

Detection of spiking motifs of arbitrary length in neural activity using bounded synaptic delays

T. Kronland-Martinet S. Viollet L. Perrinet

November 20, 2025

Abstract

In the context of spiking neural networks, temporal coding of signals is increasingly preferred over the rate coding hypothesis due to its advantages in processing speed and energy efficiency. In temporal coding, synaptic delays are crucial for processing signals with precise spike timings, known as spiking motifs. Synaptic delays are however bounded in the brain and can thus be shorter than the duration of a motif. This prevents the use of motif recognition methods that consist of setting heterogeneous delays to synchronize the input spikes on a single output neuron acting as a coincidence detector. To address this issue, we developed a method to detect motifs of arbitrary length using a sequence of output neurons connected to input neurons by bounded synaptic delays. Each output neuron is associated with a sub-motif of bounded duration. A motif is recognized if all sub-motifs are sequentially detected by the output neurons. We simulated this network using leaky integrate-and-fire neurons and tested it on the Spiking Heidelberg Digits (SHD) database, that is, on audio data converted to spikes via a cochlear model, as well as on random simultaneous motifs. The results demonstrate that the network can effectively recognize motifs of arbitrary length extracted from the SHD database. Our method features a correct detection rate of about 60% in presence of ten simultaneous motifs from the SHD dataset and up to 80% for five motifs, showing the robustness of the network to noise. Results on random overlapping patterns show that the recognition of a single motif overlapping with other motifs is most effective for a large number of input neurons and sparser motifs. Our method provides a foundation for more general models for the storage and retrieval of neural information of arbitrary temporal lengths.

1 Introduction

Increasing evidence suggests that precise spike timings play a crucial role in neuronal communication within the brain [1]. The temporal coding hypothesis has gained significant attention due to its efficiency in processing speed compared to rate coding, which typically requires longer processing times [2, 3]. A crucial model was found in spiking neural networks where neurons are randomly connected with random frozen synaptic delays [4]. Groups known as polychronous groups defined by precise millisecond-scale spiking motifs have been observed to naturally emerge as a consequence of Spike-Timing Dependent Plasticity (STDP) learning [5]. These polychronous groups were reoccurring multiple times during simulations, even when the network inputs were random, suggesting that learning motifs with precise spike timings could be highly relevant in the brain [6].

Moreover, reducing time precision leads to a degradation of results such that the precise timing of spikes is crucial for motif recognition tasks [7]. Previous studies on motif recognition have primarily focused on synaptic weights, as seen in Hopfield networks [8], often disregarding the temporal precision of learned motifs. Some Hebbian learning rules, such as STDP, consider the precise timing of spikes for learning but most often update only the synaptic weights [9]. Other studies have suggested strategies for learning temporal motifs. For instance, networks using bistable neurons and global inhibition can recognize sequences of spikes of arbitrary length [10, 11, 12]. However, these networks do not account for precise spike timings. In contrast, networks with heterogeneous delays have been used to recognize motifs with precise spike timing [13, 14]. However, although delays can be set with arbitrary lengths on a computer, they are constrained in the brain, with synaptic delays generally considered to be shorter than 30 ms [15, 16]. Therefore, as defined in [13, 14], synaptic delays highly constrain the length of learned motifs. Learning rules, such as the DELTRON [17], have been developed to learn delays but do not consider the limited range of delays imposed by biological neurons.

To overcome these constraints, we developed a network architecture capable of detecting motifs of arbitrary length using bounded delays. This network operates by dividing a spiking motif of arbitrary length into sub-motifs of bounded duration. For each sub-motif, an output neuron is connected to the input neurons to learn the sub-motif. Output neurons are then connected together sequentially, and a motif is recognized if the final output neuron fires following the sequential activation of all sub-motifs. We tested this network in simulations using audio data converted to spikes via a model of the cochlea. The network successfully recognized motifs even when they were presented simultaneously. Additionally, we evaluated the network’s capacity to recognize a motif in the presence of simultaneous motifs using random synthetic motifs. This test, related to the cocktail party problem [18], highlighted the network’s ability to recognize motifs even with high levels of background noise. In the following sections, we first describe the network in Section 2, then detail the simulation process in Section 2.3, and analyze the results in Section 3. Finally, we discuss the results in Section 4.

2 Methods: recognition network for arbitrary-length motifs

2.1 Network architecture

In this section, we describe the architecture of the network designed to recognize a single spiking motif generated by a group of input neurons. Recognition is achieved through the sequential activation of output neurons, each corresponding to a sub-motif, that is, a segment of the motif. Figure 1 provides a visual overview of this architecture.

2.1.1 Definition of spiking motifs

Let us consider $n \in \mathbf{N}^*$ input neurons that fire a motif of duration $T \in \mathbf{R}^{+*}$. The input neurons are labelled $i_1, i_2, \dots, i_n$ and a neuron i_j , with $j \in \{1, \dots, n\}$, emits $m \in \mathbf{N}$ spikes with timings $\{t_1^{i_j}, t_2^{i_j}, \dots, t_m^{i_j}\} \in \mathbf{R}^m$. These timings are ordered, that is, $0 \leq t_1^{i_j} < t_2^{i_j} < \dots < t_m^{i_j} < T$ (cf. figure 1c). The motif is defined as a sequence of spikes from the input neurons, where the timing of each spike is precise. The goal of the network is to recognize this motif by detecting the precise timing of spikes in the input neurons.

