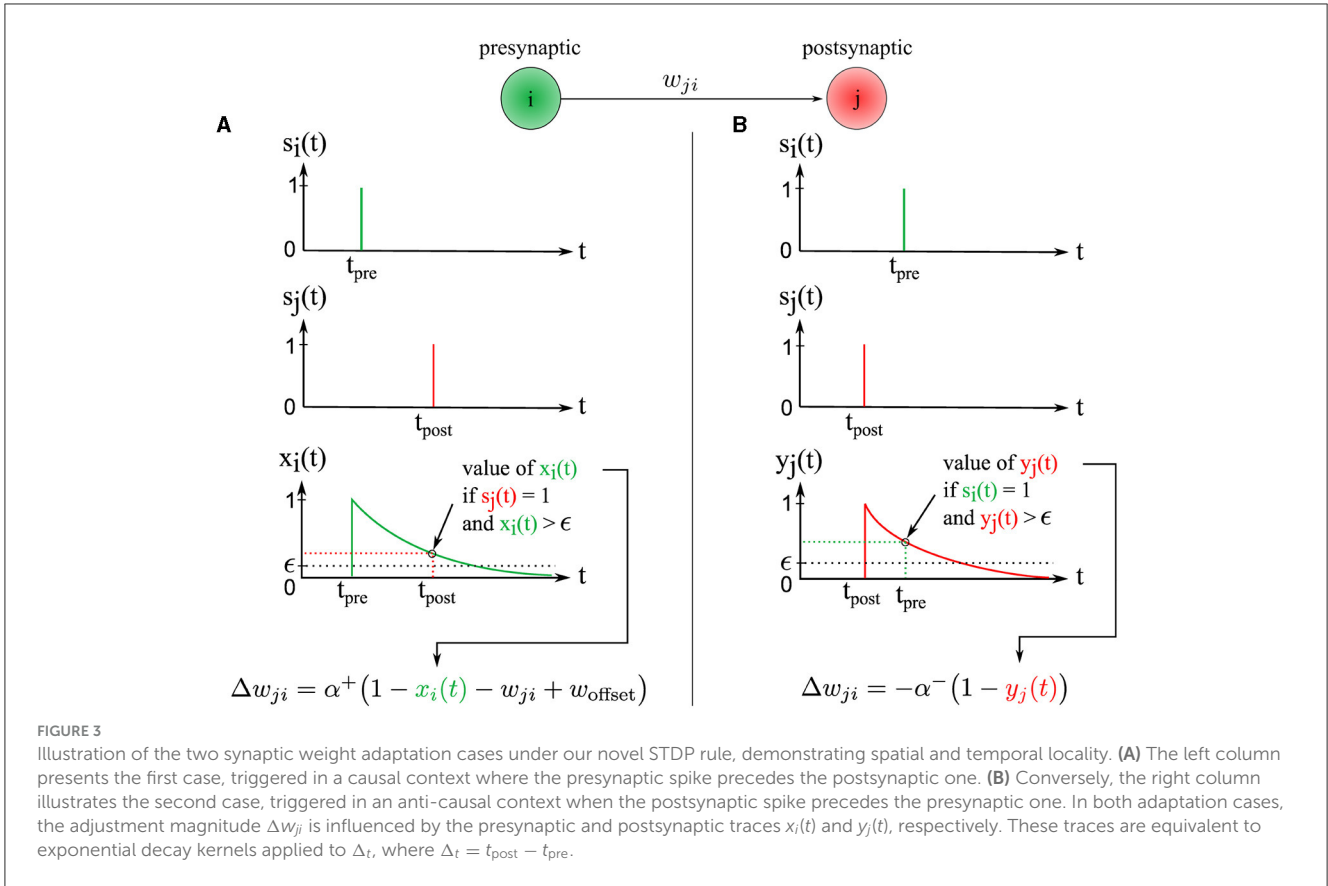


FIGURE 3

Illustration of the two synaptic weight adaptation cases under our novel STDP rule, demonstrating spatial and temporal locality. (A) The left column presents the first case, triggered in a causal context where the presynaptic spike precedes the postsynaptic one. (B) Conversely, the right column illustrates the second case, triggered in an anti-causal context when the postsynaptic spike precedes the presynaptic one. In both adaptation cases, the adjustment magnitude Δw_{ji} is influenced by the presynaptic and postsynaptic traces $x_i(t)$ and $y_j(t)$, respectively. These traces are equivalent to exponential decay kernels applied to Δt , where $\Delta t = t_{post} - t_{pre}$.

postsynaptic spikes:

$$\begin{aligned} \Delta w_{ji} &= 0 \\ \Leftrightarrow 0 &= 1 - x_i(t) - w_{ji} + w_{offset} \\ \Leftrightarrow w_{ji} &= 1 - x_i(t) + w_{offset} \\ \Leftrightarrow w_{ji} &= 1 - \exp\left(-\frac{t - t_{pre}}{\tau_x}\right) + w_{offset} \end{aligned}$$

As this adaptation case is triggered by a postsynaptic spike indicated by $s_j(t = t_{post}) = 1$, we finally obtain:

$$\Leftrightarrow w_{ji} = 1 - \exp\left(-\frac{\Delta t}{\tau_x}\right) + w_{offset} \quad (4)$$

Consider a spike pattern repeatedly presented to a postsynaptic neuron for illustration. The synaptic weight w_{ji} will converge to a value that depends on the presynaptic trace $x_i(t)$ and therefore on Δt . Recall that the trace $x_i(t)$ jumps to 1 upon the emission of a presynaptic spike, and then decays exponentially over time. Thus, the larger the value of $\Delta t \rightarrow \infty$, the smaller the value of $x_i(t) \rightarrow 0$. The first presynaptic spikes carry most of the information about the input, which can be interpreted as a hidden variable. These first spikes are reflected by a large Δt and thus a small $x_i(t) \in]0, 1]$. Therefore we introduce the term $1 - x_i(t)$ in the rule, which induces high synaptic weight values w_{ji} for the earliest firing presynaptic neurons, thereby amplifying the causal impact of the first presynaptic spikes on the postsynaptic neuron. Consequently, the first presynaptic

neuron that fires induces the highest weight value w_{ji} , while the last firing neuron induces the lowest weight value w_{ji} . This process unfolds for each presynaptic-postsynaptic neuron pair meeting the conditions of the first adaptation case. Thus, the order and latency of the spikes are represented in the resulting synaptic weight distribution w_{ji} (see Figure 4). By contrast, the traditional STDP rule (Song et al., 2000) maximally strengthens the latest spike ($t_{pre} \approx t_{post}$) for causal interactions and typically leads to weight saturation, where all weights saturate to 1.0 for instance.

We still need to address the role of the last term w_{offset} . When $\Delta t \rightarrow 0$, this is reflected by $x_i(t) \rightarrow 1$ and induces $1 - x_i(t) \rightarrow 0$ and thus $w_{ji} \rightarrow 0$. To avoid $w_{ji} \rightarrow 0$ when a presynaptic spike is emitted at a time $t_{pre} \approx t_{post}$, i.e., when $\Delta t \rightarrow 0$, we add a positive offset $w_{offset} > 0$ to the rule.

2.3.2 Second case of synaptic weight adaptation

The second adaptation case (Figure 3B) is triggered by a presynaptic spike indicated by $s_i(t = t_{pre}) = 1$, and operates locally in time because the postsynaptic trace $y_j(t)$ is sampled at time $t = t_{pre}$ and must be greater than ϵ , corresponding to a temporal window from the postsynaptic spike time t_{post} . Thus, presynaptic spikes falling outside of this temporal window do not trigger the rule. This rule handles the case of anti-causal interactions, i.e., presynaptic spikes emitted after or at the time of the postsynaptic spike, i.e., $\Delta t \leq 0$.

