# Working Memory in a Recurrent Spiking Neural Networks With Heterogeneous Synaptic Delays

**Laurent U Perrinet***
Institut de Neurosciences de la Timone (UMR7289)
Aix Marseille Univ, CNRS
27 Bd Moulin, 13005 Marseille, France
laurent.perrinet@univ-amu.fr

## Abstract

Working memory — the ability to store and recall precise temporal patterns of neural activity — remains an open challenge for spiking neural networks (SNNs). We propose a recurrent SNN of $N$ neurons in which each synapse is equipped with $D = 41$ delays, modelled as a weight tensor $\mathbf{W} \in \mathbb{R}^{N \times N \times D}$ and trained end-to-end with surrogate-gradient backpropagation through time. The network stores $M$ arbitrary target spike patterns by representing each as a sequential chain of overlapping Spiking Motifs: contiguous windows of length $D$ that uniquely predict spikes at the next time step. On a synthetic benchmark of $M = 8$ patterns ($N = 512$ neurons, $T = 1000$ steps), training achieves a mean F1 score of $0.966$, with recall emerging first near the clamped initialisation window and propagating forward in time. This result demonstrates that heterogeneous delays provide an efficient substrate for working memory in SNNs, enabling energy-efficient neuromorphic edge deployment.

## 1 Introduction

A century ago, Adrian and Zotterman [1926] established that neural communication relies on all-or-none action potentials, *spikes*. Whether information is encoded in precise spike timing or in the coarser measure of instantaneous firing rate remains debated [Softky and Koch, 1993, Bohte, 2004, Grimaldi et al., 2023] — a question that becomes especially sharp for working memory, where the brain must maintain and manipulate information across delays of seconds. In traditional artificial neural networks, long-range temporal dependencies are addressed by gated recurrent unit (GRU) architectures or Transformers [Vaswani et al., 2017], which decouple temporal binding range from the timescale of individual units. The analogous problem for spiking neural networks (SNNs) — binding activity patterns separated by intervals far exceeding a single neuron's time constant — lacks a general solution and remains a key challenge for biologically plausible sequence processing [Yin et al., 2026].

One principled approach relies on the *precise relative timing* of spikes across a population. Abeles [1982] was a precursor in proposing that cortical neurons act as coincidence detectors; Bienenstock [1995] formalised this into *synfire chains*, feedforward pools in which synchronous volleys propagate reliably. Experimental support came from Bruno and Sakmann [2006], who showed that thalamocortical inputs are individually weak but effective through precise synchrony; theoretical work further showed such synchrony can be learned in an unsupervised manner [Perrinet and Samuelides, 2002]. Ikegaya et al. [2004] then directly observed recurring spatiotemporal spike sequences in neocortex at both sub- and supra-second timescales, and complementary hippocampal evidence

*More information on https://laurentperrinet.github.io/publication/perrinet-26-airov/.

