Working Memory with Polychronous Chains

Laurent U Perrinet

Institut de Neurosciences de la Timone (UMR 7289) - Aix Marseille Univ, CNRS

Chains of polychronization

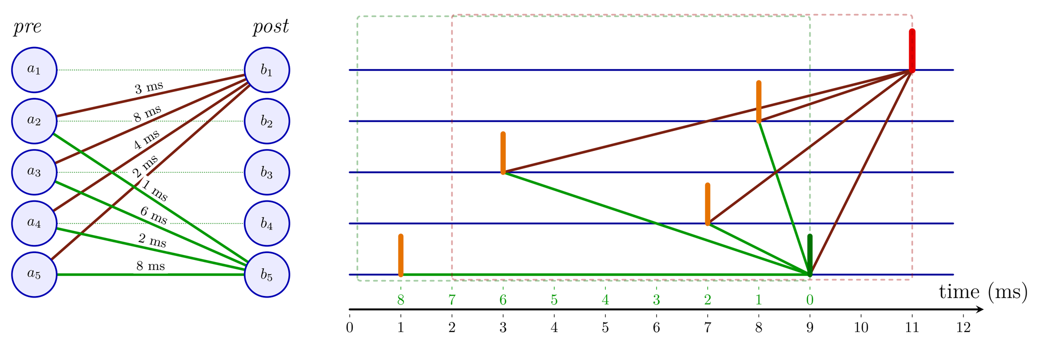

Figure 1: **Working memory as a chained sequential spike prediction via heterogeneous delays.** *Left:* A recurrent HD-SNN of five neurons numbered 1–5, connected by two subsets of recurrent connections that define two Spiking Motifs from the pre-synaptic side (neurons a) to the post-synaptic side (neurons b): dark-red connections from a_2, a_3, a_4, and a_5 target b_1 with delays 3, 8, 4, 2 ms; green connections from a_2, a_3, a_4, and a_5 target b_5 with delays 1, 6, 2, 8 ms. *Right:* Raster plot illustrating how the network predicts successive spikes from a past context window. *Step 1 — first prediction (green dashed box):* the orange input spikes emitted at staggered times by a_2–a_5 within the first context window (dashed green box) constitute a first Spiking Motif. The green delays are tuned so that all four spikes converge *synchronously* at b_5, crossing the firing threshold and producing the green output spike with a precise time. *Step 2 — second prediction (dark-red dashed box):* the green spike of a_5 re-enters the network as a new input. Together with the spikes now visible in the shifted context window (dark-red dashed box), spikes are routed through the dark-red delays in order to converge synchronously on b_1, generating a new 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 polychronous motifs in which each new temporal prediction is grounded in the preceding context window.