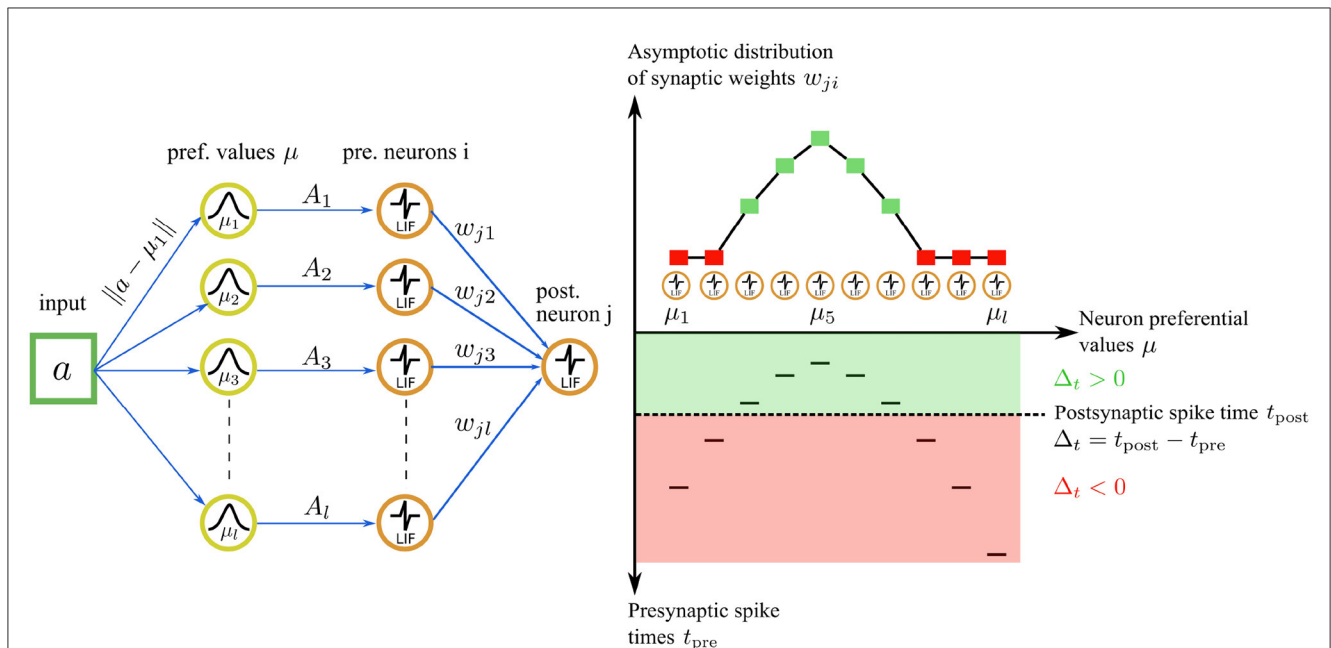


FIGURE 4 Illustration of the asymptotic weight distribution learned with our STDP rule for an input value of $a = 0.45$. **Left:** The input value is encoded by a population of $l = 10$ neurons in the encoding layer u . **Right bottom:** The neuron with the nearest preferential value μ_i to the input ($\mu_5 = 0.45$ in this case) fires a spike first, followed by the other neurons in the population. The spiking pattern is transmitted through synaptic connections to a postsynaptic neuron j in the representation layer. The postsynaptic neuron j integrates the received inputs and fires a spike at time t_{post} (dashed line). The sign of the time difference Δ_t between a postsynaptic and a presynaptic spike delimits two learning regimes (green and red areas) for the STDP rule. **Right top:** for $\Delta_t \geq 0$, the first adaptation case is triggered, resulting in an attractive fixed point weight. The asymptotic weight value w_{ji} increases as Δ_t increases (green squares). For $\Delta_t \leq 0$, the second adaptation case is triggered, leading to weight depression. The weight value w_{ji} decreases as Δ_t increases, tending asymptotically toward 0 (red squares).

We can reformulate this adaptation case to explicitly show the dependence on $\Delta_t = t_{\text{post}} - t_{\text{pre}}$:

$$\Delta w_{ji} \propto 1 - y_j(t)$$

$$\Delta w_{ji} \propto 1 - \exp\left(-\frac{t - t_{\text{post}}}{\tau_y}\right)$$

Since this adaptation case is triggered by a presynaptic spike indicated by $s_i(t = t_{\text{pre}}) = 1$, we have:

$$\Delta w_{ji} \propto 1 - \exp\left(-\frac{t_{\text{pre}} - t_{\text{post}}}{\tau_y}\right)$$

$$\Delta w_{ji} \propto 1 - \exp\left(+\frac{\Delta_t}{\tau_y}\right)$$

Δw_{ji} is proportional to $1 - \exp\left(\frac{\Delta_t}{\tau_y}\right)$, which is multiplied by $-\alpha^-$. Therefore, this second adaptation case depresses the synaptic weights. The postsynaptic trace $y_j(t)$ behaves similarly to the presynaptic trace $x_i(t)$, where it jumps to 1 upon emission of a postsynaptic spike and then decays exponentially over time. Thus, the larger the value of $\Delta_t \rightarrow \infty$, the smaller the value of $y_j(t) \rightarrow 0$. Late-arriving presynaptic spikes convey minimal information about the input. That is why we introduce $\Delta w_{ji} \propto 1 - y_j(t)$ into the rule: the later a presynaptic spike arrives relative to a postsynaptic spike, the greater the depression of w_{ji} . Thus, considering the asymptotic behavior of this synaptic depression case, the weight $w_{ji} \rightarrow 0$ (see Figure 4), as the lower bound of the

weight is fixed at 0 using a hard bound. This asymptotical behavior mirrors the traditional STDP rule (Song et al., 2000) for anti-causal interactions. However, the learning dynamics are reversed: the latest spike ($t_{\text{pre}} \gg t_{\text{post}}$) induces maximal weight depression, while close anti-causal interactions ($t_{\text{pre}} \approx t_{\text{post}}$) result in minimal weight depression.

2.3.3 Winner-Take-All circuit

Through afferent synaptic weight adjustments, postsynaptic neurons progressively adapt to distinct spatiotemporal patterns (see Figure 4). Additionally, akin to biological neurons, they emit spikes before receiving all presynaptic inputs, thus boosting the SNN's processing speed.

Remember that the first spikes reduce uncertainty about the input the most, which can be interpreted as a hidden variable. This feature is harnessed by a temporal Winner-Take-All circuit. In this circuit, the first neuron crossing its firing threshold $V_j(t) > V_\theta$ in response to a spatiotemporal input pattern is identified as the best pattern representative, termed the Spiking Best Matching Unit (SBMU). The SBMU strongly inhibits the other neurons in the representation layer, preventing them from firing and thus learning the current prototype.

This self-organizing competitive learning scheme is implemented by all-to-all lateral inhibitory connections. The circuit induces highly sparse activity in the representation layer and forces neurons to learn uncorrelated code vectors, two

important ingredients for efficient learning (Bengio et al., 2013; Falez et al., 2019).

However, for a given input vector, a pure WTA circuit only updates the code vector of a single neuron: the SBMU. Our aim is to speed up learning by recruiting more than one neuron (k-WTA) during the presentation of an input vector.

To achieve this goal, we introduce a homeostatic mechanism that increases the value of the lateral synaptic weights in the representation layer. Initially, lateral inhibition is minimal with lateral weights initialized at $w_{ji} = -c_{min}$. As learning progresses, the homeostatic mechanism raises these weights toward the target value of $w_{ji} \approx -c_{max}$. With appropriate values of c_{min} and c_{max} , the circuit can dynamically transition from a k-WTA to a WTA behavior.

The homeostatic mechanism progressively involves fewer neurons during the learning phase, enabling gradual decorrelation and specialization of the code vectors that neurons learn. The mechanism is described by the following equation:

$$\tau_w \frac{dw_{ji}}{dt} = -c_{max} - w_{ji} \quad (5)$$

where τ_w is the time constant of the homeostatic mechanism. We set τ_w to one-third of the total learning phase duration to ensure $w_{ji} \approx -c_{max}$ is reached by the end of the learning phase. Given an encoding time window of T and P input vectors, the time constant is then given by $\tau_w = 1/3 \times T \times P$.

3 Experimental protocol and results

3.1 Metrics for performance evaluation

We report experiments of representation learning with visual data in this section. A first natural objective is to evaluate the quality of visual representations learned by neurons in the representation layer v . For this, we use the Root Mean Square (RMS) reconstruction error between an input vector and the code vector of the associated SBMU, comparing the original and reconstructed image patches.

The Structural Similarity Index Measure (SSIM) is another measure that gauges the structural similarity between two images, rather than pixel-by-pixel difference as done by the Peak Signal-to-Noise Ratio (PSNR) measure based on RMS error. However, SSIM is not used in works related to ours, making it difficult to compare our performances. Furthermore, studies (Horé and Ziou, 2010; Dosselmann and Yang, 2011) have revealed analytical and statistical links between PSNR (based on RMS) and SSIM, indicating that their differences essentially stem from their sensitivity to image degradation. More generally, there is currently no satisfactory visual quality measure that fully captures human perception. Hence, in addition to the RMS reconstruction error used in our quantitative analysis, we provide a qualitative evaluation that visually examines the learned representations and the resulting reconstructed images.