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

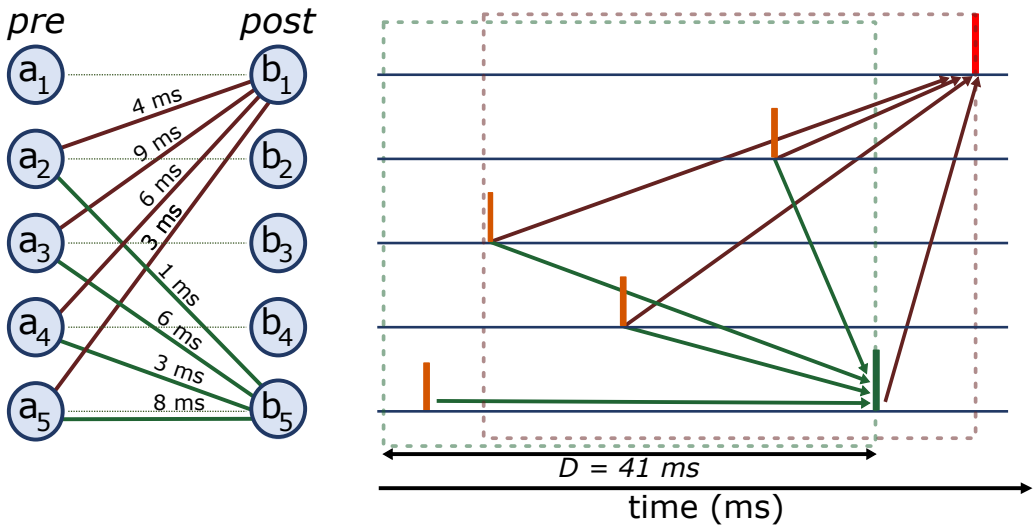


Figure 1: **Working memory as sequential spike prediction via heterogeneous delays.** *Left:* A recurrent HD-SNN of five neurons numbered 1–5, connected by two subsets of recurrent connections that define two Spiking Motifs from the presynaptic side (neurons $a$) to the postsynaptic side (neurons $b$): dark-red connections from $a_2$, $a_3$, $a_4$, and $a_5$ target $b_1$ with delays 4, 9, 6, 3 ms; green connections from $a_1$, $a_2$, $a_3$, $a_4$ target $b_5$ with delays 1, 6, 3, 8 ms. *Right:* Raster plot illustrating how the network predicts successive spikes from a past context window. *Step 1 — first prediction (green box):* the orange input spikes emitted at staggered times by $a_2$–$a_5$ within the first context window (dashed green box) with precise interspike intervals. These are used to define a heterogeneous Spiking Motif such that the (green) delays of all four spikes converge *synchronously* at $b_5$, crossing the firing threshold and producing the green output spike. *Step 2 — second prediction (dark-red box):* the green spike of $a_5$ re-enters the network as a new input and, together with the spikes now visible in the shifted context window (dashed dark-red box), is routed through the dark-red delays in order to converge synchronously on $b_1$, generating its output spike. Because output spikes re-enter the same population, detecting one Spiking Motif provides the context for predicting the next spike in the sequence, implementing working memory as a chain of temporal predictions grounded in the preceding context window.

shows that such sequences self-organise spontaneously as internal population templates [Villette et al., 2015].

Izhikevich [2006] extended the synfire framework by introducing heterogeneous axonal delays with spike-timing-dependent plasticity (STDP). The resulting *polychronous neuronal groups* (PNGs) — time-locked, non-synchronous firing patterns at millisecond precision — vastly outnumber the neurons in the network, yielding a large short-term memory capacity exploited for visual feature binding [Eguchi et al., 2018] and sequence classification Paugam-Moisy et al. [2008]. Szatmáry and Izhikevich [2010] showed that short-term plasticity can cue a PNG's reactivation and sustain a working memory. However, storing spike patterns of arbitrary length is inherently difficult: the sequential structure of the task means each context window is not an independent query but is itself the output of the network at the previous step, so small errors at early time steps propagate and amplify towards the end of the sequence — a compounding difficulty absent from static associative memory.

The limitation of the original model is that delays are fixed rather than learned. Recent work addresses this: Göltz et al. [2021] demonstrated fast, energy-efficient inference on neuromorphic hardware using first-spike-time coding, while Hammouamri et al. [2023] introduced gradient-based delay learning via dilated convolutions with learnable spacings (DCLS), achieving sparse solutions in which only $\sim$20% of delay channels remain active. Applied to event-based vision, heterogeneous learned delays in a single spiking layer suffice for fast motion detection [Grimaldi and Perrinet, 2023], scalable to always-on object recognition [Grimaldi et al., 2024]. Multiple learned delays per synapse give rise to *Spiking Motifs* (SMs) [Perrinet, 2023]: precisely timed spike patterns that generalise PNGs

to a trainable, hardware-compatible framework, here referred to as Heterogeneous-Delay SNNs (HD-SNNs). Kronland-Martinet et al. [2025] recently extended this to motifs of arbitrary length by chaining sub-motif detectors with bounded synaptic delays, achieving robust encoding of complete Spiking Heidelberg Digits patterns.

The present work generalises this chain-of-motifs framework to enable working memory in a recurrent SNN. Rather than relying on fixed delays and unsupervised STDP rules as in the original polychronisation model, we extend the spike-timing theory of [Kronland-Martinet et al., 2025] to a fully recurrent topology: once primed by a short clamping window, the network autonomously recalls arbitrary-length spike sequences. The model is trained end-to-end with surrogate gradients [Neftci et al., 2019], bridging the computational neuroscience framework of polychronisation with gradient-based machine learning and targeting the energy-efficiency and sparsity objectives central to neuromorphic edge deployment.

## 2 Methods

### 2.1 Representation of spiking activity

In neurobiological recordings or the output of event-based cameras, any generic raster plot consists of a stream of *spikes*, defined as a list of address–timestamp tuples

$$\mathcal{E} = \big\{(a_r,\, t_r)\big\}_{r\in[1,\, N_{\text{ev}}]},$$

where $N_{\text{ev}} \in \mathbb{N}$ is the total number of events and $r$ indexes each event in temporal order [Grimaldi et al., 2024]. Each event has a time of occurrence $t_r$ and an address $a_r$ in the neural population space $\mathcal{A}$. In a Single Unit Activity (SUA) recording for instance, $\mathcal{A}$ is the identified set of neurons.

Neurons are characterised by their membrane potential dynamics and by the integration of synaptic potentials along their dendritic tree. Crucially, neurons can receive inputs from multiple presynaptic neurons with *heterogeneous delays* $\delta_s$, representing the travel time of the presynaptic spike to the soma of the postsynaptic neuron. Scanning all pre- and postsynaptic pairs, we define the set of $N_s \in \mathbb{N}$ synapses as

$$\mathcal{S} = \big\{(a_s,\, b_s,\, w_s,\, \delta_s)\big\}_{s\in[1,\, N_s]},$$

where $a_s \in \mathcal{A}$ and $b_s \in \mathcal{A}$ are respectively the pre- and postsynaptic addresses, $w_s$ is the synaptic weight, and $\delta_s$ is the delay. Note that neurons in the human cortex have on average 30000 synapses [Rockland, 2002], an order of magnitude which is seldom reached in artificial networks. In our model, two neurons may contact each other via multiple synapses, each with a different delay. The *receptive field* of neuron $b$ is the sub-set of synapses that project onto it:

$$\mathcal{S}^b = \big\{(a_s,\, b_s,\, w_s,\, \delta_s) \mid b_s = b\big\}_{s\in[1,\, N_s]}.$$