Architecture of the model

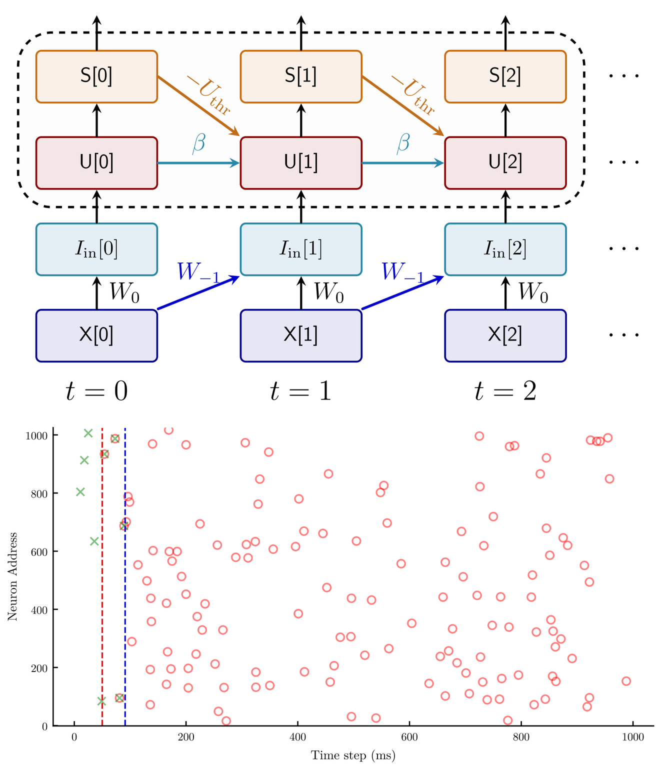

Figure 2: Network architecture and a sample target pattern. *Top:* Unrolled view of the recurrent HD-SNN over time (one column representing one time step), illustrating delayed connections between neurons: Blue arrows indicate synaptic connections that integrate past activity across heterogeneous delays; for clarity only one delay is shown. *Bottom:* One representative target spike pattern ($N = 1024$ neurons, $T = 1000$ time steps); each red circle marks one target spike. The goal of the network is to memorise $M = 16$ such patterns. Each trial is flanked by $N_{\text{pretime}} = 50$ time steps of spontaneous background activity (random spikes at rate p_A) before and after the stimulus. The trigger window (green crosses between the dashed lines) consists of the first $D = 41$ time steps of the target pattern clamped to the network input; the recurrent network then autonomously predicts the remaining spikes.

Pattern Retrieval

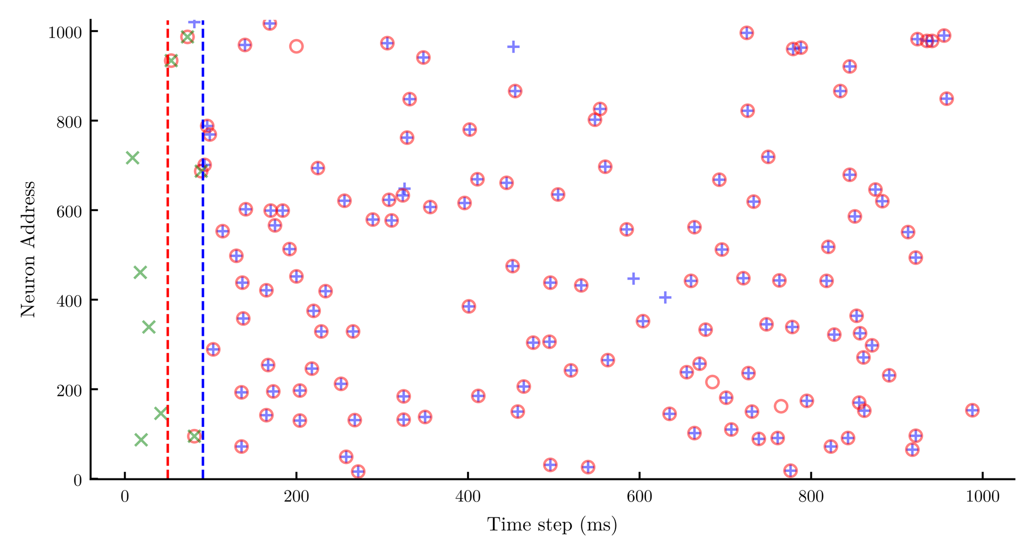

Figure 3: Pattern retrieval by the recurrent HD-SNN after training. The network is clamped with input spikes (green crosses) comprising 50 steps of spontaneous activity (left of the red dashed line) followed by the trigger window consisting of the initial sequence of one of the $M = 16$ target motifs ($0 \leq t < D = 41$, delimited by the red and blue dashed lines). We then simulate the autonomous replay over the full sequence duration ($T = 1000$ steps). We observe that the raster plot shows network activity (blue plus signs, $N = 1024$ neurons) which accurately matches the hidden target pattern which was cued during the trigger window. Note that the target spikes after the trigger window (red circles right of the blue dashed line) are hidden from the network.