Related works (Burbank, 2015; Tavanaei et al., 2018) use a correlation coefficient-based similarity measure for evaluating visual representation quality. However, we forgo this measure

due to its high sensitivity to outliers (Jenkin et al., 1991; Yen and Johnston, 1996), interpretational challenges (Lee Rodgers and Nicewander, 1988), and technical limitations, such as undefined coefficients for image patches with uniform intensity—a common occurrence in the MNIST database, particularly with pure white patches.

Beyond the quality of learned representations, we also evaluate additional SNN characteristics, such as the sparsity of activity in the representation layer and the Euclidean incoherence in the self-organized election process of the SBMU—a new measure that we introduce subsequently. The RMS reconstruction error, sparsity, and incoherence provide insights into the accuracy, efficiency, and self-organization capability of the SNN.

3.1.1 Mean squared error

We use the Root Mean Squared (RMS) error to quantify the difference between an input image patch \mathbf{a}_p and a reconstructed patch $\hat{\mathbf{a}}_p$ (6):

$$RMS = \frac{1}{P} \sum_{p=1}^P \sqrt{\frac{1}{k} \sum_{i=1}^k (a_{i,p} - \hat{a}_{i,p})^2} \quad (6)$$

Here, k is the number of pixels in a patch and P is the number of patches, and $a_{i,p}$ is the i th pixel of a_p .

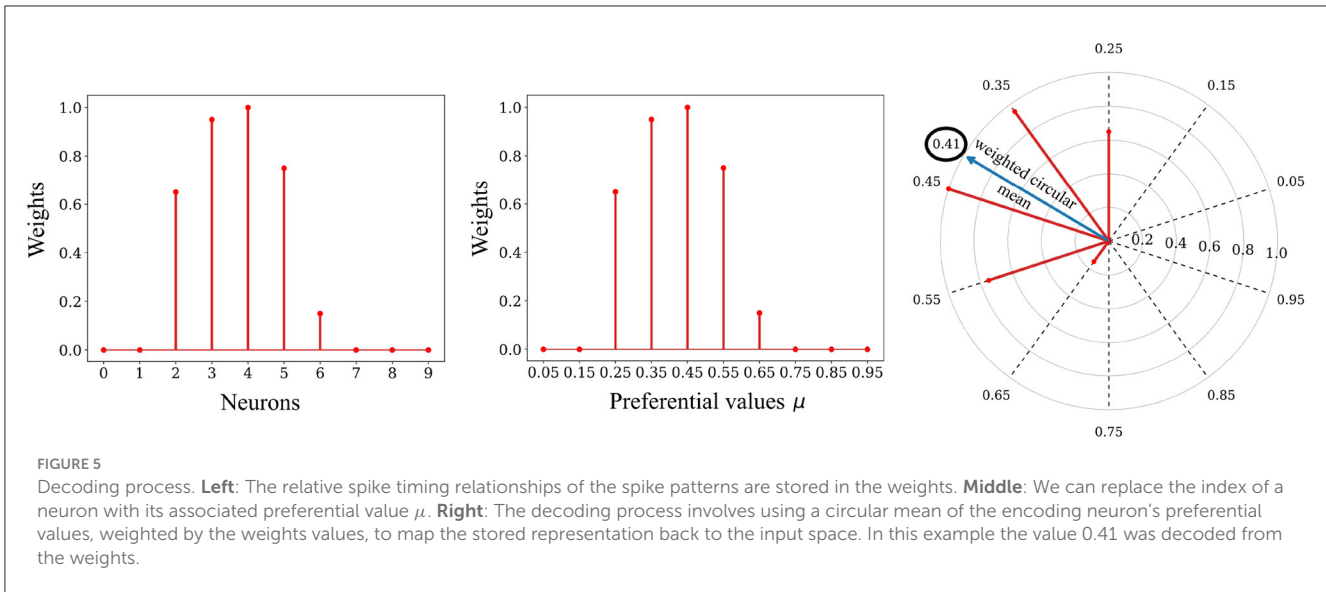
Recall that each dimension $i \in 1, 2, \dots, k$ of an input patch is distributed among l encoding neurons in the encoding layer. Each of the $z \in 1, 2, \dots, l$ encoding neurons has an associated preferential value $\mu_z \in \mu_1, \mu_2, \dots, \mu_l$. The encoding layer is fully connected to the representation layer in an all-to-all relationship. Each postsynaptic neuron in the representation layer is thus connected to $k * l$ presynaptic neurons in the encoding layer.

Each dimension $i \in 1, 2, \dots, k$ of the reconstructed patch $\hat{a}_{i,p}$ —corresponding in our case to the intensity of a pixel—is decoded from the distribution of l synaptic weights $w_1^i, w_2^i, \dots, w_l^i$ associated with it. The decoding method is based on the circular mean of neurons' preferred directions, weighted by their weight values, as exposed below (see Figure 5). The underlying principle is that the neural code exhibits unimodality (convexity) due to the presence of circular Gaussian receptive fields. This distribution of spike timings is mirrored in a unimodal synaptic weight distribution in the learned weights, a result of our STDP rule (see Figure 4). Therefore, we employed a circular weighted mean to decode the empirical mean. Interestingly, this decoding method has been successfully employed to decode the direction of arm movement from neuronal populations activity in the primate motor cortex (Georgopoulos et al., 1986).

The first step of the decoding process is to transform each preferential value $\mu_z \in \mu_1, \mu_2, \dots, \mu_l$ of the presynaptic neurons into an angle θ_z (preferred direction):

$$\theta_z = 2\pi \mu_z$$

We then calculate the weighted mean of the Cartesian coordinates \bar{x}_i and \bar{y}_i from the angles θ_z (angular coordinates) and



the synaptic weights w_z^i associated with pixel i of the SBMU (radial coordinates).

$$\begin{aligned} \text{with } w_z^i &= w_{\text{sbmu},(i-1)l+z} \\ \bar{x}_i &= \frac{1}{\Delta_i} \sum_{z=1}^l w_z^i \cos(\theta_z) \\ \bar{y}_i &= \frac{1}{\Delta_i} \sum_{z=1}^l w_z^i \sin(\theta_z) \\ \Delta_i &= \sum_{z=1}^l w_z^i \end{aligned}$$

We calculate a new angle $\bar{\theta}_i$ from the Cartesian coordinates \bar{x}_i and \bar{y}_i . This angle $\bar{\theta}_i$ corresponds to the weighted circular mean:

$$\bar{\theta}_i = \text{atan2}(-\bar{y}_i, -\bar{x}_i) + \pi$$

We end with an inverse transformation, transforming the angle $\bar{\theta}_i \in [0, 2\pi]$ into the normalized intensity of a pixel in the reconstructed patch $\hat{a}_{i,p} \in [0, 1]$:

$$\hat{a}_{i,p} = \frac{\bar{\theta}_i}{2\pi}$$

3.1.2 Sparsity

Sparsity is not evaluated on the encoding layer but on the output layer, i.e., the representation layer. Sparsity corresponds to the percentage of active neurons for an input vector during the encoding time window T :

$$\text{Sparsity} = \frac{1}{m} N_{\text{imp}} \tag{7}$$

where m is the number of neurons, and N_{imp} is the number of spikes emitted by the m neurons in response to an input vector over the time window T . A low value indicates a large number of inactive neurons and therefore a high sparsity.

3.1.3 Coherence in the election of the SBMU

The Best Matching Unit (BMU) in traditional Vector Quantization models like the Self-Organizing Map (Kohonen, 2013), is selected each iteration using global network information. However, in our SNN, the Spiking Best Matching Unit (SBMU) selection is a dynamic and self-organized process influenced by neuron competition to represent the current input.

To assess the coherence of this self-organized SBMU selection compared to a Euclidean distance-based selection, we introduce a measure of incoherence. This measure evaluates if the SBMU is among the best representatives minimizing the Euclidean distance between the reconstructed patch from the SBMU and the input.

Firstly, we define a function d which, for a given input vector, calculates the normalized Euclidean distance between the input vector and the associated code vector of a neuron indexed $j \in 1, \dots, m$ in the representation layer:

$$d: 1, \dots, m \rightarrow [0, 1]$$

Then, all distances $d(j)$ are computed, and pairs $[j, d(j)]$ for $j = 1, \dots, m$ are sorted in increasing distance order, resulting in an ordered list $[j_p, d(j_p)]$, where $p = 1, \dots, m$, and j_p denotes a pair index in the list:

$$\forall p < m \quad d(j_{p+1}) \geq d(j_p) \quad \text{and} \quad \{j_p | p = 1, \dots, m\} = \{1, \dots, m\}$$

Next, we assess if the index of the selected SBMU lies within the top $x\%$ of the ordered list. If yes, we increment the coherent SBMU count (sc), otherwise the paradoxical SBMU count (sp) is incremented. This check is repeated for each input vector to classify the elected SBMU as coherent or paradoxical. The top $x\%$ of the list sets the tolerance threshold for distinguishing between a coherent and paradoxical SBMU.

$$\begin{aligned} \text{if } \text{sbmu} \in j_p \mid p \in 1, \dots, \lceil mx \rceil, \quad \text{then } sc &\leftarrow sc + 1 \\ \text{else } sp &\leftarrow sp + 1 \end{aligned}$$

TABLE 1 W-TCRL parameters used in all simulations.