Each spike from the event stream $\mathcal{E}$ gets multiplexed in time by these synapses to later converge on the post-synaptic neurons. Crucially, as they reach the post-synaptic neuron, a synchronous convergence of multiple spikes on the soma — made possible by the interplay of heterogeneous delays — increases the postsynaptic firing probability, enabling the detection of precise spiking motifs (see Figure 1).

To allow for the simulation of this network on standard hardware we transform this event-based representation into the Boolean matrix $A \in \{0,1\}^{N\times T}$, where $N = |\mathcal{A}|$ is the number of neurons and $T$ is the number of discrete time steps of $1\,\text{ms}$ each (see Figure 2, right panel). Delays are discretised as integers in $[0,\, D)$, so the synaptic set for all neurons is compactly represented by the weight tensor $\mathbf{W} \in \mathbb{R}^{N\times N\times D}$, with

$$\mathbf{W}_{b_s,a_s,\delta_s} = w_s \quad \forall s \in [1,\, N_s], \quad \text{and zero otherwise.}$$

### 2.2 Recurrent HD-SNN architecture

The recurrent Heterogeneous-Delay SNN (HD-SNN) consists of a population $\mathcal{A}$ of $N = 512$ leaky integrate-and-fire (LIF) neurons implemented in SNNTORCH [Eshraghian et al., 2023] and simulated over $T = 1000$ steps of $1\,\text{ms}$ each (total duration: $1\,\text{s}$). Extending the feedforward Spiking Motif Detector of Perrinet [2023], our architecture includes full recurrence: each neuron $j \in \mathcal{A}$ integrates inputs from *all* other neurons across the preceding $D = 41$ time steps. This particular choice for the

maximum delay is justified from biological observations and similar to the value used in the original polychronisation study [Izhikevich, 2006]. The membrane potential $u_j(t)$ evolves as

$$u_j(t) = \beta\, u_j(t-1)\,(1 - s_j(t-1)) + \sum_{i=1}^{N}\Big(\sum_{d=1}^{D} \mathbf{W}_{j,i,d} \cdot s_i(t-d)\Big), \tag{1}$$

where $\beta \in (0,1)$ is the membrane decay constant (initialised to $\beta_0 = 0.5$ and updated during training), $s_j(t) \in \{0,1\}$ is the spike of neuron $j$ at time $t$, and $\mathbf{W}_{j,i,d}$ is the synaptic weight from neuron $i$ to neuron $j$ at delay $d$. A spike is emitted when $u_j(t)$ exceeds the threshold $\vartheta = 1$, after which the membrane is reset.

In our implementation, the spike history of all $N$ neurons over the past $D$ steps is concatenated into a flattened context vector of dimension $N \cdot D$ at each time step, so that $\mathbf{W}$ is reshaped as $\mathbf{W} \in \mathbb{R}^{N\times(N\cdot D)}$. This is equivalent to the temporal convolution that we used previously in a feedforward setting [Perrinet, 2023], here extended to a fully recurrent topology. Using such an implementation, the HD-SNN is thus corresponding to a standard computational graph which we could train using surrogate gradient descent (see Figure 2 - Left).

### 2.3 Synthetic dataset and memory cuing

To assess working-memory capacity, we generated a dataset of $M = 8$ distinct target patterns $\mathbf{S}^{(\mu)} \in \{0,1\}^{N\times T}$. Following Perrinet [2023], each pattern is drawn as independent Bernoulli trials over $N$ neurons and $T$ time steps with

$$\mathbf{S}^{(\mu)} \sim \sigma(\sigma^{-1}(p_A) + e_A^{(\mu)})$$

where $p_A$ is the mean firing rate that we set as 1 Hz (spike probability $10^{-3}$ per neuron per step) and the evidence term ${e_A}^{(\mu)} \in \{0,1\}^{N\times T}$ is first drawn from a normal distribution of standard deviation $E_{SM} = 4$:

$${e_A}^{(\mu)} \sim \mathcal{N}(0, e)$$

To simulate realistic sparse activity, we retain only the top $p_{SM} = 0.5\%$ of these activation values, setting the rest to zero. Finally, a refractory period is enforced by imposing a minimum inter-spike interval of 4 ms, yielding biologically plausible sparse activity.

Given a target pattern $\mathbf{S}^*$, each contiguous window of length $D$, i.e. the slice $\mathbf{S}^*[\cdot,\, t{-}D : t{-}1]$, constitutes a unique context that may predict the spiking activity of all neurons at time $t$. The network is therefore trained to generate a sequential chain of overlapping Spiking Motifs [Perrinet, 2023, Kronland-Martinet et al., 2025] that faithfully reproduces the target. Memory retrieval is cued by clamping all neuron activities to the target pattern during the initial $D$ time steps ($0 \le t < D$), corresponding to one elementary Spiking Motif (see Figure 2 - right); the network then generates the remainder of the sequence autonomously.