To construct the network, the motif duration T is divided in $s \in \mathbf{N}^*$ intervals of length $I = \frac{T}{s}$ (cf. figure 1a). The parameter s should be chosen such as the length of the intervals is shorter than the maximum possible synaptic delay. We can compute the number of input neurons spikes in the motif that occurs in each interval $I_\ell = [(\ell - 1)I, \ell I]$, where $\ell \in \{1, \dots, s\}$, as follows:

$$N_{I_\ell} = \sum_{j=1}^n \sum_{k=1}^m \mathbf{1}_{(\ell-1)I \leq t_k^{i_j} < \ell I} \quad (1)$$

Where $\mathbf{1}_C$ is the Kronecker delta function, which is equal to 1 if the condition C is true and 0 otherwise (cf. figure 1b).

2.1.2 Sub-motif detection

The input neurons are connected to a group of s output neurons labelled $o_1, o_2, \dots, o_s$ each associated to a time interval. Each output neuron is such that o_ℓ has N_{I_ℓ} synapses from input neurons. For each spike timing $t_k^{i_j}$ such as $(\ell - 1)I \leq t_k^{i_j} < \ell I$, a synapse connects input neuron i_j to output neuron o_ℓ with a delay tuned such that the spikes converging on the soma of output neuron o_ℓ synchronize to the end of the interval I_ℓ . In other words, we define the synaptic delay $\delta_{i_j, o_\ell, k}$ from neuron i_j to output neuron o_ℓ such as if neuron i_j spikes at time $t_k^{i_j}$, with respect to the beginning of the motif, this spike reaches neuron o_ℓ at time $t_{i\ell} = \ell I + a$. Here, $a > 0$ is a constant added to ensure that every synaptic delay is strictly positive, i.e., $\delta_{i_j, o_\ell, k} \geq a$. This prevents any synapse from having zero delay, which is biologically implausible and problematic in simulations. As a consequence, if delays are attuned regarding the following equation :

$$t_{i\ell} = \ell I + a = \delta_{i_j, o_\ell, k} + t_k^{i_j} \quad (2)$$

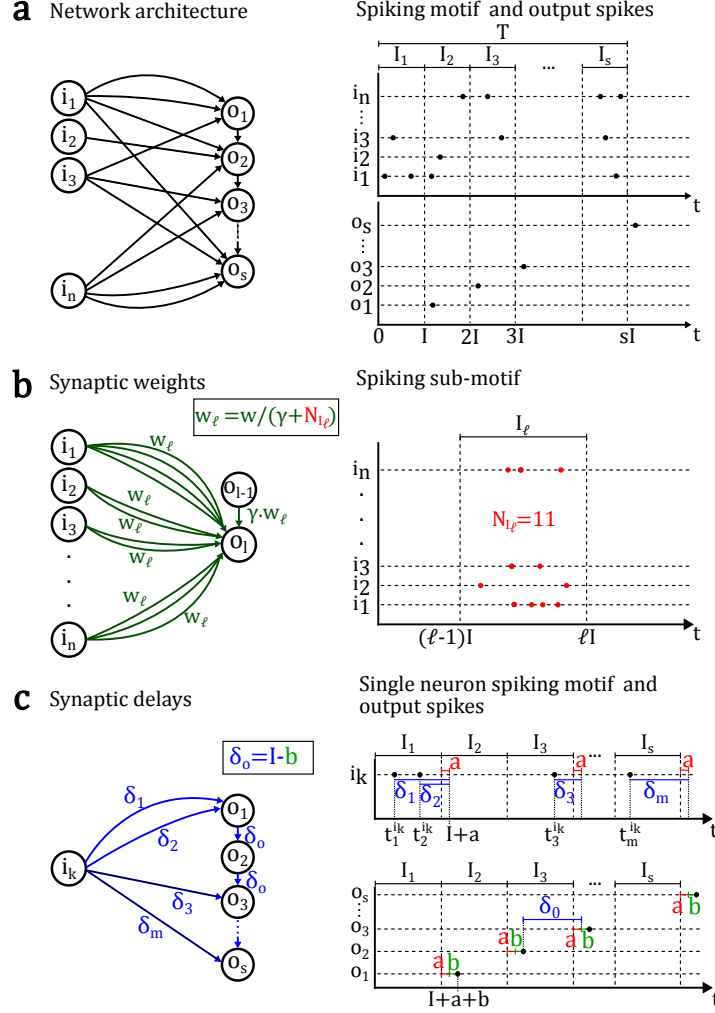


Figure 1: Neural network architecture for the recognition of arbitrary-length motifs (each black or red dot on right-side figures correspond to a spike). a) Network architecture : the connectivity between input and output neurons follows the distribution of input spikes within each sub-motif I . b) Distribution of the synaptic weights according to equation 9. c) Calculus of the synaptic delay according to equation 7 and their representation for a given motif.

then all spikes from the sub-motif reach the output neuron at the same precise time. Finally, the synaptic delay is therefore defined as (cf. figure 1c) :

$$\delta_{ij,o_\ell,k} = \ell I + a - t_k^{ij} \quad (3)$$

The synaptic weights are defined such as neuron o_ℓ fires only if all spikes of the motif occurring in the interval I_ℓ reach simultaneously the neuron o_ℓ . Details are given in the following section.