Neuronal parameters						
dt	τ_m^u	τ_m^v	V_θ^u	V_θ^v	T_{refrac}^u	T_{refrac}^v
0.1 ms	10.0 ms	1.4 ms	0.5	$0.25 kl$	6 ms	6 ms
Synaptic parameters						
$\tau_f(u \text{ to } v)$	$\tau_f(v \text{ to } v)$	τ_x	τ_y			
2.8 ms	0.3 ms	1.3 ms	4.3 ms			
STDP rule parameters						
α^+	α^-	w_{offset}	ϵ			
0.004	0.024	0.2	0.1			
Homeostatic mechanism parameters						
τ_w	c_{min}	c_{max}				
$1/3 TP$	$9 V_\theta^v$	$91 V_\theta^v$				

Lastly, the incoherence measure is defined, with sc and sp denoting the total number of coherent and paradoxical SBMU, respectively:

$$\text{Incoherence} = 1 - \frac{sc}{sc + sp}$$

High incoherence may for instance potentially lead to a high root mean square (RMS) reconstruction error.

3.2 Parameters of W-TCRL

A common challenge in training spiking neural networks (SNNs) is their limited adaptability to varying data dimensionalities, often requiring extensive hyperparameter tuning.

To address this issue, we propose a generic parametrization approach for the W-TCRL model that can handle arbitrary data dimensionalities, denoted as k . Only three parameters in the representation layer, namely the neuron firing threshold V_θ^v , the initial lateral inhibition level c_{min} , and the final inhibition level c_{max} , depend on the data dimensionality. For simplicity, we assume a linear relationship between the data dimensionality k and these three parameters.

In our parametrization, each encoder neuron emits a single spike to represent a continuous input, and the total number of spikes in the encoding layer is proportional to the data dimensionality k , given by $k \times l$, where l represents the number of neurons used to represent each dimension of the input space. As spikes are weighted by synaptic weights within the range of $[0, 1]$, a neuron can only fire if its firing threshold V_θ^v is set to $c \times k \times l$, where c is a coefficient ranging from 0 to 1.

Additionally, to maintain a Winner-Take-All (WTA) behavior in the representation layer, it is crucial to calibrate the lateral inhibition level based on the data dimensionality. This ensures that the inhibition level is sufficiently high to prevent other neurons from reaching their firing threshold V_θ^v .

We determined the coefficients of the linear equations relating these three parameters to the input data dimensionality k and

the number of encoder neurons l using the Bayesian optimization method known as Tree-structured Parzen Estimator (TPE) (Bergstra et al., 2011). Our objective was to minimize the root mean square (RMS) reconstruction error on the MNIST and natural image datasets, considering input vectors with varying dimensions. The aim was to obtain generic and robust coefficients capable of handling diverse input data distributions and dimensionalities.

By incorporating the optimized coefficient values into the linear equations, the values of the three hyperparameters can be determined as follows:

$$V_\theta^v = 0.25kl \quad (8)$$

$$c_{min} = 9V_\theta^v \quad (9)$$

$$c_{max} = 91V_\theta^v \quad (10)$$

The parameters of W-TCRL used in all simulations are given in Table 1.

3.3 Results for MNIST

Experiments were run with varying network sizes with $m \in \{16, 32, 64, 128, 256\}$ neurons in the representation layer and therefore m code vectors submitted to learning. Each experiment was evaluated with 30 independent simulations. For a fair comparison with the state of the art set by Tavanaei et al. (2018), we used the same experimental protocol, including the lack of cross-validation.

Synaptic weights between the encoding layer u and the representation layer v are randomly initialized in the interval $[0.6, 0.8]$. The inputs are normalized within the interval $[0.15, 0.85]$. We introduce a margin due to the projection of linear input data onto a circular space (receptive fields) where extreme values become equivalent (2π is equivalent to 0 radians). In addition, MNIST is essentially composed of extreme values (black and white pixels).

For training and testing, subsets of 15,000 and 1,000 handwritten digits were respectively used. These subsets provided

5×5 pixel patches extracted from 28×28 pixel digits as SNN inputs. The dimension-dependent parameters V_{θ}^v , c_{min} , c_{max} were automatically determined for $5 \times 5 = 25$ dimensions by Equations (8), (9), (10).

In the training phase, 60,000 image patches were utilized, while the test phase held the lateral inhibition in the representation layer at its maximum (c_{max}), with plasticity disabled.

Performance was assessed based on the three aforementioned metrics: the RMS reconstruction error between an input vector and the code vector of the SBMU, the sparsity of the representation layer activity, and the incoherence in the self-organized process of SBMU election.

3.3.1 Learning phase

Figure 6 shows that the different performance measures consistently enhance with the progression of learning

iterations, and also with an increasing number of neurons $m \in \{16, 32, 64, 128, 256\}$ in the representation layer v . The RMS reconstruction error decreases, indicating that W-TCRL learns to compress the distribution of input data, reaching a relatively stable plateau after 8,000 iterations with an RMS error of about 0.08 for $m \in \{64, 128, 256\}$ neurons. The sparsity also decreases, indicating that the percentage of neurons firing in the representation layer v decreases in response to an input vector. This is attributed to heightened lateral inhibition through homeostasis and increased neuronal specialization. Lastly, Euclidean incoherence in the SBMU election process reduces through training iterations for a decision threshold x at 5 and 10%, signaling fewer paradoxical SBMUs and improved neuronal receptive field selectivity for specific input regions. This shows that the first firing neuron is, on average, an accurate representative of the input spike pattern.

Figure 6 shows that the code vectors in the representation layer gradually become selective to