### 2.4 Analytical weight initialisation

Before training, the weight tensor $\mathbf{W}$ can be initialised analytically by treating the task as a linear regression problem. For each target pattern $\mathbf{S}^{*(\mu)} \in \{0,1\}^{N\times T}$, the context window at time $t$ is the flattened vector $\mathbf{c}^{(\mu)}(t) \in \{0,1\}^{N\cdot D}$ containing the spike history of all neurons over steps $t-D$ to $t-1$. Stacking all $K = M \cdot (T-D)$ context–output pairs across patterns and time steps yields

$$\mathbf{C} \in \{0,1\}^{ND\times K}, \qquad \mathbf{S}^* \in \{0,1\}^{N\times K},$$

and the initialisation problem is $\mathbf{W}\,\mathbf{C} \approx \mathbf{S}^*$. Since the system is strongly underdetermined ($K \ll ND$), the minimum-norm solution is the Moore–Penrose pseudo-inverse $\mathbf{W}^* = \mathbf{S}^*\,\mathbf{C}^{+}$. With sparse Bernoulli activity at rate $p_A = 10^{-3}$, consecutive context windows are nearly orthogonal, so the Gram matrix satisfies $\mathbf{C}\mathbf{C}^{\top} \approx N \cdot D \cdot p_A\,\mathbf{I}$, giving the closed-form Hebbian initialisation

$$w_{ij}^{(d)} = \frac{1}{NDp_A\,M} \sum_{\mu=1}^{M} \sum_{t=D+1}^{T} s_j^{*(\mu)}(t) \cdot s_i^{*(\mu)}(t-d). \tag{2}$$

Each weight $w_{ij}^{(d)}$ is thus proportional to the normalised cross-correlation between the spike train of neuron $j$ and that of neuron $i$ shifted by delay $d$, averaged over all $M$ stored patterns. This

initialisation resembles classical Hebbian learning and mirrors the form of spike-timing-dependent plasticity (STDP) as notably used in the seminal paper on polychronisation [Izhikevich, 2006]. Notice however that at inference time, the network does not receive ground-truth contexts, but it receives the context it generated itself at the previous step such that the spike train may be corrupted by previous errors.

### 2.5 Training

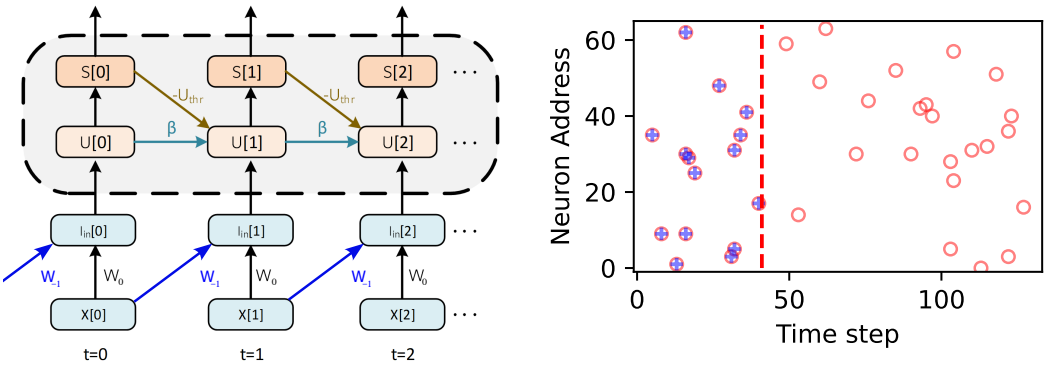


Figure 2: Network architecture and a sample target pattern. *Left:* Unrolled view of the recurrent HD-SNN over time, illustrating delayed connections between neurons: Blue arrows indicate synaptic connections that integrate past activity across heterogeneous delays; for clarity only one delay is shown. *Right:* One representative target spike pattern ($N = 512$ neurons, $T = 1000$ time steps, of which we show 64 neurons and 128 time steps); each red circle marks one target spike. The goal of the network is to memorise $M = 8$ such patterns, initialised by clamping all neuron activities to the target during the first $D = 41$ time steps (as denoted by blue crosses in the region left of the dashed red line).

The loss is defined as $\mathcal{L} = 1 - F_1$, where $F_1$ is the mean F1 score between the predicted and target spike trains, averaged over time (in the prediction window), neurons and patterns. The F1 score is the harmonic mean of precision (proportion of produced spikes that are correct) and recall (proportion of target spikes that are successfully reproduced), so minimising $\mathcal{L}$ simultaneously penalises over-active networks (low precision) and silent networks (low recall). Learning is carried out with surrogate-gradient BPTT [Neftci et al., 2019], using a fast-sigmoid surrogate with sharpness parameter $\alpha = 10$. Weights are optimised with Adam (learning rate $\eta = 10^{-6}$, momentum $\mu = 10^{-1}$, weight decay $\lambda = 10^{-3}$). Training runs for 4096 gradient steps; to mitigate overfitting and stabilise delay learning, the learning rate is annealed with a cosine schedule, as used for instance in Hammouamri et al. [2023]. The validation loss is monitored for detecting overfitting.