2.1.3 From local to global detection

An input spiking motif may be triggered after a silent period, such that we define the first output neuron o_f having connections with input neurons by its index f :

$$f = \min_{\ell} (N_{I_\ell} > 0) \quad (4)$$

For output neuron o_f , the synaptic weights are defined such that o_f fires only if it receives N_{I_f} synchronous spikes. Let us consider w the minimum weight such that a Leaky Integrate-and-Fire (LIF) neuron fires from a single input (see next section for more details). The synaptic weight associated with each input synapse to output neuron o_f is given by

$$w_f = \frac{w}{N_{I_f}} \quad (5)$$

To allow for noisy inputs, a threshold can be defined such as o_f fires if input spikes are missing or if the input spikes are only approximately synchronous. Equivalently, we can define a new weight:

$$w_{f,noisy} = w_f + \epsilon, \epsilon \in \mathbf{R}^+ \quad (6)$$

To ensure a chain of detection of the global motif by the convergence of the spiking motifs within each interval I_ℓ , each output neuron o_ℓ with $f \leq \ell < s$ is connected to the next output neuron $o_{\ell+1}$ with a synaptic delay $\delta_{o_\ell,o_{\ell+1}}$. As precised in section 2.1.2, if the learned motif occurs, the output neuron o_ℓ should receive all simultaneous input spikes at time $t_{i\ell} = \ell I + a$, with respect to the beginning of the motif. Therefore, by considering its dynamic, neuron o_ℓ would fire at time $t_{o\ell} = \ell I + a + b$, where $b > 0$ is the delay neuron o_ℓ needs to generate a spike when receiving the learned input signal. Similarly, output neuron $o_{\ell+1}$ should receive all input spikes at time $t_{i(\ell+1)} = (\ell + 1)I + a$, including the spike from output neuron o_ℓ . Therefore, the delay $\delta_{o_\ell,o_{\ell+1}}$ is defined such that (cf. figure 1c):

$$\begin{aligned} t_{i(\ell+1)} &= t_{o\ell} + \delta_{o_\ell,o_{\ell+1}} \\ \iff (\ell + 1)I + a &= \ell I + a + b + \delta_{o_\ell,o_{\ell+1}} \\ \iff \delta_{o_\ell,o_{\ell+1}} &= I - b \end{aligned} \quad (7)$$

Except for output neuron o_f , the synaptic weight of input synapses to output neuron o_ℓ are given by:

$$w_\ell = \frac{w}{N_{I_\ell} + 1} \quad (8)$$

Such that neuron o_ℓ spikes only if it receives N_{I_ℓ} synchronous spike from input neurons and one more synchronous spike from the previous output neuron.

If we want to ensure that the output neurons spike only if the previous output neuron spiked, we can define the synaptic weight of input synapses to output neuron o_ℓ by (cf. figure 1b):

$$w_\ell = \frac{w}{N_{I_\ell} + \gamma} \quad (9)$$

In that equation, $\gamma > 1$ is a parameter giving the relative strength of output to output synapses with respect to input to output synapses, such that:

$$w_{o_{\ell-1}o_\ell} = \frac{\gamma w}{N_{I_\ell} + \gamma} = \gamma w_\ell \quad (10)$$

is the weight between two successive output neurons $o_{\ell-1}$ and o_ℓ . By increasing the value of γ , we reduce the probability for neurons to spike independently from the learned sequence.

All weights can be adjusted to process noisy motifs, as shown in eq. 6. Finally, the motif is recognized if and only if neuron o_s fires.

2.2 Network implementation using LIF neurons

2.2.1 Definition of w and b from the Leaky Integrate-and-Fire (LIF) neuron equation

To explain the choice of parameters w and b defined in the previous section, we consider the LIF model of a spiking neuron for its simplicity and because it was sufficient for the efficient simulation of the network. A LIF neuron's membrane potential is defined by [19]:

$$\tau_m \frac{du}{dt} = -[u(t) - u_{\text{rest}}] + RI(t) \quad (11)$$

Where u is the membrane potential of the neuron, τ_m its membrane time constant, u_{rest} is the neuron's resting membrane potential, R the membrane resistance and I the input signal's intensity.

The solution of the LIF membrane potential equation for a neuron receiving an input spike of intensity $I(t) = wI_0$ for $0 < t < \Delta$ is:

$$\begin{cases} u(t) = u_{\text{rest}} & \text{if } t \leq 0 \\ u(t) = u_{\text{rest}} + RwI_0 \left[1 - \exp\left(-\frac{t}{\tau_m}\right) \right] & \text{if } 0 < t < \Delta \end{cases} \quad (12)$$

Where w is the synaptic weight and I_0 is the current generated by a spike.

The neuron spikes if $u \geq u_{\text{threshold}}$ ($u_{\text{rest}} < u_{\text{threshold}}$). If we choose w such as $u(\Delta) = u_{\text{threshold}}$, the delay b , defined in section 2.1, is given by $b = \Delta$, and w is given by:

$$\begin{aligned} u(\Delta) &= u_{\text{threshold}} \\ \Leftrightarrow u_{\text{threshold}} &= u_{\text{rest}} + RwI_0 \left[1 - \exp\left(-\frac{\Delta}{\tau_m}\right) \right] \\ \Leftrightarrow w &= \frac{u_{\text{threshold}} - u_{\text{rest}}}{RI_0 \left[1 - \exp\left(-\frac{\Delta}{\tau_m}\right) \right]} \end{aligned} \quad (13)$$

If the neuron's membrane potential reaches the threshold, it is reset to a value u_{reset} . In this paper, we set $u_{\text{reset}} = u_{\text{rest}}$. The parameters and spike generation are shown in figure 2

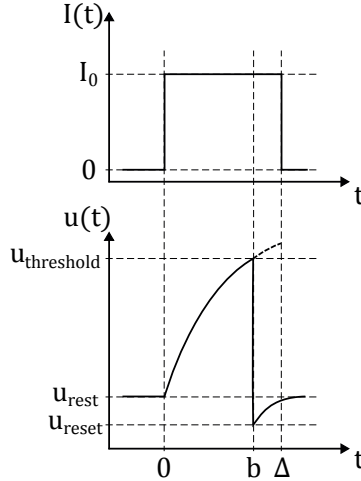


Figure 2: Neuron spike generation from an input current. Top: input current. Bottom: membrane potential of the neuron as a function of time.