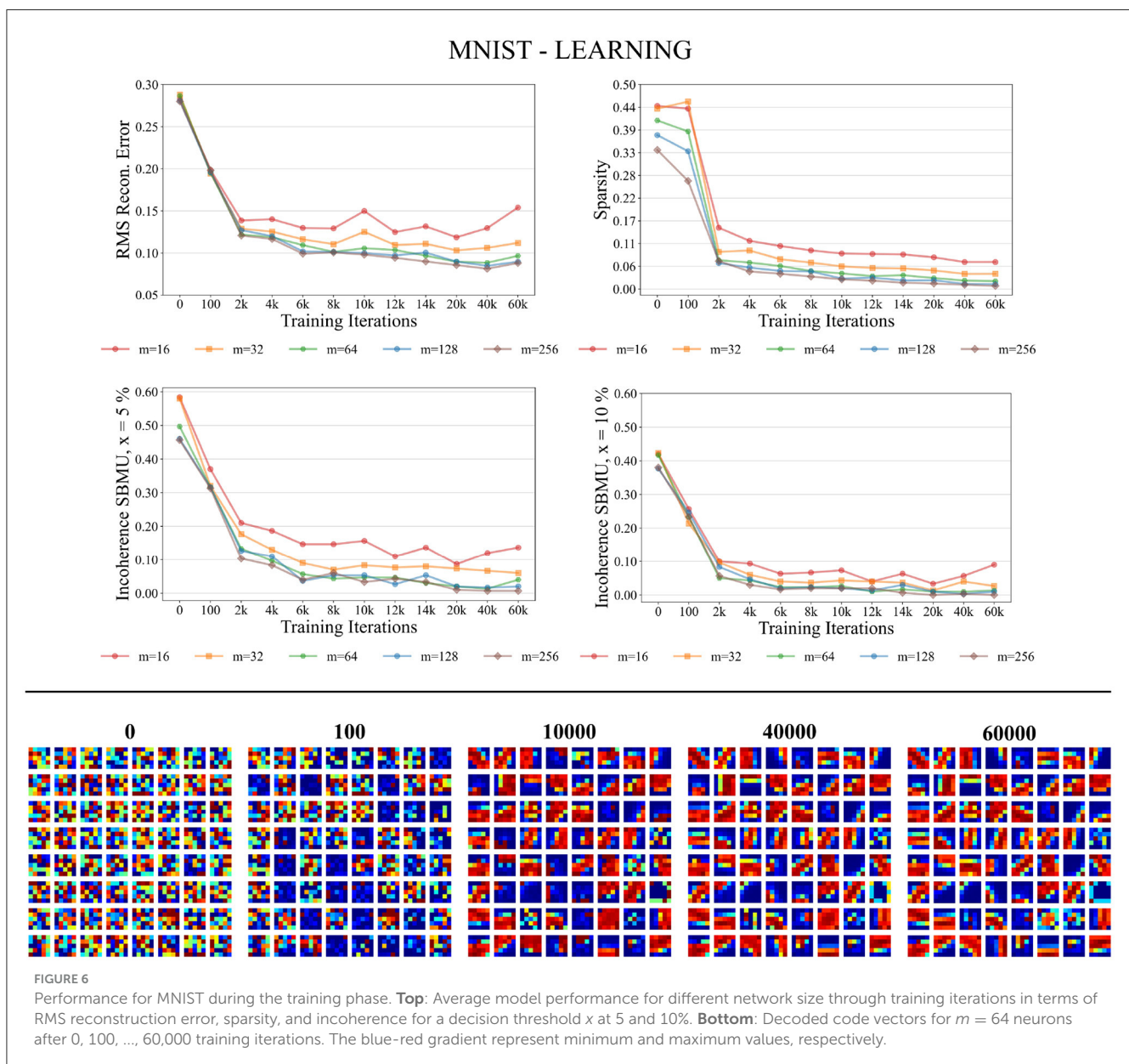
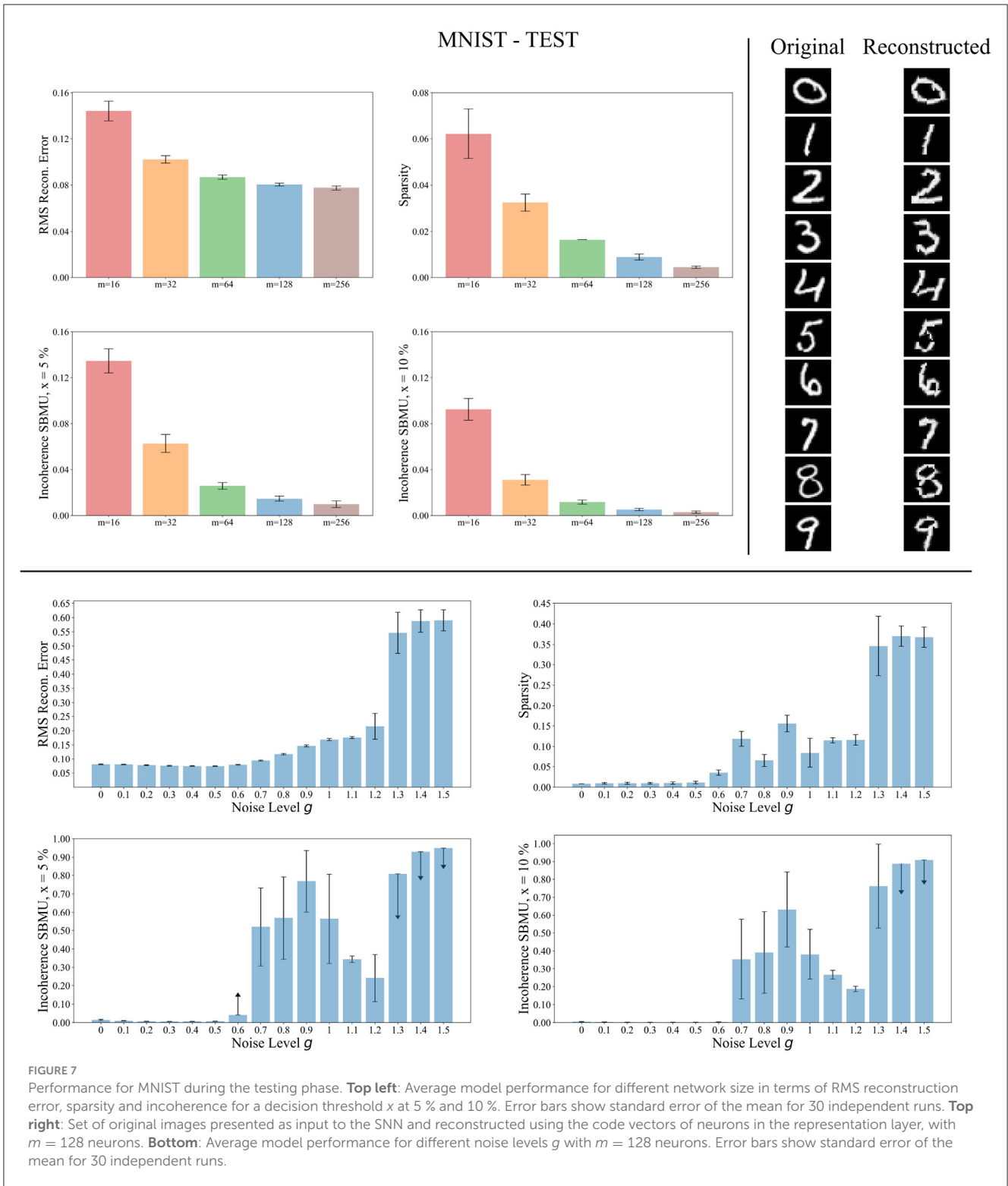


FIGURE 6 Performance for MNIST during the training phase. **Top:** Average model performance for different network size through training iterations in terms of RMS reconstruction error, sparsity, and incoherence for a decision threshold x at 5 and 10%. **Bottom:** Decoded code vectors for $m = 64$ neurons after 0, 100, ..., 60,000 training iterations. The blue-red gradient represent minimum and maximum values, respectively.



various visual orientations over the course of the learning iterations.

3.3.2 Testing phase

We now evaluate the performance of W-TCRL on the test dataset for MNIST. The achieved performances all increase with

the capacity of the SNN (see Figure 7). The average sparsity for $m = 256$ neurons reaches a value of 0.004, meaning that only 0.4% of the neurons in the representation layer are active on average. Self-organized SBMU selection is achieved with up to 99.0% of coherent SBMU for a decision threshold of 5% and $m = 256$ neurons. The best RMS reconstruction error reaches the value of 0.08 (see Table 2). This represents a relative improvement of 53%

TABLE 2 Average results on 30 independent runs obtained on MNIST and natural images datasets for a given number of neurons $m \in \{16, 32, 64, 128, 256\}$.

Dataset	MNIST					Natural images				
	16	32	64	128	256	16	32	64	128	256
RMS	0.144	0.102	0.087	0.080	0.078	0.164	0.138	0.061	0.056	0.056
Sparsity	0.062	0.032	0.016	0.009	0.004	0.255	0.234	0.141	0.076	0.043
Incoherence 5 %	0.134	0.063	0.026	0.015	0.010	0.516	0.473	0.361	0.275	0.224
Incoherence 10 %	0.092	0.031	0.012	0.005	0.003	0.378	0.344	0.271	0.176	0.064

TABLE 3 Best RMS reconstruction errors reported by Tavanaei et al. (2018) are compared to our model W-TCRL.

Dataset	Tavanaei	W-TCRL (our model)
MNIST	0.17	0.08
Natural images	0.24	0.06

Bold values represent the performances of our model.

compared to the SNN state-of-the-art (see Table 3) for learning representations using the MNIST dataset.

The reconstruction of images from the self-organized elected code vectors found by W-TCRL produces well-reconstructed handwritten digits, comparable to the original images (see Figure 7). The quality of the reconstructed digits demonstrates the effectiveness of our SNN architecture in capturing and representing the visual information accurately.

We also tested the robustness of our network's ability to learn under noise (see Figure 7). During the training phase, we varied the white noise scaling factor g in the range $[0, 1.5]$, resulting in spike jitter up to around 5 ms in spikes outputted by the encoding layer. The performance remains consistent across the different metrics for values of g up to 0.5, demonstrating that noise doesn't prevent neurons in the representation layer from learning statistical correlations from input spikes. However, with $g > 1.2$, the noise becomes too strong. It destroys the temporal information contained in the spike pattern. Consequently, the neurons fail to extract meaningful representations (high RMS reconstruction error), tend to fire simultaneously (low sparsity), and lose their selectivity (high incoherence).

3.4 Results for natural images

As for MNIST, experiments for the natural images dataset were conducted for several network sizes with $m \in \{16, 32, 64, 128, 256\}$ neurons in the representation layer and thus m code vectors subject to learning. Each experiment was evaluated with 30 independent simulations.