The gradient-based learning in our recurrent HD-SNN shares fundamental principles with biological STDP, despite operating at different scales. Both mechanisms exploit temporal correlations in spike timing to adjust synaptic strengths. However, while STDP implements local pairwise updates based on individual spike timings, our approach performs global optimisation that effectively learns the collective interaction of multiple delayed spike trains across the entire network. The analytical initialisation (see Equation 2) reveals this connection most clearly: it implements a Hebbian-like rule that is mathematically equivalent to averaging STDP updates across all stored patterns, suggesting that gradient descent in this context can be viewed as a principled, globally optimal implementation of temporally structured synaptic plasticity.

Our implementation is available at `https://github.com/laurentperrinet/MNESIS`. The recurrent HD-SNN was implemented using the snnTorch [Eshraghian et al., 2023] library with custom extensions for heterogeneous delay handling. The implementation leverages PyTorch with the Metal Performance Shader (MPS) backend for accelerated computation on Apple Silicon (M3 Ultra chip featuring 32 CPU cores and 80 GPU cores, with 512 GB of unified memory architecture) and further tested with the CUDA backend on the Jean Zay supercluster at GENCI. The MPS and CUDA backends enabled efficient training and simulation of the large recurrent networks required for working memory tasks, leveraging the parallel processing capabilities of the GPU cores while maintaining

low-latency access to the substantial memory resources necessary for storing all tensors' parameters. Training typically converged within 4096 gradient steps, requiring approximately 10 minutes of computation time per training.

# 3 Results

## 3.1 Training dynamics

Performing the training on the recurrent HD-SNN with surrogate-gradient BPTT Neftci et al. [2019] consistently maximised the mean F1 score across all $M = 8$ target patterns to an average value of $1 - 3.4 \cdot 10^{-2}$. We observed two characteristic failure modes during early training: episodes of complete silence, in which the network emits no spikes, and over-activation, in which nearly all neurons fire at every time step. Both are well-known pathological attractors of recurrent SNNs [Izhikevich, 2006]. The F1 score is the harmonic mean of precision and recall, and maximising it therefore penalises both failure modes symmetrically: low precision corresponds to an over-active network and low recall to a silent one. In addition, the cosine learning-rate schedule [Hammouamri et al., 2023] helped guide the network away from these extremes towards a balanced, sparse-firing regime.

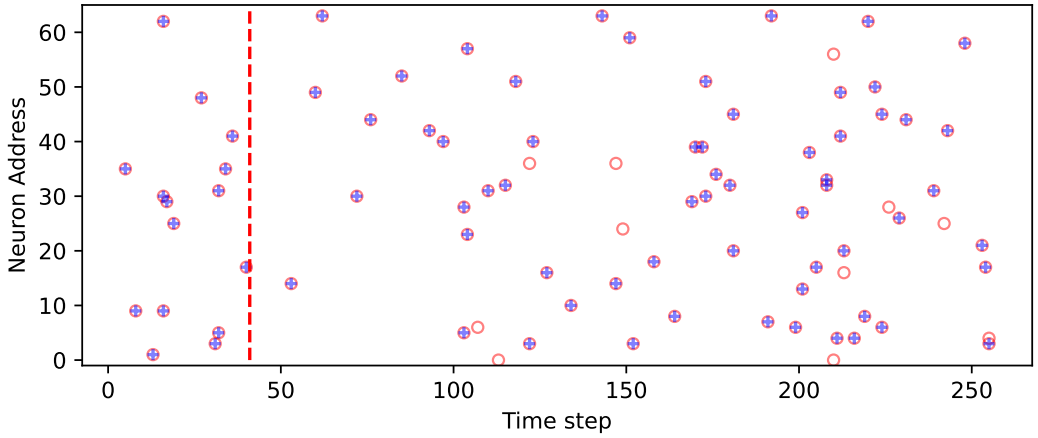


Figure 3: Learning dynamics of the recurrent HD-SNN. The raster plot shows network activity (blue crosses, $N = 512$ neurons) versus the target pattern (red circles) at an intermediate training epoch (512 epochs). Correct recall emerges first in the time steps immediately following the clamped initialisation window ($0 \leq t < D = 41$, delimited by the dashed red line) and progressively extends to the full sequence duration ($T = 1000$ steps).

Training was not uniform in time: the network first reproduced spikes immediately following the clamping window ($0 \leq t < D$), then progressively extended correct recall towards later time steps, as shown in Figure 3. This behaviour is a direct consequence of the sequential structure of the task: because each window $[t - D,\, t - 1]$ uniquely predicts the activity at $t$, errors at early time steps propagate into all subsequent predictions, making temporal proximity to the clamped initial condition the primary driver of learning progress. We therefore expect the number of gradient steps required to memorise a pattern to scale approximately linearly with its duration $T$.