2.2.2 Recognition of a spiking motif of arbitrary length

When a set of input neurons spike a learned motif, the network activates in a sequential manner. The input neurons spike the first sub-motif at time interval I_f . Input spikes propagate through the synapses with various delays, and will reach output neuron o_f simultaneously at time $t_{if} = fI + a$. Neuron o_f will therefore fire at time $t_{of} = fI + a + b$. Meanwhile, the input neurons generate spikes

from the next sub-motif starting at time interval I_{f+1} . Input spikes and the spike from output neuron o_f will reach synchronously the output neuron o_{f+1} , at time $t_{i(f+1)} = (f+1)I + a$. Neuron o_{f+1} will then fire at time $t_{o(f+1)} = (f+1)I + a + b$. This sequence of activation of output neurons will go on until neuron o_s will emit a spike, meaning that the sequence is recognized (cf. figure 1a,c for visualization of sequential activation of spikes). If one output neuron does not spike, the next output neurons will not spike (except in some particular cases where parameters allow for noisy inputs, cf. eq. 6), and the pattern will therefore not be recognized.

2.2.3 Parallel recognition of multiple motifs of arbitrary length

Since input neurons can generate several motifs, we may need to learn more than one motif. For each motif we wish to learn, a group of output neurons is connected to the input neurons. All sets of output neurons, encoding a motif, are independent from each other. A visualization of an architecture encoding two distinct patterns is given in figure 3.

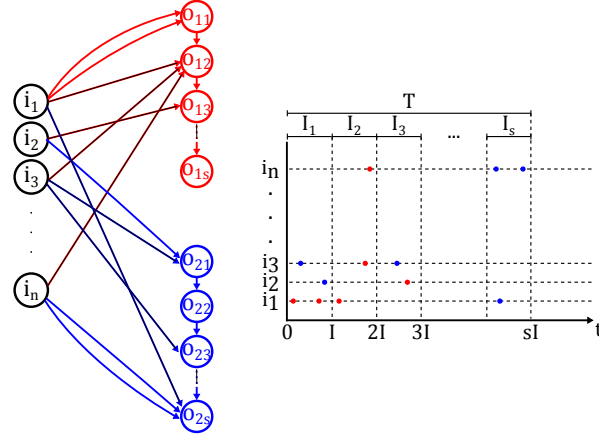


Figure 3: Neural network architecture for coding distinct motifs, here two motifs plotted respectively with blue and red spikes.

2.3 Simulation of the network

The network was simulated using Python with the PyNN Application Programming Interface (API) [20], which provides a unified language. One advantage of using PyNN is that it allows for easy switching between different simulators, such as NEST, Brian2, or NEURON, without changing the code. This flexibility is beneficial for testing the network on different platforms and ensuring compatibility with various simulation environments but also to extend it to neuromorphic hardware in the near future [21]. For this work, we used PyNN with the NEST backend, which efficiently integrates the LIF neuron model described in equation 11 using optimized C++ code.

To test the recognition of a given set of known motifs, a network was built following the description in section 2.1. For each motif, input neurons were connected to a group of output neurons, as represented in figure 3. The network was simulated using LIF neurons with fixed threshold and alpha-function-shaped post-synaptic current (IF_curr_alpha on PyNN). To avoid long term effects of input spikes on the membrane potential of output neurons, the membrane time constant of the neurons was set to 1 ms such that neurons had fast dynamics. We also set a non zero refractory time for all neurons in order to avoid excessive spiking of neurons since the network was only composed of excitatory neurons. The initial, resting and reset membrane potential of the neurons were all set to -65mV, and the threshold was set to -50mV.

The number of output neurons was defined with respect to the maximum synaptic delays and the duration of the learned motif. For instance, 10 output neurons are required to memorize a 100 ms duration motif with maximum synaptic delays of 10ms. The synapses connecting input neurons to output neurons have constant weights and delays, as defined in section 2.1. The synaptic weight of output to output neurons were 10 times higher than the weights of input to output neurons ($\gamma = 10$ in

eq. 9 and 10) Our code was partly compiled using Numba [22], and processed in parallel on multiple CPU's using Joblib. Reproducible code is available at [your_code_link_here](#).

3 Results

3.1 Random synthetic spiking motifs

To evaluate the network's ability to recognize a learned motif in the presence of simultaneous motifs (i.e., noise), we generated random synthetic spiking motifs to serve as input. Each motif was constructed by specifying the minimum and maximum interspike intervals, the total motif duration, and the maximum number of spikes per neuron. For each neuron, spike times were drawn from a uniform distribution within the specified interval, with a minimum interval of 3 ms enforced to prevent excessive firing.

In all tests, the membrane time constant was set to 1 ms for all neurons and the refractory period was set to 1 ms for input neurons and 0.5 ms for output neurons, balancing the need to prevent excessive spiking while allowing motif detection in noisy conditions. All networks used a maximum synaptic delay of 10 ms. For each configuration, 40 independent trials were performed with randomly generated motifs to ensure statistical reliability.

The first experiment evaluated the network's ability to recognize a single motif in the presence of multiple overlapping motifs, as a function of motif duration and the number of overlapping motifs. Here, the number of input neurons was fixed at 100, and the maximum interspike interval was set to 500 ms, resulting in a mean firing rate of approximately 4 Hz per neuron. Results are shown in Figure 4. The proportion of correctly recognized motifs decreases as either the number of simultaneous motifs or the motif duration increases. This is expected, as more simultaneous motifs reduce the signal-to-noise ratio, and longer motif durations increase the likelihood of spike conflicts at the output neurons. If an output neuron spikes at the wrong time due to noise, its refractory period may prevent it from firing when the actual motif occurs. Both effects contribute to an increase in false negatives, reducing the proportion of motifs successfully recognized.