The synaptic weights between the encoding layer u and the representation layer v are randomly initialized in the interval $[0.6, 0.8]$. Input normalization was performed to scale inputs within the interval $[0.05, 0.95]$. 16×16 pixel patches, extracted from natural images of 512×512 pixels, are provided as input vectors

to the SNN. The dimension-dependent parameters V_{θ}^v , c_{\min} , c_{\max} were automatically determined for $16 \times 16 = 256$ dimensions using Equations (8), (9), (10). A total of 60,000 image patches are provided to the SNN during the training phase. In the test phase, lateral inhibition in the representation layer v was set to its maximum value, c_{\max} , and plasticity was disabled.

Similarly to the previous section, we evaluated the performance of W-TCRL using the same three metrics: the root mean squared (RMS) reconstruction error, the sparsity of the representation layer activity, and the Euclidean incoherence of the SBMU.

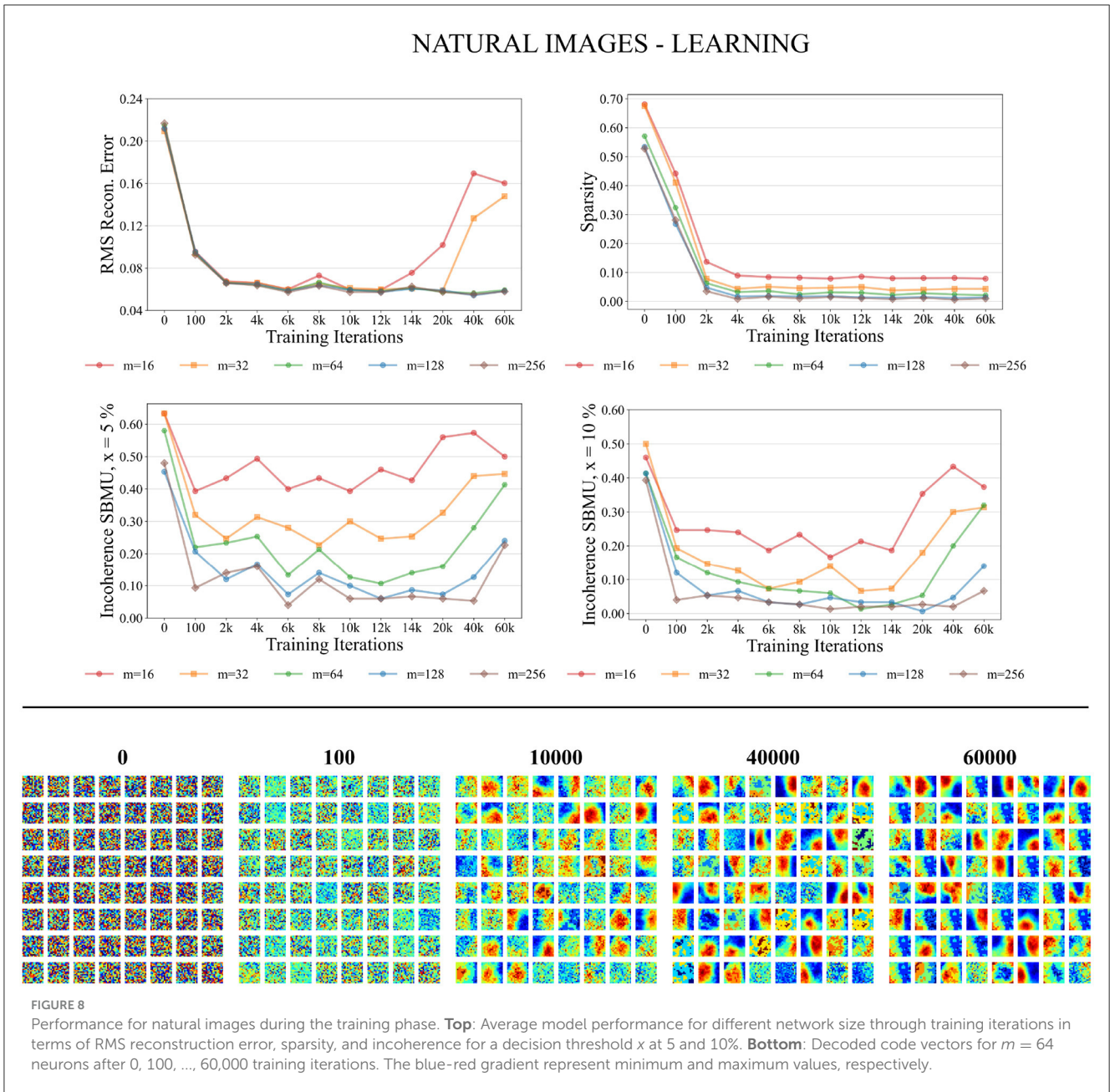
3.4.1 Training phase

As depicted in Figure 8 the performance measures for the natural images dataset in the training process consistently improve as the number of neurons in the representation layer increases. When the network capacity is too small, specifically with 16 and 32 neurons in the representation layer, the RMS reconstruction error is significantly degraded. In the case of higher network capacities, specifically with 64, 128, and 256 neurons in the representation layer, the RMS reconstruction error exhibits a consistent decrease throughout the training process. After approximately 6,000 iterations, the RMS error reaches a stable plateau, converging to a value of approximately 0.06. The sparsity of the representation layer activity decreases over the course of training, reaching a plateau after 6,000 iterations. The euclidian incoherence of the SBMU election decreases when using a decision threshold of 5% and 10%, but the euclidian incoherence exhibits more fluctuations compared to the results obtained for the MNIST dataset. This observation can be attributed to the high dimensionality of the input data (256 dimensions), which causes the Euclidean distances to become concentrated and thus difficult to distinguish. This hypothesis is further supported by a low RMS reconstruction error and its low fluctuation.

Figure 8 demonstrates that the code vectors in the representation layer exhibit an increasing selectivity toward visual patterns as the number of training iterations increases.

3.4.2 Testing phase

We now evaluate the performance of W-TCRL on the test dataset for natural images. The obtained performance increases as the capacity of the SNN increases, i.e., the number of neurons $m \in \{16, 32, 64, 128, 256\}$ in the representation layer



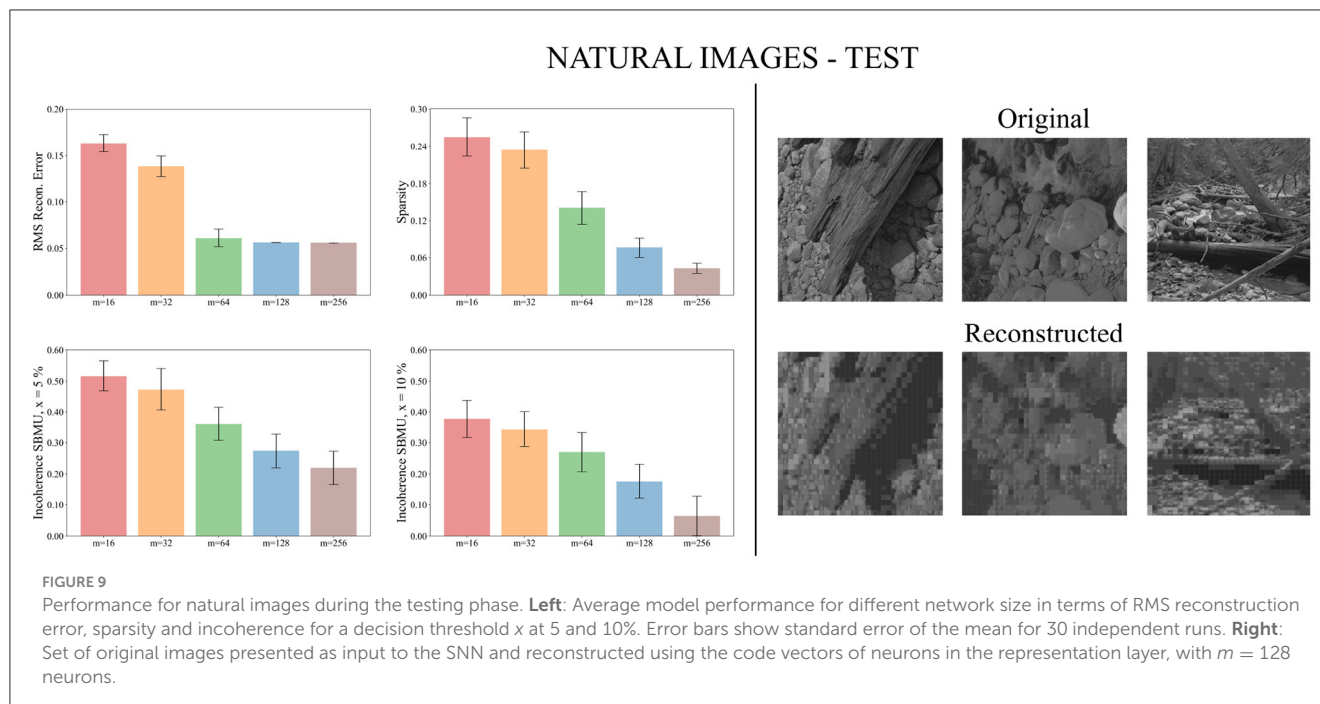
(see Figure 9). The average sparsity for $m = 256$ neurons reaches a value of 0.04 indicating that only 4% of the neurons in the representation layer are active on average. The self-organized election of the SBMU is achieved with up to 95.6% of coherent SBMU for a decision threshold of 10% and $m = 256$ neurons. The RMS reconstruction error reaches a plateau at 0.06 for $m \in \{64, 128, 256\}$ neurons (see Table 2). This represents a relative improvement of 75% compared to the SNN state of the art (see Table 3) for this dataset of natural images.