Sequential Retrieval

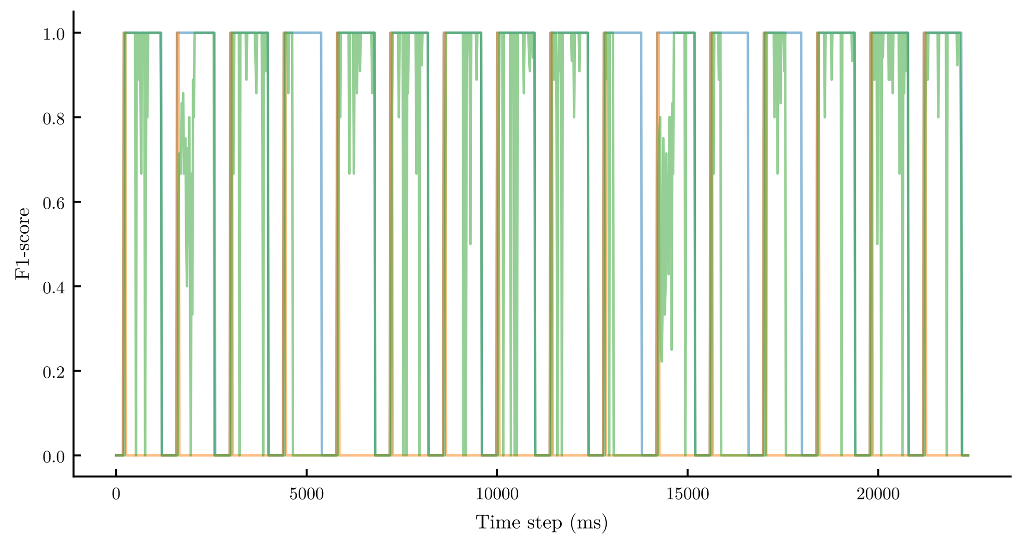

Figure 4: Sequential retrieval of the $M = 16$ stored motifs. The sliding-window F_1 score (green) is shown as a function of time, together with the trigger mask, equal to 1 on the $D = 41$ steps of each trigger window (orange), and the ideal retrieval envelope, equal to 1 over the full $T = 1000$-step duration of each motif (blue). The F_1 score rises sharply at each trigger-window onset and remains high for the full motif duration ($T = 1000$ steps ~ 1 second), then returns to baseline during the inter-trial intervals (of length $N_{\text{ITI}} = 4 \cdot N_{\text{pretime}} = 200$ steps of silence), demonstrating selective retrieval of all stored memories in sequence without cross-interference.