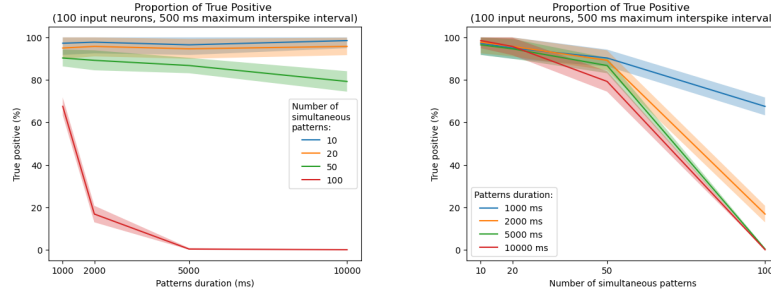


Figure 4: Results of the simulated network on simultaneous random motifs for various numbers of simultaneous motifs and motifs duration. The shaded bands give the standard deviation of the results, each configuration was tested 40 times. Both figures correspond to the same simulation. We only display the proportion of true positive in this figure since the proportion of false positive is always null, and the proportion of false negative can be deduced from the proportion of true positive and eq. 15.

The second experiment analyzed the network's ability to recognize a single motif in the presence of simultaneous motifs, as a function of the number of input neurons and the maximum interspike interval (which determines the firing rate). In this test, 50 motifs of 5000 ms duration were generated. Results are shown in Figure 5.

The results indicate that the proportion of correctly recognized motifs increases with both the number of input neurons and the maximum interspike interval (i.e., as the firing rate decreases). Increasing the number of input neurons expands the space of possible motifs, reducing the likelihood that an output neuron is incorrectly activated by noise. False negatives typically occur when an output neuron spikes prematurely due to noise and is then unable to fire at the correct time due to its

refractory period. Since output neurons are activated sequentially, a missed or mistimed spike in any output neuron prevents recognition of the entire motif.

Similarly, increasing the maximum interspike interval (lowering the firing rate) makes motifs sparser, which reduces the probability of spike conflicts and erroneous output neuron activation. At high firing rates, the proportion of false positives is elevated, but this decreases as firing rates drop. False negatives also decrease with sparser activation, as fewer spike conflicts occur. When input neurons are highly active, false positives dominate; as activity decreases, errors shift toward false negatives due to missed activations. Overall, correct recognition improves as input neuron activity becomes sparser.

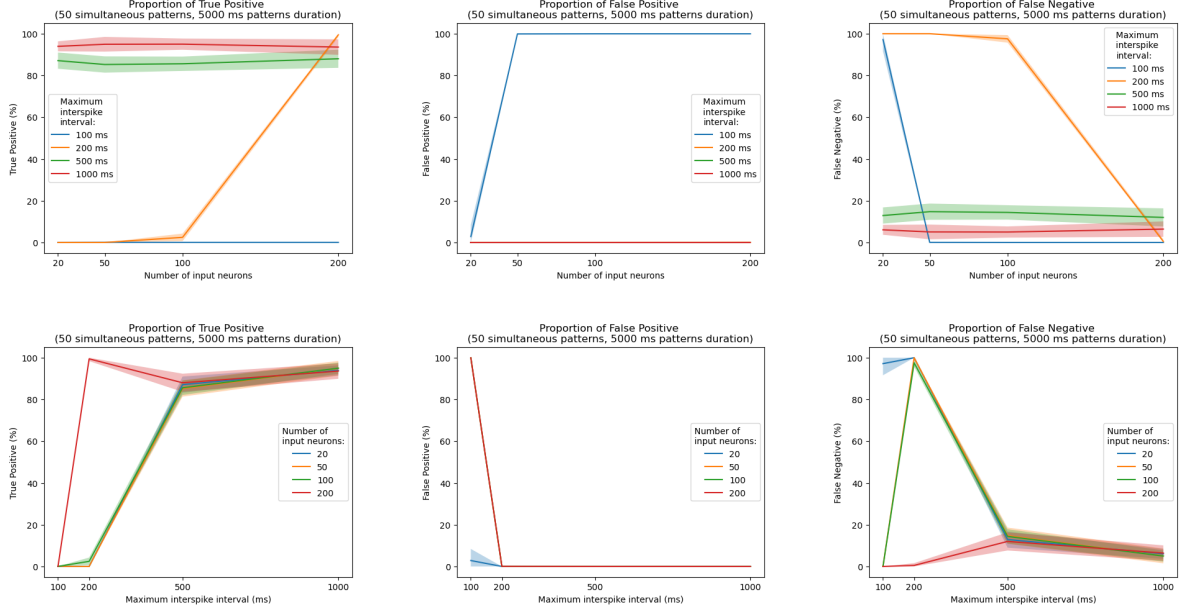


Figure 5: Results of the simulated network on simultaneous random motifs for various numbers of input neurons and maximum interspike interval. The shaded bands indicate the standard deviation across 40 trials per configuration. All figures correspond to the same simulation.

In summary, these tests demonstrate that the motif recognition network is robust to noise, with optimal detection achieved for short motifs, a large number of input neurons, and sparse neuronal activation. An unexpected result in Figure 5 is that, for 200 input neurons, the highest recognition rate occurs at a maximum interspike interval of 200 ms. This phenomenon is not yet fully understood and suggests that optimal configurations may depend on the specific characteristics of the input motifs.