## 3.2 Effect of delay depth, pattern duration, and background firing rate

To characterise how the architectural parameters govern working-memory performance, we conducted a systematic one-at-a-time scan over the maximum delay $D$, the pattern duration $T$, and the background firing rate $p_A$, holding all other hyperparameters at their default values (Section 2). Each condition was evaluated with $N_{\text{cv}} = 10$ by drawing different seeds for the random number generator and results are reported as mean loss $\mathcal{L} = 1 - F_1$ (Figure 4).

**Delay depth $D$ (Figure 4, left).** The loss decreases monotonically as $D$ grows from 3 to 127 ms, dropping from $\mathcal{L} \approx 0.85$ at $D = 3$ to near zero at $D = 127$. This confirms an intuition about

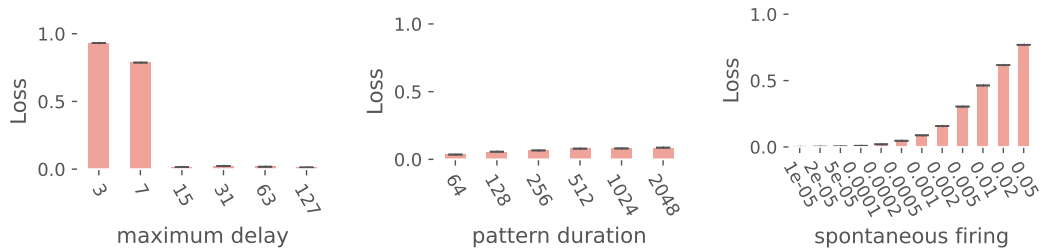


Figure 4: Effect of three key parameters on working-memory performance (loss $\mathcal{L} = 1 - F_1$, mean $\pm$ s.e. over $N_{\text{cv}} = 10$ cross-validation folds). *Left:* Increasing the maximum delay $D$ consistently reduces the loss, confirming that deeper delay windows provide a larger, more orthogonal context and that longer maximum delays directly increase the capacity of the recurrent HD-SNN. *Middle:* Loss grows slowly and monotonically with pattern duration $T$, consistent with the compounding-error and gradient-vanishing arguments: longer sequences require the network to maintain coherent predictions further from the clamped initialisation window. *Right:* Performance is optimal at low background firing rates ($p_A \lesssim 10^{-3}$) and degrades sharply at higher rates.

the capacity of the network: the parameter budget $N^2 \times D$ and the context-window size $N \times D$ both scale linearly with $D$. A wider delay window provides more temporally diverse synaptic pathways, increasing the number of simultaneously representable Spiking Motifs and reducing the interference between overlapping context windows. Concretely, at $D = 3$ the context vector has only $N \times D = 768$ dimensions with expected activity $N \times D \times p_A \approx 0.77$ per window — a regime where contexts are nearly all-zero and indistinguishable. At $D = 127$ the context has $N \times D = 32512$ dimensions with expected activity $\approx 32.5$, and near-orthogonality is strongly satisfied.

**Pattern duration $T$ (Figure 4, middle).** The loss grows monotonically with $T$, from $\mathcal{L} \approx 0.004$ at $T = 64$ to $\mathcal{L} \approx 0.08$ at $T = 2048$. This is the empirical signature of the two compounding difficulties identified in the capacity analysis. Longer sequences impose a larger open-loop rollout: the error propagation recurrence accumulates over $T - D$ steps, and even a sub-critical propagation coefficient $\lambda < 1/D$ eventually allows errors to reach a non-negligible fraction of $N$ spikes.

**Background firing rate $p_A$ (Figure 4, right).** The loss exhibits a clear optimum around $p_A = 10^{-4}$ to $10^{-3}$ and degrades rapidly for $p_A \geq 2 \times 10^{-3}$. This is consistent with the capacity bound linked to interference: increasing $p_A$ linearly reduces the SNR of the cross-correlation initialisation and increases the Gram-matrix off-diagonal terms, degrading context orthogonality.

Taken together, the three scans quantitatively validate the theoretical framework: delay depth is the primary capacity lever (loss $\propto 1/(N \times D \times p_A)$), pattern duration is the primary difficulty lever (loss grows with $T$ through gradient vanishing and error compounding), and firing rate sets the orthogonality regime that determines whether the initialisation and learning are likely to converge. These results suggest that the most efficient path to storing longer patterns is to increase $D$ rather than $N$, since $D$ enters both the parameter capacity $N^2 \times D$ and the context orthogonality $N \times D \times p_A$ while adding no new neurons or lateral connections. Yet, increasing $D$ is limited by biological constraints [Grimaldi et al., 2023].