Navigating the Lorenz attractor

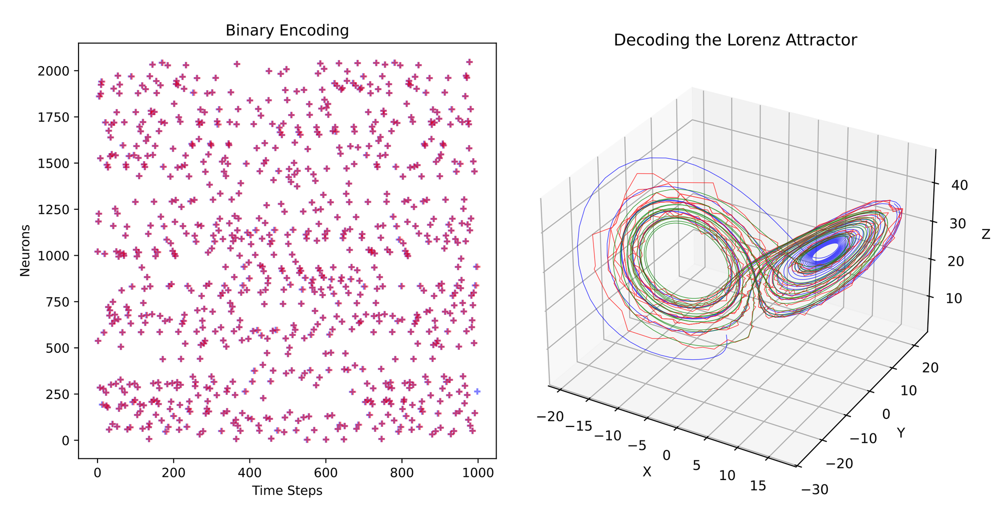

Figure 5: We used the equations of the Lorenz attractor to generate a time series of chaotic dynamics. *Left:* We used a simple encoding scheme to convert the continuous dynamics into a sequence of discrete events using a nearest-neighbor one-hot encoding. This allows us to reconstruct the trajectory from spikes. *Right:* After learning, the network is able to autonomously generate the chaotic dynamics of the Lorenz attractor, starting from a short trigger window of input spikes. The network output (red) closely matches the target trajectory (blue) as it encoded by spikes (green) for all three dimensions of the Lorenz attractor.

Learning Travelling waves

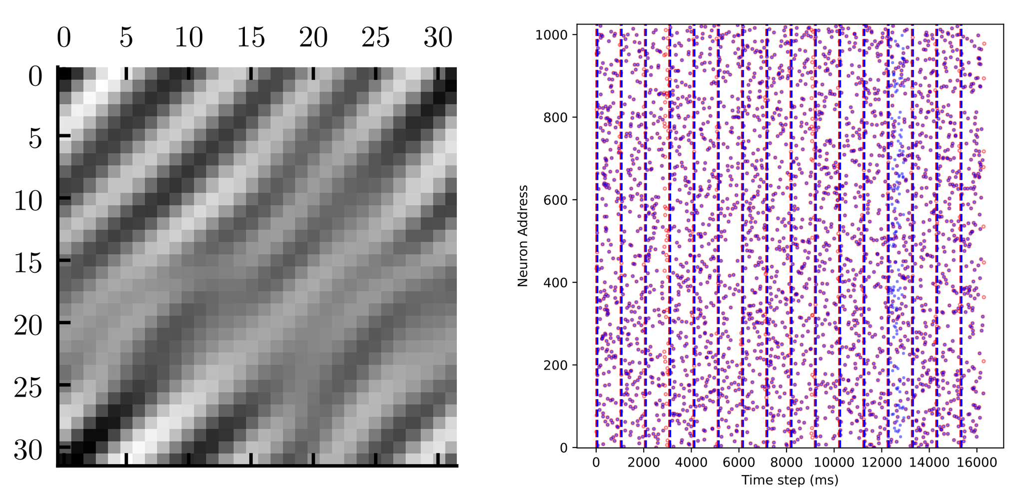

Figure 6: *Left:* We generated travelling waves in 16 different directions using a naturalistic synthetic texture which we encoded into spikes. *Right:* The model was able to learn the travelling waves and reproduce them with high fidelity. Here we show that the network's spiking activity output (blue) when triggered during 41 ms every second with a new direction. The activity closely matches the hidden target spikes (red).

Robustness

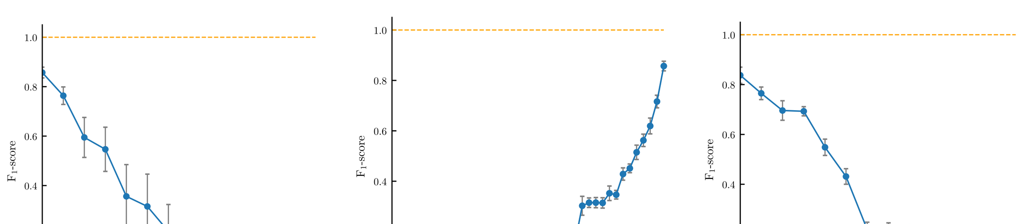

Figure 7: Robustness of working-memory retrieval to three types of trigger-window perturbation (F_1 score averaged over $M = 16$ patterns). *Left:* Bit-flip noise applied to all spikes in the trigger window ($p_{\text{flip}} \in [0, 1]$); the network maintains high F_1 up to $p_{\text{flip}} \approx 0.2$. *Middle:* Truncated trigger window (trigger duration from 0 to $D - 1 = 40$ steps); retrieval remains reliable for durations above roughly 32 steps. *Right:* Partial neuron coverage (number of silenced neurons from 0 to $N = 1024$); perfect recall is maintained until the majority of the trigger population is inactive. Orange dashed lines indicate extremal F_1 values 0 or 1.