3.2 Spiking Heidelberg Digits

The network was further evaluated using the Spiking Heidelberg Digits (SHD) dataset [23]. This dataset consists of audio recordings of spoken digits (0–9) in both English and German, converted to spike trains via the Lauscher cochlear model. In figure 6a, label “4” refers to the English digit ‘four’, and label “17” to the German digit ‘sieben’ (‘seven’). The dataset was accessed using the Tonic Python library [24]. To reduce simulation time, the spike data was downsampled by a factor of 10^3 . Since SHD data have microsecond resolution, this downsampling step leads to millisecond resolution of the data (using Tonic’s Downsample transform), and the minimum interspike interval (ISI) was set to 5 ms to prevent excessive output neuron spiking.

For clarity, we standardized the motif duration to 4 seconds, even though the actual motifs varied in length. If a motif was shorter than 4 seconds (see motif 1 in figure 6a), spikes simply propagated through the remaining output neurons without requiring additional input spikes. Each motif was encoded by connecting input neurons to a sequence of output neurons, with synaptic delays capped at 20 ms. Thus, a 4-second motif was represented by 200 output neurons (one per 20 ms interval). In figure 6b-c, recognition of motif 1 was indicated by a spike from output neuron number 199, which corresponds to the final interval of the motif.

All neurons were configured with a refractory period of 2 ms and a membrane time constant of 1 ms. Connections followed the architecture described in Section 2.1. For testing, the input signal consisted of: motif 1 (from 10 ms to 4010 ms), motif 2 (from 4010 ms to 8010 ms), and then both motifs presented simultaneously with an offset of 190 ms between the motif 1 (from 8020 ms to 12020 ms) and the motif 2 (from 8210 ms to 12210 ms). This offset allowed us to distinguish true positive motif recognition from false positives. It was chosen arbitrarily, but long enough to distinguish visually the recognition of both motifs by output neurons in Figure 6b-c, while maximizing the overlap between both patterns, as seen in Figure 6a. Motif 1 (label 4) was detected at 4013.1 ms and 12023.1 ms, and motif 2 (label 17) at 8016.6 ms and 12216.6 ms, as indicated by spikes from output neuron number 199 in Figure 6c. These detection times correspond to the end of each motif, confirming that the network accurately identified both the timing and identity of the motifs—even when they overlapped.

In addition, we performed similar tests with an increasing number of simultaneous motifs. In those tests, only overlapping motifs were tested, and motifs were offset from each other by 50 ms. The number of simultaneous motifs ranged from 2 to 20, and each condition was tested 40 times with random motifs from the SHD dataset. For each test, we considered that the recognition was successful only if all patterns were correctly detected. For instance, if one pattern was not recognized, the recognition was considered wrong. The mean proportion, over the 40 tests, of successful recognition for varying number of overlapping patterns is given in figure 7. This result demonstrates the network’s robustness to noise and its ability to reliably recognize multiple motifs within complex input patterns.

4 Discussion

This paper demonstrates that spiking neural networks with bounded synaptic delays can reliably recognize motifs of arbitrary length. The proposed architecture is biologically plausible, reflecting key constraints observed in the brain. Notably, the network features a much higher number of synapses than neurons, mirroring biological circuits [25]. Simulations show that motif recognition remains robust even when multiple motifs occur simultaneously, and that sparse neuronal activity further improves accuracy. This result is in line with experimental findings about cortical neurons typically firing less than one spike every 7 seconds [26]. Moreover, encoding information in precise spike timings, rather than firing rates, makes the network resilient to changes in stimulus intensity. For example, in the brain, sensory inputs are often encoded on a logarithmic scale, so variations in intensity affect firing rates but not the relative timing of spikes [27].

Despite these strengths, the network has several limitations. First, it currently includes only excitatory neurons. To prevent excessive spiking and false positives, we imposed a nonzero refractory period and a minimum interspike interval of 3 ms. Incorporating inhibitory neurons could help regulate activity through homeostatic mechanisms [28] and enable recognition of a broader range of motifs. Second, motif recognition depends on a single output neuron firing; if this neuron (or any in the output chain) fails, the motif is not detected—a challenge known as the ‘grandmother cell problem’ [29]. Using populations of output neurons, rather than single cells, could improve robustness. Adding skip connections between output neurons may also allow partial motif recognition, even if one neuron fails to fire. Probabilistic neurons with stochastic escape rates could further enhance detection of noisy or incomplete motifs.

Future work should address several open questions. For example, the network’s memory capacity could be analyzed more formally, as initial results suggest high storage potential for simultaneous motifs. Recent advances show that Hopfield networks can achieve exponential memory capacity by optimizing synaptic weights [30]; combining weights and delays may further increase capacity in spiking networks. The impact of interspike interval distributions also merits investigation: while this study used uniform intervals, biological spike trains are often modeled as Poisson processes, which may affect recognition performance. Synaptic delays could be combined with other temporal mechanisms, such as adaptive firing or variable membrane time constants, to reduce the number of output neurons needed for motif encoding. Learning both delays and weights through supervised or unsupervised methods could enable motif classification; relevant synaptic learning rules have been proposed [31, 32, 33]. Additionally, while all spikes within a time interval were assigned equal synaptic weights in this study, some spikes may be more informative for motif recognition, for instance because they would be more selective to a motif, suggesting that heterogeneous weights could be beneficial. The network is well suited for processing asynchronous spiking motifs, such as those from neuromorphic sensors, and could