## 4 Discussion

### 4.1 Memory capacity and the role of heterogeneous delays

We have demonstrated that a recurrent SNN with heterogeneous synaptic delays can be trained end-to-end to reproduce arbitrary target spike patterns. The result is noteworthy from a capacity standpoint: the entire network is parameterised by a single weight tensor $\mathbf{W} \in \mathbb{R}^{N \times N \times D}$ of dimension $N^2 \times D = 512^2 \times 41 \approx 10^7$ parameters, yet it can faithfully store $M$ patterns each of $N \times T$ binary values. This compression is made possible precisely by the temporal structure introduced by the $D$ delay channels: rather than requiring one parameter per stored bit, the network exploits the sequential dependencies between consecutive windows of length $D$ to represent the entire pattern as a chain of overlapping Spiking Motifs [Perrinet, 2023, Kronland-Martinet et al., 2025].

This provides concrete evidence that heterogeneous delays are a computational asset rather than a biological nuisance — a point argued theoretically in Izhikevich [2006] and in the context of learnable delays by Grimaldi and Perrinet [2023] and Hammouamri et al. [2023], and now demonstrated in a recurrent working-memory setting.

### 4.2 Limitations and directions for improvement

The present model is deliberately minimal, and several design choices impose limits on capacity and robustness that future work should address.

**Capacity and robustness.** A systematic evaluation of memory capacity — how many patterns of what duration can be reliably stored and retrieved — remains to be carried out. The theoretical capacity of polychronous networks scales super-linearly with network size [Izhikevich, 2006], but the practical limit under gradient-based training and the sensitivity of recall to initial-condition perturbations are open questions. The analogy with attractor networks [Szatmáry and Izhikevich, 2010] suggests that capacity should scale with $N^2 \times D$, but interference between stored patterns at high memory load may degrade precision in ways that the F1 training objective does not fully penalise.

**Richer neuron models and dynamics.** We used the simplest LIF model with a learned membrane decay $\beta$. Introducing adaptive threshold dynamics or more complex single-neuron models, as studied in Szatmáry and Izhikevich [2010], should increase the diversity of intrinsic timescales available to the network and improve both capacity and retrieval fidelity.

**Latent reservoir neurons.** The current architecture connects every neuron directly to every other neuron across all delays. Partitioning the population into input-output neurons and a larger pool of latent "reservoir" neurons, as in delay-based reservoir computing [Paugam-Moisy et al., 2008], would enrich the internal spiking dynamics. Such a hybrid architecture could combine the representational power of random recurrent networks with the precision of gradient-trained delay weights [Neftci et al., 2019], potentially closing the gap between biologically inspired models and the efficiency of sparse selective-update architectures [Yin et al., 2026].

### 4.3 Perspectives: unsupervised learning and neural data

The most immediate application of this framework is to neurophysiological data. Future work will explore whether self-supervised objectives can extract Spiking Motifs directly from raw multi-electrode array recordings without labelled targets. Such approaches could bridge the gap between biologically plausible learning and practical deployment on edge devices. The supervised memorisation task studied here requires a known target, but in practice one would like to discover the latent Spiking Motifs present in recorded multi-unit activity — the problem originally addressed by Ikegaya et al. [2004] and Villette et al. [2015]. A natural extension is therefore to replace the supervised F1 loss with a self-supervised or contrastive objective that trains the network to predict upcoming spikes from the recent spike history, without access to ground-truth labels. This is analogous to the contrastive predictive coding paradigm applied to continuous neural time series, here instantiated in a fully spike-based setting [Grimaldi et al., 2023]. Such a model would provide a principled, energy-efficient method for identifying the temporal structure of neural population codes, with direct relevance to the debate on rate coding versus temporal coding [Softky and Koch, 1993, Bohte, 2004] and to understanding the neural mechanisms of working memory more broadly. Deployed on neuromorphic hardware [Göltz et al., 2021, Eshraghian et al., 2023], it could operate in real time on streaming electrophysiological data, opening a new avenue for closed-loop brain-machine interface applications.

## Acknowledgments and Disclosure of Funding

This work was performed using HPC resources from GENCI-IDRIS (Grant 20XX–AD010314955R2). The authors declare no competing interests regarding this manuscript. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. 

## References


M. Abeles. Role of the cortical neuron: Integrator or coincidence detector? *Israel Journal of Medical Sciences*, 18(1):83–92, January 1982. ISSN 0021-2180.

E. D. Adrian and Yngve Zotterman. The impulses produced by sensory nerve endings. *The Journal of Physiology*, 61(4):465–483, August 1926. ISSN 0022-3751. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1514868/.

Elie Bienenstock. A model of neocortex. *Network: Computation in Neural Systems*, 6(2):179–224, January 1995. ISSN 0954-898X.

Sander M. Bohte. The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, 3(2):195–206, 2004. doi: 10.1023/B:NACO.0000027755.02868.60.

Randy M. Bruno and Bert Sakmann. Cortex Is Driven by Weak but Synchronously Active Thalamocortical Synapses. *Science*, 312(5780):1622–1627, 2006. doi: 10.1126/science.1124593. URL https://doi.org/10.1126/science.1124593.