Figure 9 demonstrates that the self-organized elected code vectors of W-TCRL yield reconstructed images that closely resemble the original images (though the visual quality inherently suffers from large sizes of patches, regardless of the model used for representation learning).

4 Discussion

In contrast to rate-based encoding methods, which require up to 255 spikes for an 8-bit pixel value (King et al., 2013; Burbank, 2015; Tavanaei et al., 2018), our approach employs latency-based encoding with population coding (Ebitz and Hayden, 2021). This method uses 10 neurons per input coordinate, each emitting a single spike, resulting in a sparse and efficient encoding with only 10 spikes.

Our encoding method diverges from the widely used Time-To-First-Spike (TTFS) coding, which necessitates knowledge of stimulus onset timing. This information is available to the experimenter but not the neurons, which makes it biologically implausible. In contrast, our use of population coding embeds information within the spike pattern itself, specifically in the



timing relative to other neurons in the population. This allows downstream neurons to process these spike patterns in an event-based manner. More generally, population-based latency coding can be viewed as an extension of TTFS. It provides a more flexible representation by distributing information across multiple dimensions (population of neurons) rather than one (single neuron). This enhances flexibility and transmission speed with minimal extra spikes. Although population-based latency coding does not require an external time reference, it does need an internal one, similar to rate coding which needs a temporal window for averaging spikes. As Zhou et al. (2012) noted, the brain may generate several internal time references, including large-scale oscillations (Hopfield, 1995), local field potentials (Eggermont, 1998), and population onsets (Stecker and Middlebrooks, 2003; Chase and Young, 2007).

One limitation of latency-coding is its sensitivity to noise (Guo et al., 2021). This issue is not exclusive to this coding scheme. All neural codes relying on precise spike timing are vulnerable to noise and jitter. A potential solution is to pool information from a neuron population, as we have done, instead of relying on a single neuron as in TTFS. Our experimental results demonstrate that the combination of population-based latency coding with our new STDP rule makes the model quite robust in learning from noisy input spike patterns. The correlation among a neuron population can establish an error-correcting code that mitigates noise effects. This is because the spike timing jitter in relation to other neurons could be less than the jitter referenced to the onset of the stimulus (Zhou et al., 2012).

The extraction of representations from a temporal code lacked event-based local learning rules aimed at achieving low reconstruction error. Our work bridges this gap by introducing a novel STDP rule that extracts spatio-temporal centroids. This simple rule enables stable learning of the distribution of relative latencies within the synaptic weight distribution.

Unlike backpropagation algorithms, which require costly global information transport and significant memory for sequence history storage, our STDP rule operates on an event-based and local level in both time and space. The event-based and local operations of our STDP rule align well with the principles of asynchrony and locality inherent in neuromorphic processors (Davies et al., 2018). This makes our STDP rule an ideal fit for such implementations.

Early presynaptic spikes, carrying the most input information, trigger the first (causal) case of weight adaptation on postsynaptic spikes, leading to synaptic weights $w_{ji} > 0$. The learned weight w_{ji} is a function of the latency $\Delta_t = t_{post} - t_{pre}$ between the pre and postsynaptic spikes. Earlier spikes result in higher weights and have a greater impact on postsynaptic spike emission. This differs from the classical STDP rule, which maximizes weight strengthening for late-arriving spikes ($t_{pre} \approx t_{post}$) and usually leads to weight saturation rather than converging to a stable value. Later presynaptic spikes, containing less information, trigger the second (anti-causal) case of weight adaptation, causing synaptic depression ($w_{ji} \rightarrow 0$). Through these adaptations, a postsynaptic neuron learns a code vector or filter in its afferent synaptic weights. This facilitates rapid decision-making before receiving all presynaptic spikes, thereby accelerating the SNN's processing speed and mirroring biological neurons.

Rank-order TTFS coding (Thorpe and Gautrais, 1998) follow a similar approach to build detectors, where the weights decrease linearly with the spike rank. The synaptic weights are M (maximum) for the first spike, then $M-1, \dots, 1$. This approach focuses on the spike order rather than their precise timings, contrary to the basis of our approach. An accompanying modulation function emphasizes the importance of earliest spikes. The conjunction of a set of weights and a modulation function allows to build accurate detectors of spike patterns (Bonilla et al., 2022). This modulation function concept can be related to other TTFS coding works that utilize precise spike timing for supervised learning

tasks (Rueckauer and Liu, 2018; Zhang et al., 2022; Sakemi et al., 2023). These studies enhance the impact of the earliest spike by linearly increasing the postsynaptic potential (PSP) over time. Our model naturally achieves a similar effect through the CuBa LIF model. The PSP shape of the CuBa LIF is determined by the ratio τ_f/τ_m of the synapse's exponential decay time constant to that of the neuron (Göltz et al., 2021). By selecting $\tau_f > \tau_m$ as done in our experiments, the earliest spikes have a greater impact on the postsynaptic neuron response due to their influence being integrated over a longer period.

To implement competitive learning among neurons in the representation layer, our SNN incorporates lateral inhibitory connections. During the learning phase, a mechanism is employed to initially recruit multiple neurons, which accelerates the convergence of the SNN. Subsequently, the number of recruited neurons gradually decreases to promote specialization and decorrelation of the learned code vectors. This behavior is achieved through a homeostatic mechanism that increases the magnitudes of inhibitory lateral weights. As a result, the circuit transitions dynamically from a k-winners-take-all (k-WTA) to a winner-take-all (WTA) behavior, thereby increasing the sparsity of activity in the representation layer.

While representation learning in ANNs is a mature field, achieving good performance with SNNs remains challenging. Our W-TCRL model, based on temporal coding and a novel STDP rule, outperforms the state-of-the-art of SNNs using a rate-based coding approach by Tavanaei et al. (2018) in terms of RMS reconstruction error. Our method exhibits a relative improvement of 53% for MNIST and 75% for natural images. The reconstructed images exhibit a good visual quality compared to the original images.

The representation layer exhibits a highly sparse activity during inference, a desirable feature (Frenkel, 2021). For instance, on the MNIST dataset with $m = 256$ neurons, the average sparsity is 0.004, indicating that only 0.4% of neurons in the representation layer fire. Notably, neurons in the representation layer fire only once, distinguishing our approach from rate coding methods. In the work of Tavanaei et al. (2018), sparsity was averaged over a time window of $T = 40$ time steps for a given number of neurons m . The reported best sparsity of 9% implies that, on average, 9% of neurons fire at each time step within the encoding window T . In contrast, in our temporal coding-based SNN, a sparsity of 0.4% means that 0.4% of neurons fire a single time during the encoding window T . Therefore, based on the best reported sparsity by Tavanaei et al. (2018), a neuron fires an average of $0.09 * 40 = 3.6$ spikes during the temporal window T . In contrast, based on our best sparsity results, a neuron emits an average of 0.004 spikes during the temporal window T . Thus, our method achieves up to 900 times fewer spikes emitted per neuron in the representation layer on average, providing substantial benefits in terms of energy consumption and bandwidth in spike communication protocols such as AER, while improving the RMS reconstruction error. This empirical evidence confirms the benefits of using temporal codes compared to rate coding.

Lastly, the self-organized selection of the Spiking Best Matching Unit (SBMU) in our SNN achieves up to 99.0% coherent SBMUs (for MNIST with $m = 256$ neurons and a decision threshold x set to

5%). The high coherence of the SBMUs indicates that the neurons in the representation layer have developed a high level of selectivity for specific regions within the spatio-temporal input space. This selectivity results in an effective clustering of the spatio-temporal input space.