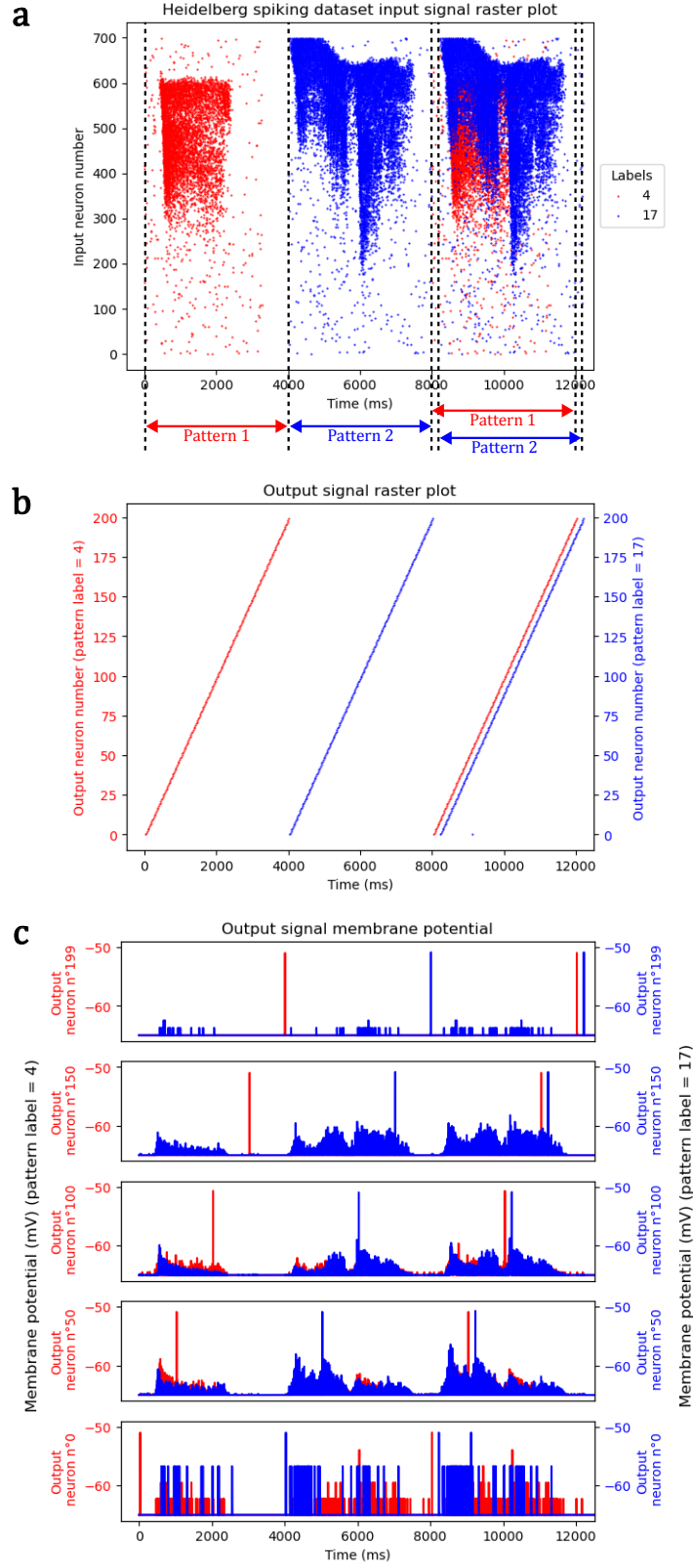


Figure 6: Simulation of the network on two motifs from the Spiking Heidelberg Digits (SHD) dataset. a) Two spiking patterns (in blue and red) corresponding to two different digits (4 and 17). To test the robustness of the NN, patterns were overlapped. b) Activation of the output neurons of each networks coding for each pattern. c) Membrane potentials versus time of 5 output neurons out of 200.

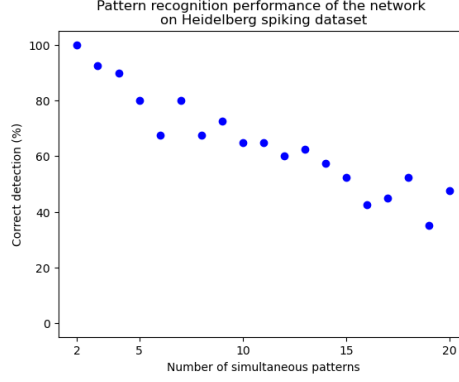


Figure 7: Performance of the network as a function of the number of simultaneous motifs from the Spiking Heidelberg Digits (SHD) dataset. For each condition, 40 tests were performed. Each point corresponds to the mean value over the 40 tests.

be tested in real time on neuromorphic hardware. Finally, as discussed in [34] about the cocktail party problem, future studies could explore our network’s ability to detect weak signals in the presence of periodic motifs.

5 Materials and methods

In section 3.1, to systematically test robustness, motifs were designed to partially overlap rather than occur strictly simultaneously. This was achieved by introducing a fixed offset of 10 ms between the start times of each motif. For example, with three motifs of 100 ms duration, the first motif would span 10–110 ms, the second 20–120 ms, and the third 30–130 ms. Since the network typically recognizes a motif within 10 ms after its end, motif 1 was considered correctly recognized only if detection occurred between 110 and 120 ms. To ensure sufficient overlap, motif durations were chosen to be at least as long as the total offset (e.g., at least 30 ms for three motifs).