Akihiro Eguchi, James B. Isbister, Nasir Ahmad, and Simon Stringer. The emergence of polychronization and feature binding in a spiking neural network model of the primate ventral visual system. *Psychological Review*, 125(4):545–571, 2018. ISSN 1939-1471. doi: 10.1037/rev0000103. URL https://psycnet.apa.org/fulltext/2018-25960-001.pdf.

Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023. doi: 10.1109/JPROC.2023.3308088.

J. Göltz, L. Kriener, A. Baumbach, S. Billaudelle, O. Breitwieser, B. Cramer, D. Dold, A. F. Kungl, W. Senn, J. Schemmel, K. Meier, and M. A. Petrovici. Fast and energy-efficient neuromorphic deep learning with first-spike times. *Nature Machine Intelligence*, 3(9):823–835, September 2021. ISSN 2522-5839. doi: 10/gm5gwd. URL https://www.nature.com/articles/s42256-021-00388-x.

Antoine Grimaldi and Laurent U. Perrinet. Learning heterogeneous delays in a layer of spiking neurons for fast motion detection. *Biological Cybernetics*, 117:373–387, September 2023. doi: 10.1007/s00422-023-00975-8. URL https://link.springer.com/article/10.1007/s00422-023-00975-8.

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, January 2023. ISSN 2076-3425. doi: 10.3390/brainsci13010068. URL https://www.mdpi.com/2076-3425/13/1/68.

Antoine Grimaldi, Victor Boutin, Sio-Hoi Ieng, Ryad Benosman, and Laurent U. Perrinet. A robust event-driven approach to always-on object recognition. *Neural Networks*, 178:106415, 2024. doi: 10.1016/j.neunet.2024.106415. URL https://www.sciencedirect.com/science/article/pii/S0893608024003393.

Ilyass Hammouamri, Ismail Khalfaoui-Hassani, and Timothée Masquelier. Learning Delays in Spiking Neural Networks using Dilated Convolutions with Learnable Spacings, June 2023. URL http://arxiv.org/abs/2306.17670.

Yuji Ikegaya, Gloster Aaron, Rosa Cossart, Dmitriy Aronov, Ilan Lampl, David Ferster, and Rafael Yuste. Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity. *Science*, 304 (5670):559–564, April 2004. doi: 10.1126/science.1093173. URL https://www.science.org/doi/10.1126/science.1093173.

Eugene M. Izhikevich. Polychronization: Computation with Spikes. *Neural Computation*, 18 (2):245–282, February 2006. ISSN 0899-7667. doi: 10.1162/089976606775093882. URL https://doi.org/10.1162/089976606775093882.

Thomas Kronland-Martinet, Stéphane Viollet, and Laurent U Perrinet. Detection of spiking motifs of arbitrary length in neural activity using bounded synaptic delays, 2025. URL https://arxiv.org/abs/2511.15296.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36(6):51–63, November 2019. ISSN 1053-5888. doi: 10.1109/MSP.2019.2931595. URL https://ieeexplore.ieee.org/document/8891809/.

Hélène Paugam-Moisy, Régis Martinez, and Samy Bengio. Delay learning and polychronization for reservoir computing. *Neurocomputing*, 71(7–9):1143–1158, March 2008. ISSN 0925-2312. doi: 10.1016/j.neucom.2007.12.027. URL http://www.sciencedirect.com/science/article/pii/S0925231208000507.

L. Perrinet and M. Samuelides. Coherence detection in a spiking neuron via Hebbian learning. *Neurocomputing*, 44–46:133–139, 2002. doi: 10.1016/s0925-2312(02)00374-0.

Laurent U. Perrinet. Accurate Detection of Spiking Motifs in Multi-unit Raster Plots. In Lazaros Iliadis, Antonios Papaleonidas, Plamen Angelov, and Chrisina Jayne, editors, *Artificial Neural Networks and Machine Learning — ICANN 2023*, pages 369–380, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-44206-3. doi: 10.1007/978-3-031-44207-0_31.

Kathleen S. Rockland. Non-uniformity of extrinsic connections and columnar organization. *Journal of Neurocytology*, 31(3-5):247–253, 2002. ISSN 0300-4864. doi: 10.1023/a:1024169925377.

W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience*, 13(1):334–350, January 1993. doi: 10.1523/jneurosci.13-01-00334.1993. URL https://doi.org/10.1523/JNEUROSCI.13-01-00334.1993.

Botond Szatmáry and Eugene M. Izhikevich. Spike-Timing Theory of Working Memory. *PLoS Computational Biology*, 6(8):e1000879, August 2010. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000879. URL https://dx.plos.org/10.1371/journal.pcbi.1000879.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *NIPS*, pages 5998–6008, 2017. doi: 10.48550/arXiv.1706.03762. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.09.052. URL https://www.sciencedirect.com/science/article/pii/S0896627315008417.

Bojian Yin, Shurong Wang, Haoyu Tan, Sander Bohte, Federico Corradi, and Guoqi Li. Efficient Sparse Selective-Update RNNs for Long-Range Sequence Modeling, February 2026. URL http://arxiv.org/abs/2603.02226.