Future research should prioritize implementing our spiking representation learning model on neuromorphic processors (Davies et al., 2018) to enable real-time, low-power, and high-throughput processing, taking advantage of their parallelism and efficiency. Additionally, exploring the unsupervised extraction of hierarchical representations from sensory data by stacking multiple layers using our STDP rule can uncover more complex and abstract features in self-organizing SNN architectures. This unsupervised approach is exemplified by the HOTS architecture (Lagorce et al., 2017), which employs a three-layer stack for unsupervised feature extraction, with the final layer's output fed to a simple histogram classifier that achieves near 100 % accuracy on DVS datasets.

Data availability statement

Publicly available datasets were analyzed in this study. These data can be found at: MNIST: <http://yann.lecun.com/exdb/mnist/>; Natural Images: <http://www.rctn.org/bruno/sparsenet/>.

Author contributions

AF was the originator of the W-TCRL model. AF and BG developed its formalized expression and its analysis. AF carried out all simulations, and wrote the manuscript with contributions of BG. Both authors contributed to the article and approved the submitted version.

Funding

This work has been supported by ANR project SOMA ANR-17-CE24-0036.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828. doi: 10.1109/TPAMI.2013.50
- Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). “Algorithms for hyperparameter optimization,” in *Advances in Neural Information Processing Systems 24*, eds J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger (Granada: Curran Associates Inc.), 2546–2554.
- Billings, G., and van Rossum, M. C. W. (2009). Memory retention and spike-timing-dependent plasticity. *J. Neurophysiol.* 101, 2775–2788. doi: 10.1152/jn.91007.2008
- Bonilla, L., Gautrais, J., Thorpe, S., and Masquelier, T. (2022). Analyzing time-to-first-spike coding schemes: a theoretical approach. *Front. Neurosci.* 16, 971937. doi: 10.3389/fnins.2022.971937
- Burbank, K. S. (2015). Mirrored STDP implements autoencoder learning in a network of spiking neurons. *PLoS Comput. Biol.* 11, e1004566. doi: 10.1371/journal.pcbi.1004566
- Chase, S. M., and Young, E. D. (2007). First-spike latency information in single neurons increases when referenced to population onset. *Proc. Natl. Acad. Sci. U.S.A.* 104, 5175–5180. doi: 10.1073/pnas.0610368104
- Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359
- Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9, 99. doi: 10.3389/fncom.2015.00099
- Dosselmann, R., and Yang, X. D. (2011). A comprehensive assessment of the structural similarity index. *Signal Image Video Process.* 5, 81–91. doi: 10.1007/s11760-009-0144-1
- Ebitz, R. B., and Hayden, B. Y. (2021). The population doctrine in cognitive neuroscience. *Neuron* 109, 3055–3068. doi: 10.1016/j.neuron.2021.07.011
- Eggermont, J. J. (1998). Azimuth coding in primary auditory cortex of the cat. II. Relative latency and interspike interval representation. *J. Neurophysiol.* 80, 2151–2161.
- Falez, P., Tirilly, P., Bilasco, I. M., Devienne, P., and Boulet, P. (2019). Unsupervised visual feature learning with spike-timing-dependent plasticity: how far are we from traditional feature learning approaches? *Pattern Recogn.* 93, 418–429. doi: 10.1016/j.patcog.2019.04.016
- Frenkel, C. (2021). Sparsity provides a competitive advantage. *Nat. Mach. Intell.* 3, 742–743. doi: 10.1038/s42256-021-00387-y
- Furber, S. (2016). Large-scale neuromorphic computing systems. *J. Neural Eng.* 13, 051001. doi: 10.1088/1741-2560/13/5/051001
- Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science* 233, 1416–1419.
- Göltz, J., Kriener, L., Baumbach, A., Billaudelle, S., Breitwieser, O., Cramer, B., et al. (2021). Fast and energy-efficient neuromorphic deep learning with first-spike times. *Nat. Mach. Intell.* 3, 823–835. doi: 10.1038/s42256-021-00388-x
- Guo, W., Fouda, M. E., Eltawil, A. M., and Salama, K. N. (2021). Neural coding in spiking neural networks: a comparative study for robust neuromorphic systems. *Front. Neurosci.* 15, 638474. doi: 10.3389/fnins.2021.638474
- Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature* 376, 33–36.
- Horé, A., and Ziou, D. (2010). “Image quality metrics: PSNR vs. SSIM,” in *20th International Conference on Pattern Recognition, ICPR 2010 (Istanbul: IEEE)*, 2366–2369.
- Jenkin, M. R. M., Jepson, A. D., and Tsotsos, J. K. (1991). Techniques for disparity measurement. *CVGIP Image Understand.* 53, 14–30.
- Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., and Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* 99, 56–67. doi: 10.1016/j.neunet.2017.12.005
- Kheradpisheh, S. R., and Masquelier, T. (2020). Temporal backpropagation for spiking neural networks with one spike per neuron. *Int. J. Neural Syst.* 30, 2050027:1–2050027:16. doi: 10.1142/S0129065720500276
- King, P. D., Zylberberg, J., and DeWeese, M. R. (2013). Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. *J. Neurosci.* 33, 5475–5485. doi: 10.1523/JNEUROSCI.4188-12.2013
- Kohonen, T. (2013). Essentials of the self-organizing map. *Neural Netw.* 37, 52–65. doi: 10.1016/j.neunet.2012.09.018
- Lagorce, X., Orchard, G., Gallupi, F., Shi, B. E., and Benosman, R. B. (2017). HOTS: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1346–1359. doi: 10.1109/TPAMI.2016.2574707
- Lee Rodgers, J., and Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *Am. Stat.* 42, 59–66.
- Lee, C., Sarwar, S. S., Panda, P., Srinivasan, G., and Roy, K. (2020). Enabling spike-based backpropagation for training deep neural network architectures. *Front. Neurosci.* 14, 119. doi: 10.3389/fnins.2020.00119
- Mozafari, M., Kheradpisheh, S. R., Masquelier, T., Nowzari-Dalini, A., and Ganjtabesh, M. (2018). First-spike-based visual categorization using reward-modulated STDP. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 6178–6190. doi: 10.1109/TNNLS.2018.2826721
- Patel, D., Hazan, H., Saunders, D. J., Siegelmann, H. T., and Kozma, R. (2019). Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to Atari Breakout game. *Neural Netw.* 120, 108–115. doi: 10.1016/j.neunet.2019.08.009
- Pfister, J.-P., and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.* 26, 9673–9682. doi: 10.1523/JNEUROSCI.1425-06.2006
- Rueckauer, B., and Liu, S.-C. (2018). “Conversion of analog to spiking neural networks using sparse temporal coding,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Florence: IEEE), 1–5.
- Rumbell, T., Denham, S. L., and Wennekers, T. (2014). A spiking self-organizing map combining STDP, oscillations, and continuous learning. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 894–907. doi: 10.1109/TNNLS.2013.2283140
- Sakemi, Y., Morino, K., Morie, T., and Aihara, K. (2023). A Supervised learning algorithm for multilayer spiking neural networks based on temporal coding toward energy-efficient VLSI processor design. *IEEE Trans. Neural Netw. Learn. Syst.* 34, 394–408. doi: 10.1109/TNNLS.2021.3095068
- Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926. doi: 10.1038/78829
- Stecker, G. C., and Middlebrooks, J. C. (2003). Distributed coding of sound locations in the auditory cortex. *Biol. Cybernet.* 89, 341–349. doi: 10.1007/s00422-003-0439-1
- Tavanaei, A., Masquelier, T., and Maida, A. (2018). Representation learning using event-based STDP. *Neural Netw.* 105, 294–303. doi: 10.1016/j.neunet.2018.05.018
- Thorpe, S. J., and Gautrais, J. (1998). “Rank order coding,” in *Computational Neuroscience*, ed J. M. Bower (Boston, MA: Springer US), 113–118.
- Yen, E. K., and Johnston, R. G. (1996). *The Ineffectiveness of the Correlation Coefficient for Image Comparisons*. Technical Report LA-UR-96-2474, Los Alamos, NM.
- Zhang, M., Wang, J., Wu, J., Belatreche, A., Amornpaisannon, B., Zhang, Z., et al. (2022). Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 1947–1958. doi: 10.1109/TNNLS.2021.3110991
- Zhou, Y., Mesik, L., Sun, Y. J., Liang, F., Xiao, Z., Tao, H. W., et al. (2012). Generation of spike latency tuning by thalamocortical circuits in auditory cortex. *J. Neurosci.* 32, 9969–9980. doi: 10.1523/JNEUROSCI.1384-12.2012