This approach allowed us to systematically assess the network’s robustness to overlapping and noisy spike patterns. Simulation results are reported as proportions of true positives, false positives, and false negatives in motif recognition. Specifically, the proportion of true positives P_{TP} is defined as:

$$P_{TP} = \frac{TP}{TP + FP + FN}, \quad (14)$$

where TP is the number of correctly recognized motifs (true positives), FP is the number of incorrectly recognized motifs (false positives), and FN is the number of missed motifs (false negatives). We do not consider the number of correctly ignored motifs (true negatives) here, since this number would be too large. Similarly, the proportions of false positives P_{FP} and false negatives P_{FN} are given by:

$$\begin{aligned} P_{FP} &= \frac{FP}{TP + FP + FN}, \\ P_{FN} &= \frac{FN}{TP + FP + FN} = 1 - P_{FP} - P_{TP} \end{aligned} \quad (15)$$

References

- [1] Antoine Grimaldi, Amélie Gruel, Camille Besnainou, Jean-Nicolas Jérémie, Jean Martinet, and Laurent U Perrinet. Precise spiking motifs in neurobiological and neuromorphic data. *Brain Sciences*, 13(1):68, 2022.
- [2] Rufin Van Rullen and Simon J Thorpe. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural computation*, 13(6):1255–1283, 2001.
- [3] Romain Brette. Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in systems neuroscience*, 9:151, 2015.

- [4] Eugene M Izhikevich. Polychronization: computation with spikes. *Neural computation*, 18(2):245–282, 2006.
- [5] Eugene M Izhikevich, Joseph A Gally, and Gerald M Edelman. Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, 14(8):933–944, 2004.
- [6] Vincent Villette, Arnaud Malvache, Thomas Tressard, Nathalie Dupuy, and Rosa Cossart. Internally Recurring Hippocampal Sequences as a Population Template of Spatiotemporal Information. *Neuron*, 88(2):357–366, October 2015. 00085.
- [7] Himanshu Akolkar, Cedric Meyer, Xavier Clady, Olivier Marre, Chiara Bartolozzi, Stefano Panzeri, and Ryad Benosman. What can neuromorphic event-driven precise timing add to spike-based pattern recognition? *Neural Computation*, 27(3):561–593, 2015.
- [8] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- [9] L. Perrinet. Coherence detection in a spiking neuron via Hebbian learning. *Neurocomputing*, 44–46(C):133–139, 2002.
- [10] Qiang Yu, Rui Yan, Huajin Tang, Kay Chen Tan, and Haizhou Li. A spiking neural network system for robust sequence recognition. *IEEE transactions on neural networks and learning systems*, 27(3):621–635, 2015.
- [11] Dezhe Z Jin. Spiking neural network for recognizing spatiotemporal sequences of spikes. *Physical Review E*, 69(2):021905, 2004.
- [12] Dezhe Z Jin. Decoding spatiotemporal spike sequences via the finite state automata dynamics of spiking neural networks. *New Journal of Physics*, 10(1):015010, 2008.
- [13] K Unnikrishnan, John J Hopfield, and David W Tank. Connected-digit speaker-dependent speech recognition using a neural network. *IEEE Transactions on Signal Process*, 39, 1991.
- [14] David W Tank and JJ Hopfield. Neural computation by concentrating information in time. *Proceedings of the National Academy of Sciences*, 84(7):1896–1900, 1987.
- [15] H. A. Swadlow. Physiological properties of individual cerebral axons studied in vivo for as long as one year. *Journal of Neurophysiology*, 54(5):1346–1362, 1985.
- [16] Jean-Didier Lemaréchal, Maciej Jedynak, Lena Trebaul, Anthony Boyer, François Tadel, Manik Bhattacharjee, Pierre Deman, Viateur Tuyisenge, Leila Ayoubian, Etienne Hugues, et al. A brain atlas of axonal and synaptic delays based on modelling of cortico-cortical evoked potentials. *Brain*, 145(5):1653–1667, 2022.
- [17] Shaista Hussain, Arindam Basu, Mark Wang, and Tara Julia Hamilton. Deltron: Neuromorphic architectures for delay based learning. In *2012 IEEE Asia Pacific Conference on Circuits and Systems*, pages 304–307. IEEE, 2012.
- [18] Josh H McDermott. The cocktail party problem. *Current Biology*, 19(22):R1024–R1027, 2009.
- [19] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [20] Andrew P Davison. PyNN: A common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2:11, 2008.
- [21] Julian Göltz, Jimmy Weber, Laura Kriener, Sebastian Billaudelle, Peter Lake, Johannes Schemmel, Melika Payvand, and Mihai A Petrovici. Delgrad: Exact event-based gradients for training delays and weights on spiking neuromorphic hardware. *arXiv preprint arXiv:2404.19165*, 2024.
- [22] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

- [23] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.
- [24] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations. Technical report, July 2021. Documentation available under <https://tonic.readthedocs.io>.
- [25] György Buzsáki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3):362–385, 2010.
- [26] Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.
- [27] John J Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.
- [28] Laurent U Perrinet. An adaptive homeostatic algorithm for the unsupervised learning of visual features. *Vision*, 3(3):47, 2019.
- [29] Charles G Gross. Genealogy of the “grandmother cell”. *The Neuroscientist*, 8(5):512–518, 2002.
- [30] Hubert Ramsauer, Bernhard Schäffl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.
- [31] Antoine Grimaldi and Laurent U Perrinet. Learning heterogeneous delays in a layer of spiking neurons for fast motion detection. *Biological Cybernetics*, 117(4):373–387, 2023.
- [32] Laurent U Perrinet. Accurate detection of spiking motifs by learning heterogeneous delays of a spiking neural network. In *ICANN Special Session on Recent Advances in Spiking Neural Networks*, 2023.
- [33] Shaista Hussain, Arindam Basu, Runchun Mark Wang, and Tara Julia Hamilton. Delay learning architectures for memory and classification. *Neurocomputing*, 138:14–26, 2014.
- [34] Maria Schlungbaum and Benjamin Lindner. Detecting a periodic signal by a population of spiking neurons in the weakly nonlinear response regime. *The European Physical Journal E*, 46(11):108, 2023.