Parameter Scan

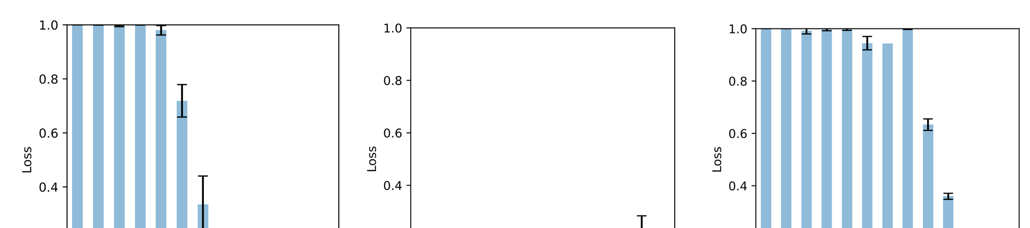

Figure 8: Effect of some key parameters on working-memory performance (loss $\mathcal{L} = 1 - F_1$, mean $\pm$ s.e. over $N_{\text{CV}} = 10$ cross-validation folds). *Left:* Increasing the maximum delay D consistently reduces the loss, confirming that deeper delay windows provide a larger, more orthogonal 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 trigger window. *Right:* Performance is optimal for a number of neurons superior to ≈ 500 the loss deteriorating rapidly for lower numbers of neurons.

[1] Eugene M. Izhikevich. "Polychronization: Computation with Spikes". In: *Neural Computation* 18.2 (Feb. 2006), pp. 245–282. ISSN: 0899-7667. DOI: `10.1162/089976606775093882`.

[2] Antoine Grimaldi et al. "Precise Spiking Motifs in Neurobiological and Neuromorphic Data". In: *Brain Sciences* 13.1 (Jan. 2023), p. 68. ISSN: 2076-3425. DOI: `10.3390/brainsci13010068`.

[3] Laurent U. Perrinet. "Accurate Detection of Spiking Motifs in Multi-unit Raster Plots". In: *Artificial Neural Networks and Machine Learning — ICANN 2023*. Ed. by Lazaros Iliadis et al. Cham: Springer Nature Switzerland, 2023, pp. 369–380. ISBN: 978-3-031-44206-3. DOI: `10.1007/978-3-031-44207-0_31`.

[4] Antoine Grimaldi and Laurent U. Perrinet. "Learning Heterogeneous Delays in a Layer of Spiking Neurons for Fast Motion Detection". In: *Biological Cybernetics* 117 (Sept. 2023), pp. 373–387. DOI: `10.1007/s00422-023-00975-8`.

[5] Thomas Kronland-Martinet, Stéphane Viollet, and Laurent U Perrinet. "Detection of Spiking Motifs of Arbitrary Length in Neural Activity Using Bounded Synaptic Delays". In: *Biological Cybernetics* (2026). DOI: `10.48550/ARXIV.2511.15296`. arXiv: `2511.15296`.

[6] Jason K Eshraghian et al. "Training Spiking Neural Networks Using Lessons from Deep Learning". In: *Proceedings of the IEEE* 111.9 (2023), pp. 1016–1054. DOI: `10.1109/JPROC.2023.3308088`.

Contact

27 boulevard Jean Moulin, 13005 Marseille, France— E-mail: `laurent.perrinet@cnrs.fr` — Source code: `https://laurentperrinet.github.io/publication/perrinet-26`